ADVANCES IN CYBER-PHYSICAL SYSTEMS

Vol. 9, No. 2, 2024

CONTAINERIZED ARTIFICIAL INTELLIGENT SYSTEM DESIGN IN CLOUD AND CYBER-PHYSICAL SYSTEMS

Yevhen Bershchanskyi[1], Halyna Klym[1], Yurii Shevchuk[2]

[1]*Lviv Polytechnic National University, 12, Bandera Str, Lviv, 79013, Ukraine,*
[2]*DataArt, 3530 Carol Ln, Northbrook, Illinois 60062, USA.*
Authors' e-mails*: yevhen.v.bershchanskyi@lpnu.ua, halyna.i.klym@lpnu.ua, shev4ukyuri@gmail.com*

https://doi.org/10.23939/acps2024.02.151

Submitted on 14.10.2024

***Abstract*: The integration of Artificial Intelligence (AI) into cloud computing and Cyber-Physical Systems (CPS) is crucial for achieving efficiency, scalability, and real-time capabilities in modern ecosystems. Containerization enhances AI deployment by improving portability, resource efficiency, and system isolation. This article addresses key design considerations and challenges in implementing containerized AI within cloud-native and CPS environments, focusing on scalability, fault tolerance, real-time responsiveness, and security. Through research analysis and case studies, it explores strategies for optimizing AI workload distribution across cloud and edge infrastructure to meet CPS demands. Future directions, including hybrid architectures and federated learning, are also discussed to support scalable, secure, and reliable AI systems for next-generation cloud and CPS applications.**

***Index Terms*: Cloud Containers, AI Model Containerization, CPS, AI Systems, Cloud-Native AI Design.**

I. INTRODUCTION

Artificial Intelligence, cloud computing, and Cyber-Physical Systems are rapidly transforming the technological landscape across various industries. AI has advanced from being a theoretical field to a driving force behind automation, decision-making, and data analytics, offering significant value in areas such as healthcare, finance, manufacturing, and autonomous systems. Meanwhile, cloud computing has enabled businesses to access on-demand computing resources and scaling operations without the need for substantial infrastructure investments. These advancements in AI and cloud computing have been particularly instrumental in the development of CPS, where physical systems are integrated with computation and communication technologies to create intelligent, responsive environments. Cyber-Physical Systems–comprising applications like autonomous vehicles, smart grids, and industrial IoT–rely heavily on real-time processing and AI-driven decision-making. The integration of AI within CPS enables predictive analytics, autonomous control, and enhanced system efficiency, making CPS a cornerstone of the next industrial revolution. As AI continues to play an increasing role in both cloud environments and CPS, the need to optimize its deployment and execution becomes paramount, especially in terms of scalability, reliability, and security.

Containerization has emerged as a crucial technology for the efficient deployment of AI systems, especially in cloud-native environments and CPS architectures [1]. In a distributed system where AI workloads span across cloud infrastructure and edge devices, achieving consistency and resource optimization is a complex challenge. Containers, by encapsulating an AI model along with its dependencies into lightweight, portable packages, address this challenge by offering a standardized environment for AI deployment across various platforms.

The benefits of containerization include portability, allowing AI systems to be moved seamlessly between different computing environments, whether in the cloud or at the edge. Additionally, containers enable fine-grained resource isolation, ensuring that individual AI processes do not interfere with each other, thus improving overall system stability. Scalability is another critical advantage, as containers can be orchestrated to automatically scale in response to fluctuating AI workloads, making them ideal for environments that require elastic resource allocation.

Furthermore, research has focused on the adoption of microservices architectures in AI deployment, where each AI function is treated as an independent service. This modular approach simplifies the management of AI models and facilitates continuous updates. However, despite the benefits of containerized AI, challenges such as resource contention, dependency management, and cross-environment integration remain. AI models are becoming more complex and managing their deployment across distributed systems while ensuring optimal performance is a persistent issue [2]. In CPS, where real-time performance is crucial, containers support low-latency AI operations and facilitate the distribution of processing tasks between the cloud and edge devices [3].

The purpose of this article is to explore the design principles and challenges of deploying containerized AI systems in cloud and CPS environments. As AI systems become increasingly critical in modern industries, their performance, reliability, and security need to be ensured through optimized deployment strategies. This work investigates the role of containerization in enhancing

these aspects by providing a framework for developing AI-driven systems that are both scalable and secure.

The growing reliance on real-time AI in CPS necessitates architectures that can handle the complexities of distributed systems [4]. Containerized AI offers a solution by enabling flexible, reliable, and secure deployment models that address the unique demands of cloud-based and CPS applications. This article aims to provide research on AI system design and how containerization can optimize AI performance while maintaining the integrity and security of complex, distributed systems.

II. LITERATURE REVIEW AND PROBLEM STATEMENT

Containerization has become a key technology in the deployment of AI systems within cloud infrastructures, offering significant advantages such as portability, resource efficiency, and scalability. Research has extensively explored the role of containerized AI, particularly through the use of technologies like Docker and Kubernetes. Docker allows AI models and their dependencies to be encapsulated into lightweight containers, ensuring consistent deployment across various platforms. This consistency is crucial for the effective deployment of AI in cloud environments, where the isolation of resources and standardized execution environments are critical.

Kubernetes, a widely adopted orchestration platform, extends these capabilities by automating the scaling, deployment, and management of containerized applications. The orchestration of AI workloads on Kubernetes supports cloud-native architectures by allowing elasticity, which ensures that AI systems can scale to meet dynamic computational demands. Several studies have demonstrated Kubernetes' ability to enhance the performance and efficiency of AI applications by enabling automated scaling, fault tolerance, and high availability.

The integration of AI into CPS has the potential to revolutionize industries through real-time decision-making, predictive analytics, and autonomous operations. CPS applications, including autonomous vehicles, smart grids, and industrial IoT devices, rely on AI to analyze environmental data and drive system responses. Existing literature underscores the importance of AI in improving the efficiency, safety, and intelligence of CPS (Fig.1). For example, autonomous vehicles use AI for navigation and obstacle avoidance, while smart grids utilize AI for energy optimization and predictive maintenance.

However, CPS environments introduce unique constraints that complicate AI deployment. Real-time processing is critical in CPS, but achieving low-latency AI decision-making is challenging due to the limited computational resources available in many CPS applications. Research on edge computing strategies–where AI processing is distributed between the cloud and local edge devices–has demonstrated potential solutions to reduce latency and bandwidth requirements. Still, balancing real-time AI performance with the resource limitations of CPS is a major technical hurdle.

In addition to performance constraints, CPS environments present heightened security challenges. Many CPS applications are vulnerable to cyberattacks due to the distributed nature of the system and the sensitivity of real-time data. Ensuring secure AI deployment in CPS requires addressing these vulnerabilities, particularly in AI-driven autonomous systems where operational integrity is paramount.

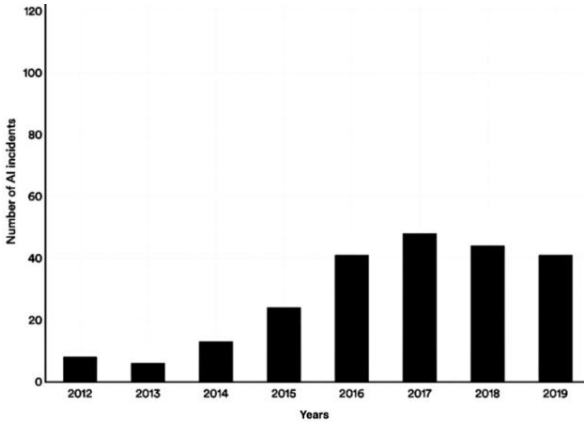

Fig. 1. Comparison of number of AI incidents

Despite significant advancements in containerized AI deployment for cloud environments, a gap exists in research focused on designing cohesive containerized AI systems that bridge both cloud and CPS environments. Current frameworks either concentrate on cloud-native AI systems or focus on CPS-specific applications, leaving a lack of integrated solutions that address the distinct challenges posed by both domains. AI systems optimized for cloud environments are typically not well-suited for the resource-constrained and real-time operational demands of CPS, while containerization techniques developed for the cloud often fall short in addressing CPS-specific performance and security issues.

Furthermore, the scalability of AI systems in hybrid cloud-edge architectures remains an unresolved issue. Real-time processing in CPS demands AI systems that can deliver low-latency responses while scaling efficiently across distributed infrastructures – a combination that existing container orchestration solutions like Kubernetes struggle to achieve. Additionally, ensuring the security of containerized AI in distributed CPS environments presents complex challenges, as traditional cloud security measures do not fully account for the specific risks inherent to CPS [5].

This article aims to address these gaps by proposing a unified framework for the design and deployment of containerized AI systems that operate efficiently across both cloud and CPS infrastructures. Specifically, it seeks to identify solutions that ensure scalability, low latency, fault tolerance, and enhanced security in distributed, heterogeneous environments, advancing the current state of research in this area.

III. SCOPE OF WORK AND OBJECTIVES

This article investigates the architecture and design of containerized AI systems optimized for deployment in both cloud and Cyber-Physical Systems environments. The primary objective is to establish a unified framework that ensures scalability, elasticity, and real-time AI processing across distributed, hybrid infrastructures. A key focus is on defining architectural principles that support seamless and adaptive operation within both cloud and CPS settings, ensuring that systems can dynamically adjust to varying computational demands and scales.

In addressing the technical challenges of CPS, the work delves into essential requirements such as low-latency communication, container orchestration, and real-time decision-making, which are particularly critical where rapid AI responses are necessary in resource-constrained environments. Security and resilience are central to the framework as well, with advanced protection techniques aimed at safeguarding AI models and data against adversarial threats across cloud and edge environments. The article further provides recommendations for integrating edge AI and hybrid cloud architectures to enhance CPS performance, reduce latency, and support distributed AI processing. By balancing computational loads across cloud and edge systems, the proposed framework ensures that CPS applications can meet their demanding real-time requirements. Altogether, this research aims to advance the reliable, efficient, and secure deployment of containerized AI systems that cater to the specific needs of both cloud and CPS environments.

IV. DESIGN PRINCIPLES FOR CONTAINERIZED AI SYSTEMS

Designing containerized AI systems requires careful attention to key architectural principles that ensure flexibility, efficiency, and seamless integration with both cloud and CPS. One foundational principle is modularity, which involves breaking down AI models into independent components that can be deployed and managed separately within containers. This modularity enables easier updates, improves maintainability, and allows for more granular control over different parts of the system, which is especially useful in hybrid cloud-CPS environments. In addition to modularity, resource efficiency is a critical consideration, particularly in resource-constrained CPS environments where computing power, memory, and bandwidth are limited. The containerization of AI models helps to optimize the use of system resources by isolating workloads and ensuring that each container has the necessary resources allocated without causing overhead (Fig. 2).

Moreover, adopting a microservices architecture allows the containerized AI system to be structured as a collection of loosely coupled, independently deployable services [6]. This approach enhances the flexibility of AI deployments, enabling the system to be scaled horizontally by adding more containers or services without needing to redeploy the entire system. Microservices further support the independent scaling and updating of individual AI models, ensuring continuous operation and reducing downtime.

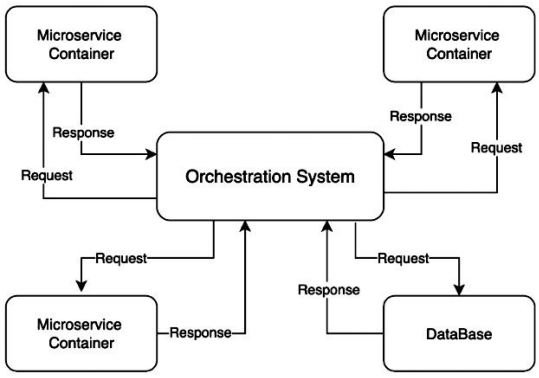

Fig. 2. Microservice Architecture Principles

Scalability is one of the most important considerations when designing containerized AI systems, especially in cloud environments where workloads often fluctuate. Cloud-native systems must be capable of auto-scaling in response to varying levels of demand. In AI applications, this might involve scaling up computational resources when training large models or when handling increased traffic in inference requests. Kubernetes, as a leading container orchestration platform, offers powerful auto-scaling capabilities that allow containerized AI workloads to be dynamically adjusted based on real-time metrics such as CPU and memory utilization.

In the context of CPS, managing real-time scaling requirements poses additional challenges. Unlike cloud environments, CPS often involve real-time processing of data from sensors and other IoT devices, where even slight delays can impact operational integrity. AI models deployed within CPS must not only scale in response to workload changes but also do so without compromising system responsiveness or reliability. Real-time scaling in CPS requires a fine balance between resource availability, task prioritization, and latency minimization to maintain system stability and avoid interruptions in critical applications, such as autonomous vehicles or industrial control systems.

Ensuring high availability and fault tolerance is essential for both cloud-based and CPS-integrated AI systems. Kubernetes plays a crucial role in maintaining fault tolerance through its orchestration capabilities. By distributing AI containers across multiple nodes in a cluster, Kubernetes ensures that workloads remain available even if individual nodes fail. This redundancy mechanism allows the system to recover automatically by rescheduling containers to healthy nodes, minimizing downtime and ensuring continuous availability.

In CPS environments, fault tolerance takes on heightened importance due to the critical nature of many real-time applications. Any system downtime in a CPS

could lead to significant disruptions, ranging from economic losses in industrial systems to life-threatening situations in autonomous systems or healthcare applications. To address this, AI systems in CPS must be designed with redundant components and failover strategies. In some cases, a combination of edge and cloud computing can enhance fault tolerance by offloading critical tasks to local edge nodes, which can continue operating independently if connectivity to the cloud is lost.

Furthermore, special attention must be given to system reliability in CPS, where predictable, real-time operation is crucial [7]. Techniques such as container health checks, service mesh for managing inter-container communication, and proactive monitoring can all be implemented to ensure the reliability and responsiveness of containerized AI workloads. Ultimately, the design of these systems must prioritize not only fault recovery but also the prevention of faults through redundancy, monitoring, and dynamic reallocation of resources to mitigate risks.

V. SECURITY AND PRIVACY IN CONTAINERIZED AI SYSTEMS

Containerized AI systems, while offering advantages such as scalability and portability, introduce a unique set of security challenges. One major concern is the potential vulnerability of container images, which are often shared across environments and can be susceptible to tampering or the inclusion of malicious code. Image tampering, if unchecked, can result in the deployment of compromised AI models that threaten the integrity of both cloud and CPS. Another critical vulnerability lies in insecure APIs, which are commonly used in AI systems for model deployment and data exchange. These APIs, if improperly secured, can expose the system to unauthorized access, data leaks, or even manipulation of AI decision-making processes [8].

Runtime threats also pose significant risks. Containers share the host system's kernel, making them vulnerable to exploits targeting the kernel itself. If a container is compromised, an attacker can potentially gain access to the underlying system and other containers running on the same host. Securing containerized workloads requires multiple layers of defense, including network isolation to limit communication between containers and external services, thereby reducing the attack surface. In Fig 3. these multiple layers are presented.

Techniques like image scanning are essential for detecting vulnerabilities in container images before they are deployed, ensuring that only verified, secure images are used in production environments. Runtime protection mechanisms, such as security policies that monitor and control container behavior during execution, help detect and respond to abnormal activities that may indicate a security breach.

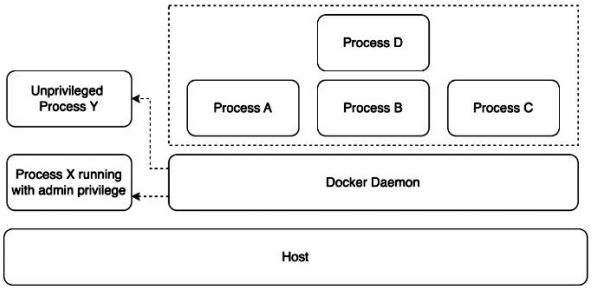

Fig. 3. Container runtime security privileges example

In addition to the general container security concerns, AI models themselves present specific security challenges, especially in CPS where real-time, data-driven decision-making is critical. One key threat is adversarial attacks, where malicious inputs are crafted to manipulate AI models into making incorrect predictions or decisions. In CPS environments, these attacks can have severe consequences, particularly in applications like autonomous vehicles or industrial control systems. To mitigate these risks, it is essential to implement robust threat modeling for AI-driven CPS. This involves identifying potential attack vectors, such as adversarial manipulation of input data, and developing strategies to harden AI models against these attacks.

Techniques like differential privacy can be employed to secure the training data used in AI models, ensuring that sensitive information is protected even if adversaries gain access to the data. Encryption of both AI models and their outputs during inference also helps prevent unauthorized access or tampering. For containerized AI, it is important to ensure that AI models and the data they process remain secure within the (GDPR) in the European Union or the Health Insurance Portability and Accountability Act (HIPAA) in the United States, is critical to maintaining the privacy and integrity of data processed by AI systems. In CPS environments, which often handle personal or mission-critical data, ensuring compliance becomes even more challenging.

Containerized AI systems must incorporate mechanisms for maintaining data sovereignty, ensuring that data is stored and processed in accordance with jurisdictional requirements. This includes implementing robust data governance policies that specify how data is collected, used, stored, and shared. Privacy-preserving techniques, such as anonymization and secure multi-party computation, can help ensure that sensitive data remains protected while still enabling valuable insights from AI models. Organizations must also ensure that their AI deployments meet industry-specific regulations, such as those governing healthcare data or financial transactions, by implementing governance frameworks that audit and monitor compliance.

In CPS environments, which often integrate cloud, edge, and physical systems, maintaining regulatory compliance can be particularly complex. Special attention must be paid to the transmission of data

between CPS nodes and the cloud to ensure that data privacy and security standards are met across the entire system. Addressing compliance requirements holistically, from data collection at the edge to AI processing in the cloud, ensures that containerized AI deployments remain secure, transparent, and aligned with legal and regulatory frameworks.

VI. CASE STUDIES AND PRACTICAL APPLICATIONS

The deployment of containerized AI systems in cloud environments has become a standard practice in industries seeking to leverage scalability, efficiency, and cost savings. One prominent example is the finance industry, where institutions have implemented containerized AI to enhance fraud detection, automate trading algorithms, and improve customer service through AI-driven chatbots. The ability to containerize these models allows for rapid scaling in response to fluctuating transaction volumes, especially during market peaks, and ensures that the infrastructure can adjust dynamically without over-provisioning.

In healthcare, containerized AI systems have been deployed in cloud environments to accelerate medical image analysis, personalized treatment recommendations, and predictive diagnostics. For instance, hospitals and research institutions can deploy containerized AI models in a hybrid cloud infrastructure, balancing heavy computational loads in the cloud while allowing sensitive patient data to be processed at the edge for privacy and compliance [9,10]. This modular approach has improved efficiency in medical imaging by enabling AI models to be updated continuously without interrupting services.

In the manufacturing industry, cloud-based AI is being used to optimize production lines through predictive maintenance. By deploying AI models in containers, manufacturers can monitor the health of machinery in real time, predict failures before they occur, and reduce downtime. These containerized solutions provide a scalable approach to integrating AI with industrial control systems while minimizing disruption to ongoing operations.

Practical lessons learned from these implementations highlight the importance of container orchestration for managing large-scale AI deployments. Kubernetes has emerged as a critical tool for orchestrating these containers, ensuring efficient load balancing, fault tolerance, and resource optimization. However, these industries also face challenges related to data security, latency, and governance, all of which must be carefully managed to maintain trust and reliability in AI systems.

In CPS containerized AI plays a transformative role in enabling real-time decision-making and autonomous control. One of the most compelling use cases is in the automotive industry, where containerized AI is integral to autonomous vehicle (AV) development. AVs rely on real-time processing of sensor data, such as lidar, radar, and cameras, to navigate and make decisions on the road. By containerizing AI models, automotive companies can deploy updates quickly and ensure that AI systems are continuously learning from new data. These containerized systems are often deployed in hybrid cloud-edge architectures, where critical inference is performed at the edge (onboard the vehicle), while more computationally intensive tasks, such as model training, are handled in the cloud [11].

Smart city projects provide another illustration of AI-driven CPS applications. Cities around the world are deploying AI to manage traffic flow, monitor energy consumption, and enhance public safety through real-time surveillance systems. In these environments, containerized AI models enable scalability and flexibility by allowing city planners to adjust and update AI systems in response to changing urban dynamics without overhauling the entire infrastructure. For example, smart traffic management systems use AI to predict congestion and adjust traffic signals dynamically. The use of containerized AI allows these models to be redeployed quickly as new data becomes available or as traffic patterns change.

In the IoT sector, containerized AI systems are being used to monitor and manage vast networks of connected devices. For example, industrial IoT systems deploy containerized AI at the edge to process sensor data in real time, ensuring that machinery and equipment operate efficiently. By integrating AI with IoT devices in a containerized architecture, companies can deploy new models to the edge without requiring extensive reconfigurations or downtime.

These real-world case studies underscore the growing reliance on hybrid cloud-edge architectures in CPS, where edge AI reduces latency by processing critical data closer to the source, while cloud computing provides the scalability needed for large-scale data processing and model training. The integration of these two architectures allows for efficient real-time AI processing and decision-making in CPS.

Despite the growing success of containerized AI systems in cloud and CPS environments, several challenges persist. One of the most pressing challenges is latency minimization, especially in real-time CPS applications where even millisecond delays can be critical. Autonomous vehicles, for instance, cannot afford high latencies in AI processing, as this would impair their ability to make split-second decisions. The solution often lies in edge computing, where containers run locally on the device to handle time-sensitive tasks, while less urgent processing occurs in the cloud. Balancing the load between edge and cloud through container orchestration can help minimize latency and ensure real-time performance.

Another challenge is resource optimization, particularly in CPS where computational resources may be constrained. AI models often require significant amounts of processing power, and containerization can introduce overhead. To address this, developers must

carefully optimize AI models for deployment in resource-limited environments, using techniques such as model compression or distributed computing. Additionally, container orchestration platforms like Kubernetes can dynamically allocate resources based on real-time demand, improving efficiency.

Security remains a major concern, especially in hybrid environments where data flows between cloud and edge systems. To mitigate security risks, solutions such as encrypted communication channels, secure container images, and robust access controls are critical. In industries like healthcare, where regulatory compliance is paramount, ensuring that containerized AI systems adhere to privacy laws (such as GDPR and HIPAA) adds another layer of complexity.

These challenges highlight the importance of robust design, careful orchestration, and an integrated approach to managing containerized AI systems in cloud and CPS environments. By addressing these obstacles through innovative solutions, industries can harness the full potential of containerized AI, driving operational efficiency and enabling advanced, real-time applications.

VII. CONCLUSION

This article highlighted several unique challenges that arise from containerizing AI in both cloud and CPS architectures. Key obstacles include minimizing latency in real-time applications, managing constrained resources in CPS environments, and addressing security vulnerabilities, particularly in hybrid cloud-edge deployments. The use of orchestration tools like Kubernetes, along with techniques such as network isolation, runtime protection, and model optimization, provides effective solutions to these challenges. However, the evolving nature of AI and CPS demands continued research and innovation to refine these approaches further.

The integration of containerized AI systems into cloud and CPS marks a critical advancement in computing, delivering substantial gains in scalability, responsiveness, and security. In cloud settings, containerization facilitates a modular, scalable framework for handling AI tasks, allowing organizations to dynamically adjust to varying demand and optimize resource utilization. For CPS, where real-time processing and decision-making are essential, containerized AI systems support, reliable deployment while enabling the flexibility required for ongoing updates.

As the demands on AI systems in cloud and CPS environments grow, there is a clear need for continued exploration into advanced containerization strategies. Researchers and practitioners alike are encouraged to investigate more efficient and secure methods for deploying containerized AI in complex, distributed infrastructures. The integration of AI with CPS, in particular, presents an exciting frontier where innovations in containerized architectures can drive the next generation of autonomous systems, smart cities, and industrial applications. By advancing the design principles, orchestration techniques, and security measures surrounding containerized AI, the industry can unlock new possibilities for scalable, reliable, and real-time AI deployment across a range of critical sectors.

References

[1] Pahl, C., Jamshidi, P., & Zimmermann, O. (2020). Microservices and containers. pp. 115–116. DOI: https://doi.org/10.18420/SE2020_34

[2] Al-Garadi, M. A., Mohamed, A., Al-Ali, A. K., Du, X., Ali, I., & Guizani, M. (2020). A survey of machine and deep learning methods for internet of things (IoT) security. IEEE communications surveys & tutorials, 22(3), 1646-1685. DOI: https://doi.org/10.1109/COMST.2020.2988293

[3] Qayyum, A., Usama, M., Qadir, J., & Al-Fuqaha, A. (2020). Securing connected & autonomous vehicles: Challenges posed by adversarial machine learning and the way forward. IEEE Communications Surveys & Tutorials, 22(2), 998-1026. DOI: https://doi.org/10.1109/COMST.2020.2975048

[4] Cicconetti, C., Conti, M., Passarella, A., & Sabella, D. (2020). Toward distributed computing environments with serverless solutions in edge systems. IEEE Communications Magazine, 58(3), 40-46. DOI: https://doi.org/10.1109/MCOM.001.1900498

[5] Shi, Y., Yang, K., Jiang, T., Zhang, J., & Letaief, K. B. (2020). Communication-efficient edge AI: Algorithms and systems. IEEE Communications Surveys & Tutorials, 22(4), 2167-2191. DOI: https://doi.org/10.1109/COMST.2020.3007787

[6] Al-Doghman, F., Moustafa, N., Khalil, I., Sohrabi, N., Tari, Z., & Zomaya, A. Y. (2022). AI-enabled secure microservices in edge computing: Opportunities and challenges. IEEE Transactions on Services Computing, 16(2), 1485-1504. DOI: https://doi.org/10.1109/TSC.2022.3155447

[7] Zhang, J., Tian, J., Luo, H., Wu, S., Yin, S., & Kaynak, O. (2024). Prognostics for the Sustainability of Industrial Cyber-Physical Systems: From an Artificial Intelligence Perspective. IEEE Transactions on Industrial Cyber-Physical Systems. DOI: https://doi.org/10.1109/TICPS.2024.3433492

[8] Hoenig, A., Roy, K., Acquaah, Y., Yi, S., & Desai, S. (2024). Explainable AI for Cyber-Physical Systems: Issues and Challenges. IEEE Access. DOI: https://doi.org/10.1109/ACCESS.2024.3395444

[9] Subasi, A., Ozaltin, O., Mitra, A., Subasi, M. E., & Sarirete, A. (2023). Trustworthy artificial intelligence in healthcare. In Accelerating Strategic Changes for Digital Transformation in the Healthcare Industry (pp. 145-177). Academic Press. DOI: https://doi.org/10.1016/B978-0-443-15299-3.00015-4

[10] Bershchanskyi, Y., & Klym, H. (2023). Information System for Administration of Medical Institution. In 2023 13th International Conference on Dependable Systems, Services and Technologies (DESSERT) (pp. 1-4). IEEE. DOI: https://doi.org/10.1109/DESSERT61349.2023.10416537

[11] Al-Doghman, F., Moustafa, N., Khalil, I., Sohrabi, N., Tari, Z., & Zomaya, A. Y. (2022). AI-enabled secure microservices in edge computing: Opportunities and challenges. IEEE Transactions on Services Computing, 16(2), 1485-1504. DOI: https://doi.org/10.1109/TSC.2022.3155447

Bershchanskyi Yevhen was born in 1997, in Ukraine. In 2019, he received a master's degree in Computer Engineering in the department of Specialized Computer Systems at Lviv Polytechnic National University. In 2023, he entered a PhD program in the department of Specialized Computer Systems at Lviv Polytechnic National University. His research interests include AI, security, machine learning, and cloud computing.

Halyna Klym professor of the department of Specialized Computer Systems of the Institute of Computer Technologies, Automation and Metrology of Lviv Polytechnic National University. In 2016, she received a Doctor of Science degree in Technical Sciences at Lviv Polytechnic National University. She conducts lecture courses on the design of ultra-large integrated circuits and methods and means of automated design of computer systems. She is an author of more than 170 scientific articles in international publications.

Yurii Shevchuk holds a master's degree from Lviv Polytechnic National University in Information Security Management. He has extensive experience in developing corporate software solutions in the fields of healthcare, e-commerce. His research interests include cybersecurity, integration of advanced security protocols into scalable systems, web accessibility, computing technologies, and software architecture.