Machine Learning for Performance Optimization in Resource-Constrained Environments

Edoardo Di Caro*, Mauro Tortonesi*
* Distributed Systems Research Group, University of Ferrara, Ferrara, Italy
Email: {edoardo.dicaro, mauro.tortonesi}@unife.it

***Abstract*—The rapid growth of digital technology has significantly increased reliance on computer systems across many different fields. This widespread adoption has also led to the deployment of devices in challenging scenarios characterized by limited resources, such as constrained computational capabilities or unreliable communication. These environments require alternative, innovative solutions, often specifically tailored to individual use cases. However, if operational conditions change at runtime, causing systems to function in unforeseen scenarios, there is a considerable risk of performance degradation or inefficient use of already limited and valuable resources. To address these complexities, Machine Learning (ML) methods offer significant potential by enabling system to autonomously adapt to evolving conditions. Techniques including predictive analytics and self-optimization allow systems to intelligently adjust at runtime, dynamically optimizing resource usage and performance. However, the effectiveness of ML-driven optimization is significantly limited by the scarcity of realistic data, particularly in scenarios involving limited resources. To mitigate this challenge, generative artificial intelligence emerges as a pivotal tool, facilitating data augmentation, enhancing the effectiveness of the development and testing processes of software solutions.**

***Index Terms*—Resource-Constrained Environments, Machine Learning, Generative AI, Industrial Edge, Disaster Recovery**

I. Introduction

The rapid expansion of digital transformation has significantly increased reliance on computational systems across diverse environments. While cloud computing has traditionally offered abundant resources for data processing, storage, and analytics, there is a growing need for efficient computation in more resource-constrained environments. These kind of environments—ranging from industrial edge computing and disaster recovery operations to tactical networks—introduce unique challenges due to limited processing power, energy constraints, and intermittent or unreliable connectivity [1].

Given these constraints, optimizing the available resources is essential to ensure effective and reliable system performance. However, the variability of operating conditions makes it impractical to predefine an optimal configuration for all possible scenarios. This calls for adaptive systems that can dynamically reconfigure themselves at runtime, autonomously adjusting their behavior in response to changing conditions and requirements.

To address these necessities, ML techniques have emerged as a promising solution to enhance system performance, improving reliability and availability in challenging conditions [2]. Systems can leverage techniques like predictive analytics and self-optimization mechanisms to achieve runtime performance optimization, ensuring efficiency even in dynamic and unpredictable environments.

Given the wide variety of applications that could benefit from a dynamic optimization approach, the challenges and variables involved are equally diverse. Some environments may face unreliable connectivity and limited network resources, while others may be characterized by computationally constrained devices, or a combination of both. As such, to fully unlock the potential of machine learning solutions in resource-constrained settings, a comprehensive adaptation framework is essential to enable automatic and dynamic optimization [3] [4]. This framework should integrate real-time monitoring with an intelligent decision-making process, allowing it to adapt seamlessly based on environmental feedback.

The design and development of such a comprehensive and complex solution requires extensive training and testing of the ML models and software systems involved. To achieve this, however, it is necessary to have access to diverse and realistic data—something that is often scarce, especially for resource-constrained environments. In this context, generative Artificial Intelligence (AI) [5] emerges as both enabling and supporting technology, mitigating data scarcity [6] and enhancing systems' situational awareness, strengthening the decision-making process.

The remainder of this paper follows the progression of my doctoral research. It begins by exploring resource-constrained environments, highlighting their unique challenges and specific requirements, underlying both the common grounds and the key differences between industrial edge computing and disaster recovery operations. This analysis helps clarify the role of ML in such contexts and identify the most suitable approaches. Subsequently, the paper presents ongoing research activities, showcasing the contributions made and the encouraging results achieved. Finally, it concludes by outlining future directions intended to drive further innovation and advancement in the field of resource-constrained environments.

II. Resource-Constrained Environments

Resource-constrained environments refer to settings where computational, power, storage, or communication resources are significantly limited, often requiring specialized approaches to maintain functionality and efficiency. In such

978-3-903176-75-1

scenarios, devices may have low processing power, limited memory, or operate on battery power with strict energy constraints. As such, ensuring system reliability and adaptability is crucial.

These environments are common in industrial edge computing, where processing is performed close to data sources—such as sensors, embedded systems, or industrial controllers—rather than centralized cloud servers. They are also prevalent in disaster recovery operations, where infrastructure damage can severely restrict access to power and network connectivity, making reliable communication and computation difficult. Understanding the unique challenges of resource-constrained environments is key to designing systems that can operate effectively despite these limitations.

A. Industrial Edge Scenarios

The Industrial Edge refers to computing environments at the periphery of industrial networks, where data is processed closer to its source rather than being sent to centralized cloud servers. This approach is increasingly used in smart manufacturing, automation, and critical infrastructure to enable low latency decision-making, reducing cloud reliance while enhancing data privacy. However, Industrial Edge environments are often characterized by limited resources, intermittent connectivity, and the integration of many different devices. Constraints in computational power, memory, bandwidth, and energy, pose challenges for data-intensive tasks like predictive maintenance and quality control [7]. These challenges include, but are not limited to, efficient resource utilization, adaptive workload management and optimizing ML models to run effectively on edge devices.

Machine Learning plays a critical role in optimizing Industrial Edge environments. Techniques like Federated Learning (FL) allow for localized model training, reducing reliance on cloud communication and enhancing privacy. In addition, more capable devices can perform heavier computations, optimizing network use through local operations like filtering and aggregation [8] to communicate fewer, but still significative, data. By leveraging ML-driven optimizations, Industrial Edge computing enhances reliability, efficiency, and intelligence, offering a strong alternative to centralized processing for industrial applications. As these systems evolve, advancements in edge-aware ML models and federated intelligence will be key to overcoming current limitations and unlocking new capabilities.

B. Disaster Recovery Operations

Disaster Recovery Operations in humanitarian contexts refer to the strategies and technologies used to restore essential services and infrastructure following a natural disaster or major disruption. These operations are critical for minimizing the impact on affected communities, ensuring access to basic needs and supporting recovery efforts. As such, rapid service restoration is crucial for saving lives and maintaining public safety. However, these operations face unique challenges, including damaged communication infrastructure, limited communication resources, and the potential for widespread damage to devices and equipment [9].

Key characteristics of these scenarios include the need for rapid deployment in a disrupted environment, often with limited resources, and the reliance on backup systems that may not have the full capabilities of the original infrastructure. Communication networks may be impaired, leaving only sporadic or low-bandwidth connections available, and devices may be damaged or non-operational. Additionally, energy constraints are common, with recovery efforts often relying on battery-powered, energy-efficient devices, which limit computational capacity and duration of operations.

The challenges in Disaster Recovery Operations are multifaceted. Efficient resource utilization is crucial, as computational power, memory, and bandwidth are often constrained. Communication gaps can hinder coordination, requiring innovative solutions to maintain data flow and synchronize recovery efforts. Moreover, with devices possibly damaged or energy-constrained, ensuring system resilience and reducing the need for high-power operations is a significant challenge.

Solutions to these challenges involve a mix of advanced technologies and strategies. Lightweight data transmission protocols and edge computing can help maintain operations in environments with limited bandwidth and damaged infrastructure. Additionally, ML models can be used to predict recovery needs and allocate resources dynamically based on real-time conditions. As these systems evolve, further advancements in resilient and autonomous strategies will be critical for ensuring effective Disaster Recovery Operations, even in the most challenging environments.

III. Ongoing Research

The challenges faced in both industrial edge computing and disaster recovery operations are often strikingly similar, despite their distinct contexts. Both environments are characterized by limited resources, and systems must adapt dynamically to changes in the environment, whether it's the fluctuating connectivity in industrial edge setups or the unpredictable nature of post-disaster recovery efforts.

This section will highlight ongoing research and preliminary results on how both ML and generative models can boost performance in resource-constrained environments.

A. Cross-Sector Technologies to Enable and Support Adaptation Capabilities

As previously mentioned, in resource-constrained environments the ability to dynamically adjust system configurations based on real-time conditions is crucial for optimizing both resource utilization and overall system performance. My early work in this area led to the development of the DYnamic NEtwork Reconfiguration (DYNER) framework, which utilizes a Sense-Decide-Adapt loop designed to enable autonomous system adaptations in response to changing or unpredictable operational conditions.

As depicted in Figure 1, the first step in this framework is sensing. This phase involves the continuous collection of

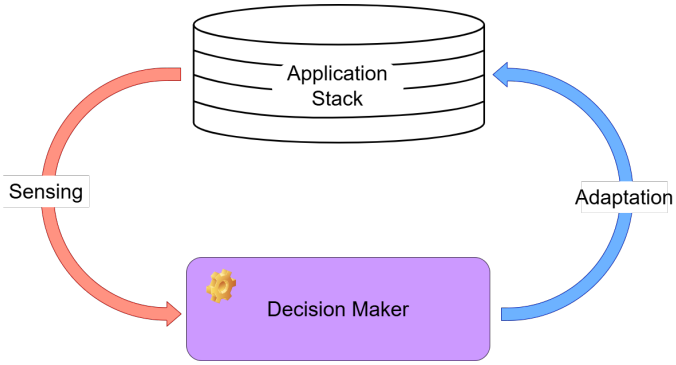

Fig. 1. Sense-Decide-Adapt loop of the DYNER framework.

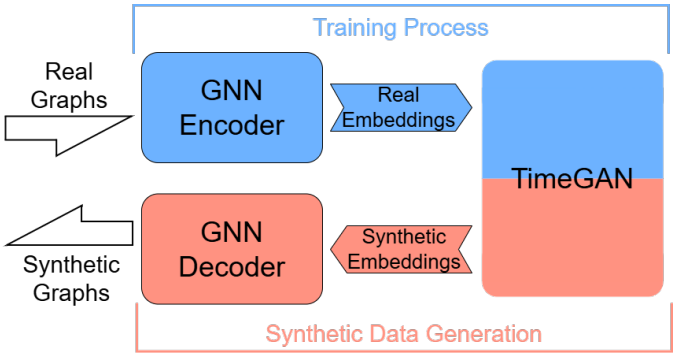

Fig. 2. Architecture of TimeGraph for synthetic generation of graphs.

information regarding the system's operational state and its environment. Different sensing strategies can be employed to monitor key parameters, such as available computational power, network bandwidth, energy levels, and other context-specific variables.

Once relevant data is gathered, it must be processed to make informed decisions and determine the appropriate course of action. This is the role of the Decision-Maker component, which can leverage either domain-specific algorithms or ML models to analyze the sensed data and predict the best actions to optimize system performance. If this process determines that the current system configuration is suboptimal, adaptation is triggered. This may involve reallocating resources, reconfiguring tasks, or switching between different operational modes to enhance efficiency. Notably, certain components of the DYNER framework itself can be reconfigured at runtime to better align with new conditions.

To demonstrate the effectiveness of this approach, a Mininet-WiFi [10] testbed was used to simulate network communication with two disruptive events. In the first, infrastructure failure forced devices to switch to Ad-Hoc communication, while in the second, network resource constraints required reducing control traffic to avoid congestion. The DYNER-enabled system responded effectively to both events, while a system without such capabilities suffered from node isolation and communication breakdown.

TimeGraph is a result of this ongoing research, a generative model capable of creating realistic connectivity graphs that evolve over time. The system architecture consists of two main components, as shown in Figure 2. The first component is a Graph Neural Network (GNN) that takes a connectivity graph as input and outputs an encoding in a latent space. This process utilizes state-of-the-art solutions, such as NESS [11] for GNN training and Optuna [12] for hyperparameter optimization, to maximize the quality of the encodings while minimizing the latent space size. This reduces the risk of error propagation throughout the pipeline. The GNN transforms a sequence of connectivity graphs into a sequence of encodings, which are then treated as a multivariate time series to train a Time-series Generative Adversarial Networks (TimeGAN) [13]. This captures temporal correlations in the data and generates artificial time sequences that closely, but not exactly, resemble the original data. By decoding the generated sequences using the same GNN, the system produces realistic artificial connectivity graphs that evolve over time.

To test the performance of TimeGraph, the Vignette 2 of the Anglova Scenario [14], a real tactical edge scenario, was used. It features key characteristics of constrained networks, such as limited resources, intermittent connectivity, and node mobility. The entire two-hour scenario was divided into smaller time windows to create a dataset large and varied enough for effective training. The results were highly promising, with the system succeeding at modeling both temporal dependencies and structural features, ultimately generating artificial scenarios that closely follow the distribution of the original data.

B. ML Solutions for Resource Usage Optimization

In parallel to these works, which have a broader scope, during my early stages of PhD I also worked on more practical solutions, that can bring benefit on their own but can also be integrated in a more comprehensive future solution.

Focusing on optimizing communication in edge networks, which often face severe resource constraints like limited bandwidth and high latency, we leverage the concept of Semantic Filtering. We designed a platform that utilizes state-of-the-art embeddings generated by both large and small language models to filter out semantically redundant data. By transforming data into vector representations, we can compare the semantic similarity between data points, enabling us to identify and eliminate redundant transmissions.

This approach significantly reduces the amount of data sent over the network, which is crucial in environments with tight bandwidth constraints. At the same time, it ensures that the most relevant and critical information is preserved, preventing the loss of valuable data that could impact decision-making or operational efficiency. The semantic filtering process is particularly effective in edge environments, where local computation is a key advantage. Ultimately, this solution not only saves bandwidth but also optimizes the overall performance of edge networks by enabling more intelligent, context-aware communication. By focusing on the semantic value of data and minimizing unnecessary transmissions, we create a more

efficient and responsive system, tailored to the unique demands of edge environments.

In a similar vein, we then focused on optimizing communication and resource usage in more complex distributed edge networks, like the ones found in industrial settings. We designed the FedEdge-Learn framework to address several critical challenges prevalent in such environments, such as the scarcity of labeled data, concerns around data privacy, and the inherently decentralized nature of industrial data. These issues are particularly challenging in the context of Industry 5.0, which integrates advanced technologies like Internet of Things (IoT) and edge computing to enable sustainable, efficient, and personalized manufacturing processes. By utilizing FL, FedEdge-Learn ensures that machine learning models can be trained across multiple edge devices without centralizing raw data. Instead of transmitting large amounts of sensitive information to a central server, the framework only shares model updates, reducing bandwidth consumption and ensuring data privacy of potentially sensitive information. This not only addresses privacy concerns but also reduces the overall communication overhead, which is essential for maintaining efficiency in resource-constrained industrial IoT environments.

Like semantic filtering, FedEdge-Learn emphasizes local computation on edge devices, allowing them to process data and adapt models independently. This localized approach enables the system to overcome bandwidth and latency challenges while maintaining efficient model updates. Both methods optimize the use of limited resources by minimizing unnecessary data transmissions and focusing on data privacy and security. They ensure that sensitive data remains within the local network, reducing the risk of exposure while still enabling critical machine learning capabilities. This ability to enable edge devices to perform machine learning tasks with minimal data exchange positions it as a powerful solutions for resource-constrained environments.

IV. Conclusions and Future Works

This paper presents my early doctoral research on optimizing performance in resource-constrained environments, focusing on the role of ML and generative AI. We highlight the potential of self-optimizing systems like the DYNER framework, which leverages a Sense-Decide-Adapt loop for real-time adaptation, and show how generative AI, through the TimeGraph model, enhances training datasets and simulates resource-limited scenarios to support more robust system development and testing. Together, these approaches enable systems to efficiently respond to dynamic conditions while addressing data scarcity.

As part of our future work, we identify two key open questions that will guide the next steps of this research. (1) How can we design adaptive decision-making frameworks, such as DYNER, that are lightweight yet intelligent, providing stability, privacy, and resource efficiency even under intermittent connectivity in constrained edge environments? (2) How can we create and validate realistic datasets that both capture the complexity of heterogeneous real-world scenarios and support reproducible research, reducing reliance on purely synthetic traces while improving the robustness of ML-driven optimization solutions?

In conclusion, the development of adaptive, ML-driven solutions, supported by generative AI, is essential for advancing resource-constrained environments. By addressing system limitations and exploring new approaches, future work will help create more efficient, intelligent, and resilient solutions capable of operating in challenging conditions.

References

[1] D. Fadhil and R. Oliveira, "Characterization of the end-to-end delay in heterogeneous networks," in *2021 12th International Conference on Network of the Future (NoF)*, 2021, pp. 1–5.

[2] P. P. Shinde and S. Shah, "A review of machine learning and deep learning applications," in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 2018, pp. 1–6.

[3] C. Barz, E. Cramer, R. Fronteddu, M. Hauge, K. Marcus, J. Nilsson, F. Poltronieri, M. Tortonesi, N. Suri, and M. Zaccarini, "Enabling adaptive communications at the tactical edge," in *MILCOM 2022 - 2022 IEEE Military Communications Conference (MILCOM)*, 2022, pp. 1038–1044.

[4] C. Szabo, B. Sims, T. Mcatee, R. Lodge, and R. Hunjet, "Self-adaptive software systems in contested and resource-constrained environments: Overview and challenges," *IEEE Access*, vol. 9, pp. 10 711–10 728, 2021.

[5] A. Figueira and B. Vaz, "Survey on synthetic data generation, evaluation methods and gans," *Mathematics*, vol. 10, no. 15, 2022. [Online]. Available: https://www.mdpi.com/2227-7390/10/15/2733

[6] L. Alzubaidi, J. Bai, A. Al-Sabaawi, J. Santamaría, A. S. Albahri, B. S. N. Al-dabbagh, M. A. Fadhel, M. Manoufali, J. Zhang, A. H. Al-Timemy, Y. Duan, A. Abdullah, L. Farhan, Y. Lu, A. Gupta, F. Albu, A. Abbosh, and Y. Gu, "A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications," *Journal of Big Data*, vol. 10, no. 1, p. 46, Apr 2023. [Online]. Available: https://doi.org/10.1186/s40537-023-00727-2

[7] T. Qiu, J. Chi, X. Zhou, Z. Ning, M. Atiquzzaman, and D. O. Wu, "Edge computing in industrial internet of things: Architecture, advances and challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2462–2488, 2020.

[8] L. Colombi, S. Dahdal, E. Di Caro, R. Fronteddu, A. Gilli, A. Morelli, F. Poltronieri, M. Tortonesi, N. Suri, and C. Stefanelli, "Efficient data dissemination via semantic filtering at the tactical edge," in *MILCOM 2024 - 2024 IEEE Military Communications Conference (MILCOM)*, 2024, pp. 457–462.

[9] D. E. M. Ahmed and O. O. Khalifa, "An overview of manets: applications, characteristics, challenges and recent issues," 2017.

[10] R. R. Fontes, S. Afzal, S. H. B. Brito, M. A. S. Santos, and C. E. Rothenberg, "Mininet-wifi: Emulating software-defined wireless networks," in *Network and Service Management (CNSM), 2015 11th International Conference on*, Nov 2015, pp. 384–389.

[11] T. Ucar, "Ness: Node embeddings from static subgraphs," 2023. [Online]. Available: https://arxiv.org/abs/2303.08958

[12] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 2623–2631. [Online]. Available: https://doi.org/10.1145/3292500.3330701

[13] J. Yoon, D. Jarrett, and M. van der Schaar, "Time-series generative adversarial networks," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/c9efe5f26cd17ba6216bbe2a7d26d490-Paper.pdf

[14] N. Suri, J. Nilsson, A. Hansson, U. Sterner, K. Marcus, L. Misirlioğlu, M. Hauge, M. Peuhkuri, B. Buchin, R. in't Velt, and M. Breedy, "The angloval tactical military scenario and experimentation environment," in *2018 International Conference on Military Communications and Information Systems (ICMCIS)*, 2018, pp. 1–8.