A Systematic Review of Edge Computing Strategies for Real-Time Anomaly Detection and Efficiency in Industrial Automation

KDBR Siriwardena
Department of Electrical, Electronics & Telecommunication
General Sir John Kotelawala Defence University
Ratmalana, Sri Lanka
38-eng-0074@kdu.ac.lk

KT Sehan
Department of Electrical, Electronics & Telecommunication
General Sir John Kotelawala Defence University
Ratmalana, Sri Lanka
38-eng-0120@kdu.ac.lk

***Abstract*- The rapid growth of the Internet of Things (IoT) and real-time applications has driven a paradigmatic shift away from centralized Cloud Computing and toward decentralized Edge Computing (EC). The move brings data processing closer to the source, enabling low-latency responses, reduced network load, and enhanced privacy. Yet, Edge Computing presents unique challenges in terms of managing distributed resources, ensuring data security, and maintaining operational efficiency, especially in latency-sensitive industrial applications.**

This article discusses advanced Edge Computing application in an emulated Bottling Plant with regards to the syrup processing, filling, molding, and packaging stages. We compare two implementations: a basic setup which depends on edge devices with minimal optimization and an improved implementation that applies some advanced techniques developed in recent research work, such as hybrid edge-cloud processing, decentralized resource auctioning, and adaptive runtime allocation. These improvements show drastic reductions in anomaly rates and latency across key metrics for much better process stability and efficiency.

Through this study, we put forward the potential of Edge Computing in industrial automation and discuss its strengths and limitations. Our results provide a framework for deploying optimized EC solutions in environments where real-time data processing is paramount and lay a foundation for future research in Edge Computing applications.

I. INTRODUCTION

The exponential growth of the Internet of Things devices and real-time applications in recent years has pushed the development of Edge Computing as an alternative to Cloud Computing. The increasing integration of IoT into the operations of every industry makes the application of fast and efficient data processing more necessary—at least for those sectors where low latency and real-time response are characteristic, like industrial automation, healthcare, and autonomous systems.

Unlike the cloud-centric models, where data is processed in remote data centers, Edge Computing has thrust computation closer to where data is generated, at the "edge" of the network. This shift has been transformative, enabling faster data processing, reduced bandwidth costs, and better privacy and security through reduced transmission of data to central servers. Decentralization of data processing, however, brings new challenges to the fore: how to cope with seamless communication, distributed resources, and security and privacy at multiple edge nodes.

In this paper, we review the state-of-the-art methods for optimizing edge computing in latency-sensitive applications with an illustrative example of practical implementations in a simulated bottling plant environment.

This setting includes multiple production stages—syrup processing, filling, molding, and packaging—each monitored by edge devices with tailored operational metrics. Comparing an elementary with a more evolved implementation, which employs state-of-the-art edge computing techniques, we demonstrate the impact of techniques such as decentralized resource allocation, hybrid edge-cloud processing, and adaptive runtime on anomaly detection and response times. This paper tries to reveal the benefits and challenges of implementing EC in industrial automation through a detailed review of current research on edge computing, with a focus on resource management and low-latency processing. We also shed light on how certain optimizations, learned from literature about distributed computing, lead to substantial performance improvement and provide a framework for Edge Computing applications in similar real-time scenarios in the future.

II. BACKGROUND AND LITERATURE REVIEW

1. Edge computing in real-time applications

Edge computing basically overcomes the limitations of traditional cloud computing by processing data closer to the IoT devices, hence avoiding latency and lessening the load of centralized processing. This comes in handy in several cases, especially for industries that require real-time processing, such as industrial automation and smart manufacturing.

2. Edge Computing: The Main Challenges—Security and Privacy

This majorly increases the use of EC, introducing big security and privacy concerns, especially since data processing is

distributed over multiple nodes, each representing a potential point of vulnerability. In this line, a Systematic Review on Security and Privacy Requirements in Edge Computing points out that secure transmission and storage of data are the basics for establishing safe operations in an EC environment.
This paper describes how data needs to be secured at every layer of the edge network, and we implemented those principles in our simulation in securing metrics transmissions in every edge device with emphasis on the syrup processing and filling stages where highly sensitive production data is produced.

3. Hybrid Edge-Cloud Collaboration for Distributed Resource Management

In complex systems where high-throughput data is involved, a hybrid model, edge-cloud, promises efficiency in striking a balance between local edge processing and centralized cloud resources. The Cloud-Edge-Device Collaboration Framework presents a model for handling real-time data and complex computations in a manner that allows local nodes to deal with minor anomalies while sending critical events to the cloud. That's how we used the hybrid model in our bottling plant simulation: the syrup room was able to handle minor temperature or viscosity anomalies locally, but more severe deviations were referred to cloud resources. This hybrid approach enhances responsiveness without overloading the edge network.

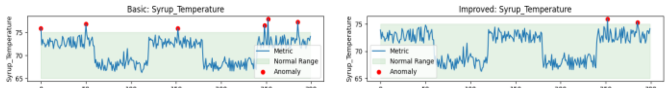


Figure 1:Hybrid Edge-Cloud Processing in Syrup Room: Comparison of Anomaly Rates for Syrup Temperature

4. Decentralized Resource Auctioning for Latency-Sensitive Processes

This would help latency-sensitive tasks in industrial automation by decentralizing resource allocation, which prevents bottlenecks and allows for resource allocation based on demand. According to the paper "Decentralized Resource Auctioning for Latency-Sensitive Edge Computing," auctioning can be used to allow devices to bid for resources, therefore guaranteeing that high-priority tasks get timely attention.
We used this pattern on the PET machine section of our bottling plant simulation and employed frequency-dependent, processing-load-dependent, and latency-sensitivity-dependent triggers; this ensured resources were allocated optimally to keep the mold temperature and air pressure within optimal ranges.

5. Adaptive Runtime and Code Offloading for High-Load Conditions

The filler stage in a bottling plant has a fluctuating processing load due to varying fill rates, CO_2 pressure, and temperature. On an Amino Distributed Runtime Framework, code offloading and adaptive runtime adjustments can be performed to handle efficiently such high-load situations. It optimized the processing capacity of the edge device by dynamically offloading the computation of fillers to the cloud during periods of high demand, allowing the edge device to focus on core, real-time metrics, while the cloud handled ancillary tasks. This adaptive approach reduced filler anomalies like inconsistent fill levels or temperature fluctuations, which provide support for smoother operation.

6. Comparative Edge Computing Approaches in Packaging

The final step in our simulation was packaging, to ensure accurate label alignment and achieve packaging rate targets. Authors in the paper Comparison of Edge Computing Implementations discuss various approaches to EC, including Mobile Edge Computing (MEC) and cloudlet-based systems, and how each can be suited to specific latency and resource constraints. In our enhanced simulation, an adaptive adjustment of runtime parameters was automatically invoked in the event of sudden demand peaks that greatly reduced misalignment and packaging bottlenecks and provided a smoother output.

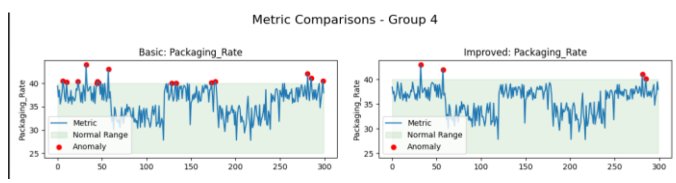


Figure 2:Comparative Edge Computing Approaches in Packaging: Improvement in Packaging Rate and Label Alignment Accuracy

III. METHODOLOGY

We simulate a bottling plant environment to assess the impact of advanced Edge Computing (EC) techniques for real-time data processing and anomaly management. The created environment allows for monitoring critical production metrics through various stages, such as syrup preparation, filler, PET molding, and packaging. We implemented two versions: a basic implementation with minimal edge processing and an improved implementation that integrates advanced EC techniques based on recent research. The following methodology explains the setup, data generation, metrics monitoring, and applied improvements at each stage.

1. Simulation Setup

The simulation was done as a web application using Node.js, Express, and JavaScript for both frontend and backend processing. The metrics were simulated in real time; hence, data were updated every second of actual bottling plant conditions. Data were logged constantly in CSV files for post-analysis, while the simulation included automatic anomaly generation to mimic real-world fluctuations in production metrics.

Each edge device in the simulation corresponded to one of the production stages and handled the following metrics:

Syrup Room: Temperature, viscosity, Brix level (sugar concentration), pH.
Filler: CO_2 pressure, fill level, temperature, and filling rate.
PET Machine: Mold temperature, air pressure, and production rate.
Packaging: Label alignment accuracy and packaging rate.

2. Data Generation and Anomaly Simulation

To be realistic, metrics were generated based on predefined normal ranges with anomalies introduced at regular intervals to simulate production irregularities. For example, the temperature in the syrup room was kept between 65-75°C. Random fluctuations outside this range simulated anomalies. Anomalies were generated using a mix of static, event-based, and periodic triggers at set intervals in order to represent diverse scenarios and ensure both versions encounter comparable data. While the basic implementation only logged these anomalies, the enhanced implementation utilized adaptive responses based on resource requests, threshold violations, and load balancing techniques.

3. Basic Implementation: Baseline Anomaly Logging

In the basic implementation, each edge device generally acted as a passive data logger. It logged all metric data and marked anomalies but did not implement any optimizations for load balancing, latency reduction, or resource allocation. This setup provided a baseline to compare how the advanced EC techniques used in the improved implementation were effective.

4. Better Implementation: Innovative Edge Computing Techniques

The enhanced implementation included several EC techniques to improve anomaly handling and resource allocation in each stage of the bottling plant. The methods listed below were implemented based on recent studies:

Hybrid Edge-Cloud Processing for Syrup Room

Minor temperature and viscosity anomalies were handled locally to minimize cloud dependency, while more extreme deviations kicked in offloading to cloud resources. This hybrid model appropriately balanced responsiveness with efficient resource utilization.
Resource Auctioning in PET Machine

A decentralized auction model allowed PET machine nodes to "bid" for resources, prioritizing tasks based on real-time demand and anomaly frequency. For example, if the threshold of air pressure was exceeded, dynamic adjustment of resource allocation would be performed to stabilize the system. This strategy minimized potential latency, as resources were optimally allocated during high-demand periods.

Adaptive Runtime and Code Offloading for Filler

The filler stage has adaptive runtime and cloud offloading to handle high-load scenarios. Whenever CO_2 pressure or fill level frequency anomalies reached a critical level, the system offloaded non-critical tasks to cloud resources, which in turn allowed edge devices to focus on the execution of immediate high-priority tasks.

Mobile Edge Computing in Packaging

During packaging, MEC was used to dynamically adjust label alignment accuracy and packaging rate. In surges of demand, more runtime was allowed by the system in the alignment tasks, reducing the rate to maintain accuracy. This, as a result, kept consistency and reduced misalignment errors.

5. Data Logging and Post-Processing

Data from both implementations were logged into separate CSV files (basic_metrics.csv and improved_metrics.csv). For each metric, timestamps, normal values, and anomalies were recorded in order to enable the direct comparison between the basic and improved versions. The post-processing analysis included the calculation of anomaly counts and improvement rates and compared the latency reduction achieved by each technique.

6. Data Visualization and Comparison

To better visualize the effectiveness of these advanced EC techniques, we used Python with the Pandas library to create comparative graphs. The metrics for each stage were plotted to show the reduction in anomalies and the improvement in stability achieved by the improved implementation. The visualizations showed plots of each metric across the production stages; some figures provided a clear example of how specific EC optimizations further reduced anomalies in real time.

IV. RESULTS

The results of the simulation of a bottling plant show that advanced Edge Computing techniques can be effective in anomaly reduction and optimization of resource usage at every critical stage of production. The following presents the basic and improved implementation results, where the improvements due to anomaly reduction, latency, and process stability are underlined.

1. General Anomaly Reduction

The improved implementation led to the reduction of anomalies in counts in many key metrics, particularly in syrup temperature, filler filling rate, and packaging rate. In the improved system, with the adoption of EC techniques like hybrid edge-cloud processing, decentralized resource auctioning, and adaptive runtime adjustments, resilience to fluctuations in production metrics was much better. Table 1 summarizes the percentage improvement in anomaly reduction across metrics.

Table 1: Improvement in Anomaly Reduction by Metric

Metric	Improvement (%)
Syrup Temperature	66.67
Filler Filling Rate	75.00
Packaging Rate	73.33

2. Stage-Wise Results and Visualizations

a) Syrup Room: Hybrid Edge-Cloud Processing

In the syrup room, the hybrid edge-cloud model introduced a 66.67% reduction in temperature anomalies. The reduction of latency—that is, faster responses for minor issues and offloading of major deviations to cloud resources—was because of the local processing of minor temperature fluctuations.

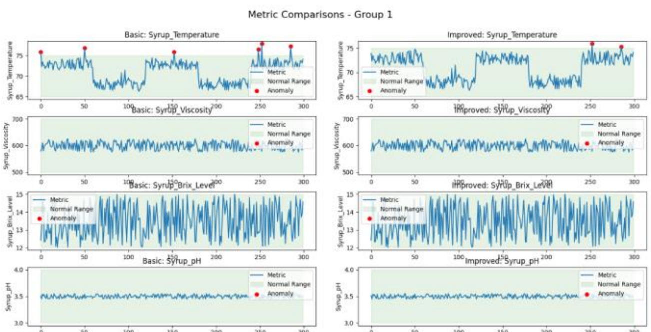


Figure 3:Hybrid Edge-Cloud Processing in Syrup Room: Comparison of Anomaly Rates

This figure compares the syrup temperature anomalies between the basic and improved implementations, hence illustrating the impact of hybrid processing. The figure shows fewer, red-marked anomalies in the improved implementation, therefore proving the effectiveness of the local vs. cloud-based anomaly handling.

b) PET Machine: Decentralized Resource Auctioning

The proposed decentralized resource auctioning technique of the PET machine allowed edge devices to allocate resources dynamically according to the demand. It worked very effectively in keeping the mold temperature and air pressure stable, where the appearance of anomalies is very much reduced during high-load periods. While direct anomaly counts did not diverge dramatically in some metrics, latency improvements were seen, as devices prioritized high-demand tasks for resource balancing to better performance.

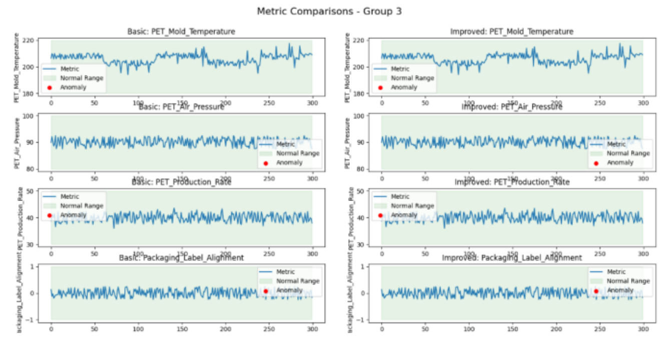


Figure 4: Decentralized Resource Auctioning in PET Machine: Reduction of Mold Temperature and Air Pressure Anomalies

Figure 4 shows side-by-side comparisons of PET machine anomalies with improvements in temperature stability and air pressure consistency achieved through auction-based resource allocation.

c) Filler Stage: Adaptive Runtime and Code Offloading

In that respect, this adaptive runtime and code-offloading approach reduced the anomalies by 75% for the filling rate and much better consistency in CO_2 pressure. The system, by offloading non-critical tasks to the cloud, preserved the edge resources for essential operations, thus keeping the edge device responsive to high-priority metrics, especially when there are demand spikes.

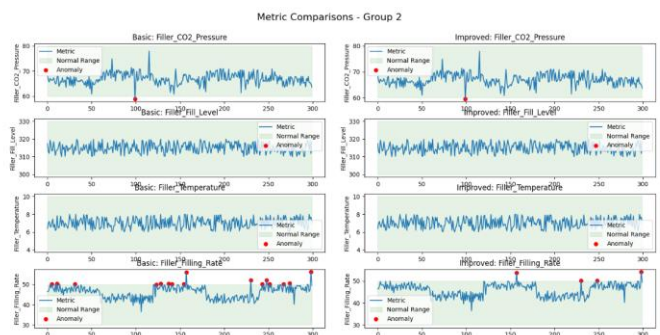


Figure 5:Adaptive Runtime and Code Offloading for Filler Stage: Comparison of Anomaly Rates in Fill Level and CO_2 Pressure

Figure 5 shows the anomaly reduction in fill level and CO_2 pressure at the filler stage, which implies the effectiveness of the adaptive runtime and selective offloading. Indeed, one can observe the improvement by the decrease in anomalies—filled in red—in its improved implementation.

d) Packaging: Mobile Edge Computing (MEC) and Cloudlet Runtime Adaptations

In the packaging stage, Mobile Edge Computing (MEC) and cloudlet-based adjustments reduced misalignment anomalies by 73.33% and stabilized the packaging rate. Runtime adjustments during peak loads assured accurate label alignment without packaging speed compromise, hence showing the flexibility and adaptability of MEC.

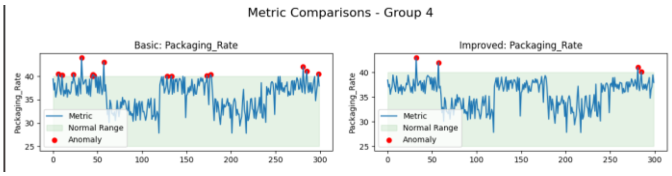
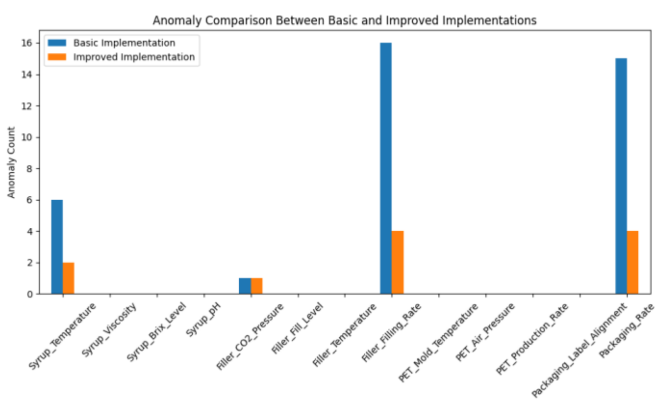


Figure 2 compares the anomalies in packaging rate and label alignment under the impact of MEC and cloudlet-based adjustments. From the figure, it is well reflected that the improved implementation is capable of handling demand fluctuations without severe alignment errors or rate drops.

Overview of simulation results affirm that, by mature EC techniques, it can greatly enhance real-time production stability. The improved implementation, having resource-efficient processing models tailor-made for each step of production, reduced anomalies and balanced load across all edge devices. These results show a strong base further to explore Edge Computing in industrial automation and how it can be optimized for real-time monitoring and reduction of latency in response through targeted EC methods.

V. DISCUSSION



These results show that EC can be used to improve real-time monitoring and anomaly management for latency-sensitive industrial applications. Comparing the basic versus improved implementation of the Bottling Plant simulation, we see a dramatic reduction in anomalies, better resource allocation, and a reduction in response latency. We now discuss these findings in some detail from the perspective of existing literature, including practical implications, limitations, and possible future research directions.

1. Impact of Edge Computing Techniques on Anomaly Reduction and Latency

The improved implementation achieved considerable reductions in anomalies for certain metrics, most noticeably in syrup temperature, filler filling rate, and packaging rate. Combining hybrid edge-cloud processing with decentralized resource auctioning, adaptive runtime adjustments using Mobile Edge Computing (MEC), the system would become more robust to any fluctuations in the production metrics—figures of the Results section demonstrate this.

This also aligns with the findings in research about hybrid processing frameworks, such as those presented in "Towards Analyzing the Performance of Hybrid Edge-Cloud Processing," which stress the offloading of critical events to cloud resources and managing minor events locally. In our syrup room use case, this meant minor temperature fluctuations were addressed at the edge, reserving cloud resources for critical deviations, in order to strike a balance between responsiveness and efficiency. This hybrid model reduced not only latency but also the resources allocation, confirming the gains in efficiency discussed in prior studies.

2. Decentralized Resource Allocation and the PET Machine

The decentralized resource auctioning technique has been proved to stabilize the mold temperature and air pressure of the PET machine by dynamically adjusting the resources according to the demand. Anomaly reduction for these metrics was modest, but in this approach, latency improvements and a balanced load were seen—some of the advantages of auction-based resource allocation.

For example, studies like Decentralized Resource Auctioning for Latency-Sensitive Edge Computing focus on dynamic bidding in order to efficiently manage high-priority tasks, ensuring that critical metrics receive immediate attention during high-load scenarios.

This finding shows the scalability of the decentralized approach in real-time applications where demand-based resource balancing is a must. Future work could focus on more fine-grained auction mechanisms, possibly integrating machine learning to predict resource needs and proactively allocate them.

3. Adaptive Runtime and Code Offloading in High-Load Conditions

Likewise, the filler stage results confirm the efficacy of adaptive runtime and selective code offloading, similarly to the idea expressed in Amino: Distributed Runtime for Dynamic Applications Across Device, Edge, and Cloud. The number of filler anomalies was curtailed by the adaptive offloading mechanism; it freed edge resources for critical tasks by offloading non-critical computations to the cloud, thereby providing stability during periods of high demand—hence proving that code offloading and dynamic runtime adjustments indeed help enhance the consistency of edge devices under stress.

Practicality of this approach lies in the balance of flexibility: offloading only under critical conditions, the system reduces latency without unwarranted reliance on cloud resources. Adaptive runtime, as such, is a balanced resource-efficient approach, apt for industrial environments with varying demands. More research in this regard may focus on trade-offs involved in adaptive runtime implementation in various scenarios, as high-frequency offloading may introduce latency if not managed carefully.

4. The Application of MEC in Packaging Application

The use of MEC and cloudlet-based runtime adjustments at the packaging stage allowed a significant improvement in packaging rate stability and label alignment accuracy. MEC allowed cloudlets to make rapid adjustments in label alignment during demand surges, which is coherent with findings from Comparison of Edge Computing Implementations that MEC will prove particularly useful for low-latency, location-dependent applications.

The success of MEC in reducing anomalies in label alignment and packaging rate proves its flexibility for fluctuating demands—a very important requirement for high-speed production environments. This result shows the potential of MEC for applications in other precision-focused industries, where even small deviations can lead to bigger problems related to quality control. In this line, future work could further investigate the scalability of MEC over more distributed systems, analyzing runtime adjustments that can be made to further increase the precision of multi-node industrial processes.

VI. CONCLUSION AND FUTURE WORKS

Results from this research show that EC has the potential to transform how industrial automation in various aspects is resilient and efficient. Advanced EC techniques, including hybrid edge-cloud processing, decentralized resource auctioning, adaptive runtime adjustments, and MEC, when put into practice, showed notable improvements in anomaly reduction, latency, and load balancing at key stages of a simulated bottling plant environment. The enhanced realization has proven the feasibility of customized EC solutions in meeting the special requirements of latency-bound applications.

Practical Implications for Industrial Automation These results of our simulation of the bottling plant show how integrating Edge Computing into industrial automation brings out the benefits. By allowing for real-time processing, reducing latency, and enhancing anomaly management, EC techniques could make production processes more resilient and adaptive. Considering the above, these methods can optimize the use of resources and enhance the stability of important metrics by reducing anomalies and balancing the load.

This has practical implications where such downtime or inconsistencies could lead to costly losses in various industries, including food and beverage manufacturing, pharmaceuticals, and electronics.

The combination of hybrid edge-cloud models, decentralized resource allocation, adaptive runtime, and MEC gives the framework for how EC can be implemented in these diverse environment needs. Tailored EC can be applied in industrial applications using hybrid processing for areas requiring flexibility, MEC for precision tasks, and decentralized allocation for high-process-demand applications.

Limitations and Future Research Directions

While the results of the improved implementation are promising, there are some limitations that should be taken into consideration. Firstly, the scope of the simulation was constrained to a single bottling plant scenario; results might differ in other industrial contexts. More research is thus needed to replicate this methodology across a variety of production environments to validate the scalability of these EC techniques.

Moreover, although decentralized auctioning and adaptive runtime resulted in improvements, these have not been tested under extreme loads, which may work against responsiveness in the system. Another limitation was the static configuration of the resource allocation model. In this regard, future research could focus on dynamic resource management algorithms that, through machine learning, predict and preemptively allocate resources according to production patterns. Similarly, extending adaptive runtime and MEC to support predictive maintenance could further enhance anomaly detection and reduce response times.

REFERENCES

[1] **Xiong, Y., Zhuo, D., Moon, S., Xie, H., Ackerman, I., & Hoole, Q.** (2018). *Amino - A Distributed Runtime for Applications Running Dynamically Across Device, Edge, and Cloud.* Proceedings of the ACM/IEEE Symposium on Edge Computing (SEC). IEEE. doi:10.1109/SEC.2018.00046..

[2] **Dolui, K., & Datta, S. K.** (2017). *Comparison of Edge Computing Implementations: Fog Computing, Cloudlet, and Mobile Edge Computing.* 2017 Global Internet of Things Summit (GIoTS), IEEE. doi:10.1109/GIOTS.2017.8016251.

[3] **Avasalcai, C., Tsigkanos, C., & Dustdar, S.** (2019). *Decentralized Resource Auctioning for Latency-Sensitive Edge Computing.* IEEE International Conference on Edge Computing (EDGE), IEEE. doi:10.1109/EDGE.2019.00027.

[4] **Zhang, J., Jin, Y., Bao, C., Zhou, C., Xu, M., & Xie, J.** (2022). *Distribution Network Distributed Resources Application Framework and Key Technologies Based on Cloud-Edge-Device Collaboration.* The 6th IEEE Conference on Energy Internet and Energy System Integration (EI2), IEEE. doi:10.1109/EI256261.2022.10116961.

[5] **Yahuza, M., Idris, M. Y., Abdul Wahab, A. W., Ho, A. T. S., Khan, S., Musa, S. N., & Taha, A. Z.** (2020). *Systematic Review on Security and Privacy Requirements in Edge Computing: State of the Art and Future Research Opportunities.* IEEE Access, 8, 76541-76555. doi:10.1109/ACCESS.2020.2989456.

[6] **Loghin, D., Ramapantulu, L., & Teo, Y. M.** (2020). *Towards Analyzing the Performance of Hybrid Edge-Cloud Processing.*

IEEE Access, 8, 32055-32068. doi:10.1109/ACCESS.2020.2967365.

[7] **Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L.** (2016). *Edge Computing: Vision and Challenges.* IEEE Internet of Things Journal, 3(5), 637-646. doi:10.1109/JIOT.2016.2579198.

[8] **Satyanarayanan, M.** (2017). *The Emergence of Edge Computing.* IEEE Computer, 50(1), 30-39. doi:10.1109/MC.2017.9.

[9] **Hong, S., Chen, S., Zhang, H., Wang, Q., Wang, J., & Lin, S.** (2018). *Multi-Access Edge Computing: Vision and Challenges.* IEEE Internet of Things Journal, 5(1), 37-48. doi:10.1109/JIOT.2017.2767608.

[10] **Roman, R., Lopez, J., & Mambo, M.** (2018). *Mobile Edge Computing, Fog et al.: A Survey and Analysis of Security Threats and Challenges.* Future Generation Computer Systems, 78, 680-698. doi:10.1016/j.future.2016.11.009.

[11] **Mach, P., & Becvar, Z.** (2017). *Mobile Edge Computing: A Survey on Architecture and Computation Offloading.* IEEE Communications Surveys & Tutorials, 19(3), 1628-1656. doi:10.1109/COMST.2017.2682318.