ISSN: 2959-6386 (Online), Vol. 2, Issue 2

Journal of Knowledge Learning and Science Technology

journal homepage: https://jklst.org/index.php/home

Dynamic Resource Allocation in Edge Computing for AI/ML Applications: Architectural Framework and Optimization Techniques

Lavanya Shanmugam[1], Suhas Jangoan[2], Kapil Kumar Sharma[3]

[1]Tata Consultancy Services, USA
[2]Zendesk, USA
[3]Cisco, USA

Abstract

This research article proposes a comprehensive architectural framework and optimization techniques for dynamic resource allocation in edge computing environments specifically tailored for AI/ML applications. Edge computing has emerged as a promising paradigm for handling the computational demands of AI/ML tasks by leveraging resources closer to data sources. However, effective resource allocation poses significant challenges due to the heterogeneity and dynamic nature of edge environments. In response, this paper presents a novel framework that integrates dynamic resource allocation strategies with AI/ML application requirements. The proposed framework encompasses various optimization techniques tailored to efficiently allocate resources, considering factors such as workload characteristics, resource availability, and latency constraints. Through extensive simulations and evaluations, we demonstrate the efficacy of the proposed approach in improving resource utilization, minimizing latency, and enhancing overall performance for AI/ML workloads in edge computing scenarios.

Keywords: Edge Computing, Resource Allocation, AI/ML Applications, Architectural Framework, Optimization Techniques.

Article Information:
Article history: *Received:* 01/10/2023 *Accepted:* 10/10/2023 *Online:* 16/10/2023 *Published*: 16/10/2023
DOI: https://doi.org/10.60087/jklst.vol2.n2.p397

Correspondence Author: Lavanya Shanmugam

Introduction

In recent years, the convergence of Artificial Intelligence (AI) and Machine Learning (ML) with edge computing has transformed the landscape of computing paradigms. Edge computing, characterized by its proximity to data sources and end-users, offers unparalleled opportunities for enhancing the efficiency, responsiveness, and scalability of AI/ML applications. However, leveraging the full potential of AI/ML at the edge requires sophisticated resource allocation mechanisms to address the challenges posed by limited computational resources, heterogeneous environments, and dynamic workloads.

This introduction provides an overview of dynamic resource allocation in edge computing for AI/ML applications, focusing on the development of an architectural framework and optimization techniques to effectively manage computational resources at the edge.

1. Contextualizing Edge Computing: The proliferation of Internet of Things (IoT) devices, coupled with the demand for real-time data processing and low-latency applications, has fuelled the adoption of edge computing. By distributing computational tasks closer to data sources, edge computing reduces latency, bandwidth usage, and reliance on centralized cloud infrastructure.

2. The Role of AI/ML in Edge Computing: AI/ML algorithms are increasingly deployed at the edge to extract actionable insights from vast volumes of data generated by IoT devices. These applications span various domains, including smart cities, healthcare, industrial automation, autonomous vehicles, and more. However, deploying AI/ML models at the edge presents unique challenges related to resource constraints, energy efficiency, and scalability.

3. Challenges in Resource Allocation: Dynamic resource allocation in edge computing involves dynamically provisioning computational resources, such as CPU, GPU, memory, and storage, to accommodate varying workloads and application requirements. Key challenges include resource contention, heterogeneity of edge devices, fluctuating network conditions, and the need to optimize resource utilization while meeting Quality of Service (QoS) constraints.

4. Architectural Framework for Dynamic Resource Allocation: A robust architectural framework is essential for

orchestrating resource allocation in edge computing environments. This framework should encompass components for workload monitoring, resource provisioning, decision-making, and enforcement mechanisms. Moreover, it should support flexibility, scalability, and adaptability to evolving edge environments.

5. Optimization Techniques: Various optimization techniques, including heuristic algorithms, machine learning-based approaches, and game theory, can be employed to optimize resource allocation in edge computing. These techniques aim to maximize resource utilization, minimize latency, energy consumption, and operational costs, and ensure QoS guarantees for AI/ML applications running at the edge.

In summary, dynamic resource allocation plays a pivotal role in unlocking the full potential of AI/ML applications at the edge. This introduction sets the stage for exploring the architectural principles, optimization techniques, and practical considerations involved in designing efficient resource allocation mechanisms tailored to the unique characteristics of edge computing environments.

objectives

1.Developing a Scalable Architectural Framework: The first objective is to design and develop a scalable architectural framework for dynamic resource allocation in edge computing environments. This framework should encompass components for real-time workload monitoring, adaptive resource provisioning, decision-making algorithms, and enforcement mechanisms to effectively manage computational resources at the edge.

2.Optimizing Resource Utilization: The second objective focuses on optimizing resource utilization while ensuring Quality of Service (QoS) for AI/ML applications running at the edge. This involves leveraging optimization techniques such as heuristic algorithms, machine learning-based approaches, and game theory to dynamically allocate CPU, GPU, memory, and storage resources based on workload characteristics, device capabilities, and network conditions.

3.Enhancing Performance and Efficiency: The third objective aims to enhance the performance and efficiency of AI/ML applications deployed at the edge by minimizing latency, energy consumption, and operational costs. This involves fine-tuning resource allocation policies, adapting to changing workload patterns, and dynamically scaling resources to meet fluctuating demand, ultimately improving the overall responsiveness and user experience of edge computing systems.

Literature Review

Dynamic resource allocation in edge computing for AI/ML applications involves optimizing task offloading and resource allocation efficiently [1] [2]. Various challenges like low scalability and high training costs exist, prompting the need for novel approaches. One such approach involves a link-output Graph Neural Network (LOGNN) for flexible resource management with low algorithm inference delay [3]. Additionally, a cloud-edge-end computing architecture is proposed to handle multi-source data streams efficiently, utilizing a combination of proximal policy optimization and convex optimization for resource allocation [4]. Furthermore, a configurable model deployment architecture (CMDA) is introduced for edge AIaaS, enabling joint configuration of data quality ratios and model complexity ratios to enhance energy and delay performance of AI services [5]. These frameworks and optimization techniques aim to improve resource utilization and performance in edge computing for AI/ML applications.

Edge Computing

The rise of the Internet of Things (IoT) has spurred the creation and deployment of a vast array of hardware devices and sensors on a global scale. These devices possess the capability to perceive their surrounding physical environment and convert this environmental data into actionable information. Subsequently, this wealth of data is typically transmitted to centralized cloud servers for processing or storage, allowing data consumers to access and extract pertinent information tailored to their individual needs [3].

However, as IoT continues to evolve and expand in its applications, cloud computing has begun to reveal increasingly prevalent challenges. For instance, when data generated by global terminal devices undergo computation and storage within centralized cloud infrastructure, it can lead to a host of issues including diminished throughput, heightened latency, bandwidth constraints, data privacy concerns, centralized vulnerabilities, and added expenses such as transmission, energy, storage, and computational costs. Notably, numerous IoT application scenarios, particularly within the realm of the Internet of Vehicles (IoV), necessitate swift data processing, analysis, and response, demanding high speed and minimal latency [4].

In response to the limitations of traditional cloud computing highlighted above, a novel computing paradigm known as edge computing (EC) has garnered considerable attention. In essence, the core tenet of the EC model is to offload the data processing, storage, and computing tasks originally entrusted to centralized clouds to the network's edge, in close proximity to terminal devices. This approach serves to alleviate data transmission delays and device response times, mitigate strain on network bandwidth, reduce the overhead associated with data transmission, and promote decentralization [5].

Artificial Intelligence

Artificial intelligence (AI) represents a technological advancement that imbues machines with cognitive capabilities, enabling them to perform tasks akin to human beings [6]. While heuristic-based algorithms and data mining (DM) have historically been pivotal in AI solutions for IoT, our focus primarily lies on machine learning (ML), an increasingly popular domain within AI. It's noteworthy that while DM and ML share similarities in leveraging vast datasets, ML specifically aims to emulate the human learning process, whereas DM is geared towards extracting rules from data [7, 8, 9]. ML, being a higher-level intelligence, represents the future trajectory of AI.

The widespread adoption of AI, particularly ML, has become an inexorable trend in the "big data era" catalyzed by IoT. It's important to highlight that this discussion centers on cutting-edge AI algorithms like deep learning (DL) and others. Notably, certain applications within this domain necessitate stringent requirements for latency and network stability, criteria often unmet by conventional cloud computing. In contrast, the burgeoning EC model can address these needs by deploying AI at the edge and allocating computing and storage resources to edge devices situated close to terminals. While EC offers advantages such as reduced latency, enhanced data privacy, and bolstered security, the finite computing and storage capacities of edge devices introduce new challenges. Leveraging AI to optimize EC and address its associated issues has emerged as a pivotal trend in related research [10].

Combination of Edge Computing and Artificial Intelligence

The integration of Artificial Intelligence (AI) and Edge Computing (EC) in recent research is driven by two primary motivations, illustrating the symbiotic relationship between these two domains:

1. Addressing Challenges in EC Development: The advancement of EC encounters numerous challenges such as task scheduling, resource allocation, delay optimization, energy consumption optimization, and privacy and security concerns. In response, many researchers have turned to AI-based solutions to foster progress in EC development.

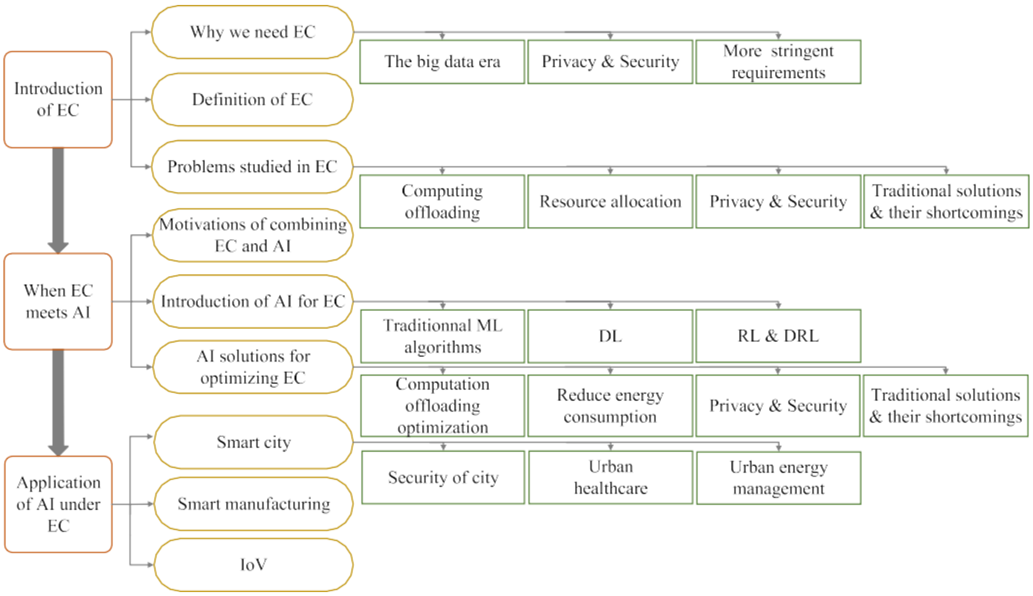
2. Enhancing AI Applications: Despite the rapid evolution of AI, its effective application relies heavily on robust computing power. While traditional cloud computing offers ample computing and storage resources, relying on cloud-based AI reasoning and training can introduce significant delays and raise privacy and security issues. By executing AI tasks in edge nodes situated closer to end-users, EC can effectively mitigate these challenges, enhancing stability, reliability, and user experience.

Currently, researchers have made significant strides in addressing these research challenges. This article aims to consolidate and summarize these achievements, providing readers with updated insights into the latest research status and relevant outcomes.

Review of Existing Surveys

Edge Computing (EC) and Artificial Intelligence (AI) represent burgeoning research domains, with several pertinent reviews already published. In Reference [11], the authors delve into the motivations and research endeavors surrounding the deployment of AI algorithms at the network edge. Reference [12] provides a comprehensive overview of the latest advancements in Machine Learning (ML) within mobile EC, encompassing developments in 5G networks, automatic adaptive resource allocation, mobility modeling, security, and energy efficiency. Survey [13] explores the application of Deep Learning (DL) in EC, spotlighting its role in fostering the advancement of edge applications such as intelligent multimedia, transportation, cities, and industries. Furthermore, Reference [14] examines various techniques for swiftly implementing DL reasoning across end devices, edge servers, and the cloud, along with strategies for training DL models across multiple edge devices. To optimize DL training and reasoning performance, Reference [15] offers an in-depth discussion on designing EC architectures considering communication, computational power, and energy consumption constraints.

Despite the abundance of research, the synergistic relationship between EC and AI, particularly traditional ML, DL, reinforcement learning (RL), and deep reinforcement learning (DRL), has received limited attention in prior surveys. Hence, this article fills the gap by reviewing existing works on EC performance optimization and various AI application scenarios. In addition to the DL methodologies explored in References [13–15], this article also delves into other ML algorithms, notably RL and DRL, broadening the discourse on the intersection of EC and AI.



Our Contributions

The structure of the survey is depicted in Fig. 1.

Our main contributions in this article are as follows:

1. We begin by providing an overview of the fundamental definition and architecture of Edge Computing (EC) and elaborate on the necessity of EC alongside cloud computing. Furthermore, we delineate the challenges investigated within the domain of EC.

2. We delve into the motivations behind integrating Artificial Intelligence (AI) and EC from two distinct perspectives: (a) leveraging AI algorithms to optimize EC, and (b) employing EC to facilitate the deployment of AI at the edge, thereby enhancing response times and network stability for AI applications across various domains. Additionally, we summarize three approaches for deploying AI training and reasoning tasks within the EC architecture, drawing insights from existing studies, and assess their respective advantages and limitations.

3. We predominantly introduce popular Machine Learning (ML) algorithms within the AI domain and analyze their individual strengths. Furthermore, we synthesize the latest research efforts aimed at addressing EC challenges and optimizing EC performance through the utilization of AI algorithms. Additionally, we review recent advancements in applying AI to various other domains within the EC framework.

Roadmap:
The subsequent sections of this article are structured as follows:
- Section 2 introduces the definition of EC, explores the rationale behind its necessity, and outlines the challenges encountered by EC along with traditional (non-AI) solutions.
- In Section 3, we merge EC and AI. We discuss the trends and motivations driving the integration of these two domains, introduce relevant AI algorithms, and comprehensively review research endeavors aimed at leveraging AI algorithms to optimize EC.
- Section 4 summarizes recent efforts in applying AI to other domains within the EC framework.
- Finally, we conclude this article in Section 5. Figure 1 provides a visual representation of the article's structure.

Introduction to Edge Computing

Cloud computing has become ubiquitous over the past decade, offering myriad conveniences to businesses, particularly small and medium-sized enterprises. These enterprises can access cloud server resources at relatively low costs, bypassing the need to invest heavily in hardware and equipment. This significantly reduces operational expenses and lowers the barriers for companies to engage in technology research and development.

However, the centralized nature of cloud computing, encompassing computing, storage, and network resources, has revealed several drawbacks over time. In response, Edge Computing (EC), a novel computing paradigm, has begun to garner attention across various sectors. In this section, we provide a concise overview of EC, delineating its necessity, defining its core concepts, and highlighting associated challenges along with traditional solutions, while also pinpointing their limitations.

Why We Need Edge Computing

The necessity of EC can be elucidated from three key perspectives:

1. The Big Data Era Caused by the Internet of Things (IoT):

The inception of the Internet of Things (IoT) dates back to 1999, initially proposed for supply chain management. However, IoT has since expanded its reach into various industries, spawning new applications such as smart homes, grids, traffic systems, and manufacturing. With IoT's integration into traditional industries, an exponential increase in global data volume is anticipated, projected to reach 175 zettabytes (ZB) by 2025 according to the International Data Corporation (IDC) [18]. In this era of big data, the conventional method of transferring data to the cloud for processing is becoming less viable due to the cloud's linearly increasing computing power, which lags behind the rapid growth of data.

2. More Stringent Requirements of Network Stability and Response Speed:

Certain IoT applications necessitate exceptionally fast response times. For instance, in autonomous driving scenarios, sensors continuously gather data from the vehicle's surroundings. Uploading this data to the cloud for processing and awaiting results back to the vehicle's control chip can lead to significant delays, potentially jeopardizing timely decision-making and resulting in adverse outcomes. Similarly, augmented reality (AR) and virtual reality (VR) applications demand high-resolution video transmission, imposing rigorous requirements on data computing capabilities, network stability, and response speed. However, the current pace of data growth renders the cloud's computing power insufficient to meet these demands.

3. Privacy and Security Concerns:

Cloud computing's outsourcing features necessitate users to entrust local data to the cloud, raising pertinent issues related to data security and privacy. Data loss during long-distance transmission between devices and the cloud can compromise data integrity and accuracy. Moreover, highly centralized computing and storage architectures pose significant risks, wherein errors or malicious attacks affecting one device can propagate to others. Data privacy concerns arise from unauthorized access and utilization by external entities, as data owners relinquish control over their uploaded data, thereby challenging data privacy assurances.

In summary, the advent of Edge Computing arises from the limitations of traditional cloud computing in addressing the burgeoning data volumes, stringent requirements for network stability and response speed, and escalating privacy and security concerns. Edge Computing offers a promising alternative by decentralizing computational resources, enhancing responsiveness, and bolstering data privacy and security measures.

The genesis of Edge Computing (EC) can be traced back to 1999 when Akamai introduced content delivery networks (CDN) for caching web pages closer to clients, with the aim of enhancing web page loading efficiency. The concept of EC was derived from cloud computing infrastructure, expanding upon the principles of CDN.

EC encompasses various definitions. For instance, OpenStack defines EC as a model providing cloud services and IT environmental services to application developers and service providers at the network's edge [27]. In Reference [28], the "edge" in EC is interpreted as any computing and network resources situated between the data source and the cloud, including smartphones, gateways, micro data centers, and cloud networks. Conceptually, EC involves offloading certain cloud resources and tasks to the edge, closer to users and data sources.

It's imperative to recognize that EC does not aim to supplant the roles and advantages of cloud computing, given the indispensable computing power and storage capacity of the cloud. Rather, EC emerges to address the limitations of cloud computing, necessitating a complementary relationship between EC and cloud computing. Consequently, exploring methods to optimize the collaboration between the cloud and the edge, ensuring efficient and secure cooperation, becomes a pertinent area for further study.

The general architecture of EC is typically structured into three layers, as depicted in Figure 2:

1. End: This layer serves two primary functions. Firstly, it perceives the physical world by observing, acquiring, and digitizing information from various sensors, such as speed sensors in smart cars or cameras in smart cities. Secondly, it receives information or data from the edge or cloud and executes corresponding tasks. Data from the end undergo processing by the edge and the cloud before being fed back to the end based on user requirements, such as control signals in smart driving or video traffic received by smartphones. Devices in this layer may possess limited computing and storage capabilities.

2. Edge: Positioned between the cloud and the end, this layer houses specific computing, storage, and network resources. Tasks originally performed in the cloud can be delegated to this layer for execution. Being closer to end devices, EC at the edge offers the advantage of low latency. Typically, the edge layer comprises gateways, control units, storage units, and computing units.

3. Cloud: This layer denotes the cloud servers widely employed in practical scenarios. Apart from its robust computing and storage capabilities, the cloud possesses the capacity for macro-control over the entire EC architecture.

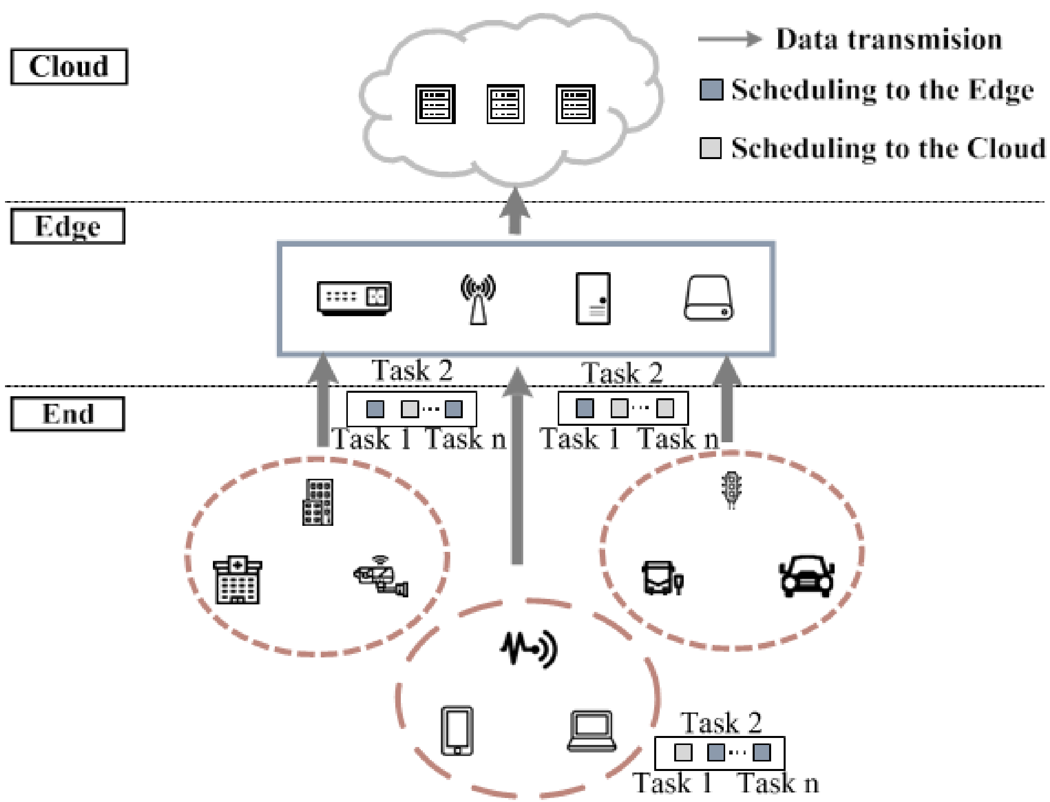


Figure 2 illustrates the Architecture of Edge Computing (EC). Gray arrows represent data transmission between the end, the edge, and the cloud. Blue and gray boxes indicate tasks scheduled to the edge and the cloud, respectively. Edge Computing (EC) offers several advantages by offloading certain resources and tasks from the cloud to the edge. The edge layer's proximity to end users and data sources significantly shortens transmission distances, thereby reducing transmission times and enhancing response speeds to user requests. Simultaneously, the shortened transmission distance mitigates the costs and data security concerns associated with long-distance transmission.

From the cloud's perspective, large-scale raw data undergoes initial processing at the edge to filter out irrelevant and erroneous data. Subsequently, the edge uploads pertinent data or information to the cloud. This approach effectively alleviates bandwidth pressure, minimizes transmission costs, and reduces the risk of user privacy breaches.

Challenges Addressed in Edge Computing

In the subsequent discussion, we delve into three key challenges prevalent in the realm of Edge Computing (EC): computing offloading, resource allocation, and privacy and security concerns. Additionally, we elucidate the limitations of conventional approaches in tackling these issues.

1. Computing Offloading:

Originally proposed in cloud computing, computation offloading involves terminal devices with limited computing power delegating part or all of their computing tasks to the cloud for execution. Similarly, in EC, computing offloading pertains to the scenario where terminal devices delegate their computing tasks to the edge. This entails considerations such as determining whether terminal devices will offload, the extent of offloading, and the designated nodes for offloading. Computing offloading addresses challenges related to insufficient resources and high energy consumption in terminal devices.

Traditional methods of computing offloading, rooted in cloud computing, assume that the default server possesses ample computing power and disregards concerns regarding energy consumption or network conditions. However, these assumptions are unsuitable for solving computing offloading challenges in EC, where edge devices and servers have limited computing capabilities. Therefore, devising rational computing offloading strategies is imperative for reducing energy consumption and latency, making it a pivotal research area for optimizing EC.

2. Resource Allocation:

A notable advantage of EC over traditional cloud computing is its ability to distribute tasks across edge nodes, thus alleviating the need to upload all data to the cloud for computing and storage. This significantly liberates network bandwidth and other resources typically monopolized by cloud computing. However, efficient resource management solutions are essential due to the distributed nature of tasks across edge nodes with limited resources.

3. Privacy and Security:

EC introduces novel challenges concerning data security and privacy. Some of these challenges stem from inherent issues in cloud computing, while others arise from the distributed and heterogeneous nature of EC itself. Conventional solutions for addressing data security and privacy concerns in cloud computing are not directly applicable to the decentralized computing model of EC. Hence, enhancing data security and privacy protection in EC warrants further attention from researchers.

Conclusions

While traditional methods have made commendable strides in addressing resource allocation, computing offloading, and security concerns in EC, they still exhibit certain shortcomings. These include a reliance on known underlying models, susceptibility to local optima convergence, and limited capacity for deep and high-dimensional data mining. Conversely, AI algorithms possess the potential to overcome these limitations, as they excel in adaptability, feature extraction, decision optimization, and prediction. The subsequent section will elucidate how AI algorithms optimize EC in light of these challenges.

This section provides insights into the conceptual framework and motivations driving EC while highlighting the obstacles encountered in its development. Although traditional methods have achieved notable success in tackling these issues, there remains room for improvement. In the future, AI algorithms are poised to offer enhanced adaptability and efficiency in addressing evolving challenges within EC, particularly with abundant data and dynamic constraints.

References

[1] A.U.R. Khan, M. Othman, S.A. Madani, and S.U. Khan. 2014. Exploring Application Models in Mobile Cloud Computing: A Survey. IEEE Communications Surveys & Tutorials. 16, 1 (2014), 393–413.

[2] F. Durao, F. Carvalho, A. Fonseka, and V.C. Garcia. 2014. A Systematic Review on Cloud Computing: Trends, Challenges, and Future Directions. Journal of Supercomputing. 68, 3 (2014), 1321–1346.

[3] W. Shi and S. Dustdar. 2016. Unveiling the Potential of Edge Computing: A Perspective. Computer. 49, 5 (2016), 78–81.

[4] M. Qin, L. Chen, N. Zhao, Y. Chen, F.R. Yu, and G. Wei. 2018. Power-Constrained Edge Computing with Maximum Processing Capacity for IoT Networks. IEEE Internet of Things Journal. 6, 3 (2018), 4330–4343.

[5] A.M. Ghosh and K. Grolinger. 2021. Edge-Cloud Computing for Internet of Things Data Analytics: Embedding Intelligence in the Edge with Deep Learning. IEEE Transactions on Industrial Informatics. 17, 3 (2021), 2191–2200.

[6] P. Zhou, W. Chen, S. Ji, H. Jiang, L. Yu, and D. Wu. 2019. Privacy-Preserving Online Task Allocation in Edge-Computing-Enabled Massive Crowdsensing. IEEE Internet of Things Journal. 6, 5 (2019), 7773–7787.

[7] E.I. Gaura et al. 2013. Edge Mining in the Internet of Things: A Survey. IEEE Sensors Journal. 13, 10 (2013), 3816–3825.

[8] Z. Xue et al. 2020. Artificial Intelligence for Securing IoT Services in Edge Computing: A Comprehensive Survey. Security and Communication Networks. 2020 (2020), 1–13.

[9] C. Savaglio and G. Fortino. 2021. A Simulation-Driven Methodology for IoT Data Mining Based on Edge Computing. ACM Transactions on Internet Technology. 21, 2 (2021), 1–22.

[10]. Schumaker, R. P., Veronin, M. A., Rohm, T., Boyett, M., & Dixit, R. R. (2021). A data driven approach to profile potential SARS-CoV-2 drug interactions using TylerADE. *Journal of International Technology and Information Management*, *30*(3), 108-142.
DOI: https://doi.org/10.58729/1941-6679.1504
[11]. Schumaker, R., Veronin, M., Rohm, T., Dixit, R., Aljawarneh, S., & Lara, J. (2021). An Analysis of Covid-19 Vaccine Allergic Reactions. *Journal of International Technology and Information Management*, *30*(4), 24-40.
DOI: https://doi.org/10.58729/1941-6679.1521

[12]. Dixit, R. R., Schumaker, R. P., & Veronin, M. A. (2018). A Decision Tree Analysis of Opioid and Prescription Drug Interactions Leading to Death Using the FAERS Database. In *IIMA/ICITED Joint Conference 2018* (pp. 67-67). INTERNATIONAL INFORMATION MANAGEMENT ASSOCIATION.

https://doi.org/10.17613/1q3s-cc46
[13]. Veronin, M. A., Schumaker, R. P., Dixit, R. R., & Elath, H. (2019). Opioids and frequency counts in the US Food and Drug Administration Adverse Event Reporting System (FAERS) database: A quantitative view of the epidemic. *Drug, Healthcare and Patient Safety*, 65-70.
https://www.tandfonline.com/doi/full/10.2147/DHPS.S214771

[14]. Veronin, M. A., Schumaker, R. P., & Dixit, R. (2020). The irony of MedWatch and the FAERS database: an assessment of data input errors and potential consequences. *Journal of Pharmacy Technology*, *36*(4), 164-167.
https://doi.org/10.1177/8755122520928

[15]. Veronin, M. A., Schumaker, R. P., Dixit, R. R., Dhake, P., & Ogwo, M. (2020). A systematic approach to'cleaning'of drug name records data in the FAERS database: a case report. *International Journal of Big Data Management*, *1*(2), 105-118.
https://doi.org/10.1504/IJBDM.2020.112404

[16]. Schumaker, R. P., Veronin, M. A., & Dixit, R. R. (2022). Determining Mortality Likelihood of Opioid Drug Combinations using Decision Tree Analysis.
https://doi.org/10.21203/rs.3.rs-2340823/v1

[17]. Schumaker, R. P., Veronin, M. A., Dixit, R. R., Dhake, P., & Manson, D. (2017). Calculating a Severity Score of an Adverse Drug Event Using Machine Learning on the FAERS Database. In *IIMA/ICITED UWS Joint Conference* (pp. 20-30). INTERNATIONAL INFORMATION MANAGEMENT ASSOCIATION.

[18]. Dixit, R. R. (2018). Factors Influencing Healthtech Literacy: An Empirical Analysis of Socioeconomic, Demographic, Technological, and Health-Related Variables. *Applied Research in Artificial Intelligence and Cloud Computing*, *1*(1), 23-37.

[19]. Dixit, R. R. (2022). Predicting Fetal Health using Cardiotocograms: A Machine Learning Approach. *Journal of Advanced Analytics in Healthcare Management*, *6*(1), 43-57.
Retrieved from https://research.tensorgate.org/index.php/JAAHM/article/view/38

[20]. Dixit, R. R. (2021). Risk Assessment for Hospital Readmissions: Insights from Machine Learning Algorithms. *Sage Science Review of Applied Machine Learning*, *4*(2), 1-15.
Retrieved from https://journals.sagescience.org/index.php/ssraml/article/view/68

[21]. Ravi, K. C., Dixit, R. R., Singh, S., Gopatoti, A., & Yadav, A. S. (2023, November). AI-Powered Pancreas Navigator: Delving into the Depths of Early Pancreatic Cancer Diagnosis using Advanced Deep Learning Techniques. In *2023 9th International Conference on Smart Structures and Systems (ICSSS)* (pp. 1-6). IEEE.
https://doi.org/10.1109/ICSSS58085.2023.10407836

[22]. Khan, M. S., Dixit, R. R., Majumdar, A., Koti, V. M., Bhushan, S., & Yadav, V. (2023, November). Improving Multi-Organ Cancer Diagnosis through a Machine Learning Ensemble Approach. In *2023 7th International Conference on Electronics, Communication and Aerospace Technology (ICECA)* (pp. 1075-1082). IEEE.

https://doi.org/10.1109/ICECA58529.2023.10394923

[23]. Sarker , M. (2023). Assessing the Integration of AI Technologies in Enhancing Patient Care Delivery in U.S. Hospitals. *Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online)*, *2*(2), 338-351. https://doi.org/10.60087/jklst.vol2.n2.p351