Internet of Things

Sanjay Misra
Amit Kumar Tyagi
Vincenzo Piuri
Lalit Garg *Editors*

# Artificial Intelligence for Cloud and Edge Computing

Springer

# Internet of Things

## Technology, Communications and Computing

**Series Editors**
Giancarlo Fortino, Rende (CS), Italy
Antonio Liotta, Edinburgh Napier University, School of Computing, Edinburgh, UK

The series Internet of Things - Technologies, Communications and Computing publishes new developments and advances in the various areas of the different facets of the Internet of Things.

The intent is to cover technology (smart devices, wireless sensors, systems), communications (networks and protocols) and computing (theory, middleware and applications) of the Internet of Things, as embedded in the fields of engineering, computer science, life sciences, as well as the methodologies behind them. The series contains monographs, lecture notes and edited volumes in the Internet of Things research and development area, spanning the areas of wireless sensor networks, autonomic networking, network protocol, agent-based computing, artificial intelligence, self organizing systems, multi-sensor data fusion, smart objects, and hybrid intelligent systems.

** Indexing: *Internet of Things* is covered by Scopus and Ei-Compendex **

More information about this series at http://www.springer.com/series/11636

Sanjay Misra • Amit Kumar Tyagi • Vincenzo Piuri
Lalit Garg
Editors

# Artificial Intelligence for Cloud and Edge Computing

Springer

*Editors*
Sanjay Misra
Department of Computer Science
and Communication
Ostfold University College
Halden, Norway

Amit Kumar Tyagi
School of Computer Science and
Engineering
Vellore Institute of Technology
Tamil Nandu, India

Vincenzo Piuri
Dipartimento di Informatica
Universita' degli Studi di Milano
Milano, Italy

Lalit Garg
Faculty of Information and Communication
Technology
University of Malta
Msida, Malta

ISSN 2199-1073 ISSN 2199-1081 (electronic)
Internet of Things
ISBN 978-3-030-80820-4 ISBN 978-3-030-80821-1 (eBook)
https://doi.org/10.1007/978-3-030-80821-1



This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

# Preface

In this smart era, artificial intelligence and cloud computing are two significant technologies that are changing the way we live as humans. That is, artificial intelligence (AI) and cloud computing (and edge computing) have emerged together to improve the lives of millions of people. Edge computing is a new advanced version of cloud computing. Today's many digital (smart) devices like Siri, Google Home, and Amazon's Alexa are blending AI and cloud computing in our lives every day. On a larger scale, AI capabilities are employed in the business cloud computing environment to make organizations more efficient, strategic, and insight driven.

Today's cloud and edge computing acts as an engine to increase the scope of an area and impact of AI. AI and cloud (also edge) computing are transforming businesses at every level, especially with a major impact on large-scale businesses. A seamless flow of AI and cloud-based resources make many services a reality. In general, cloud computing offers businesses more flexibility, agility, and cost savings by hosting data and applications in the cloud. A cloud in business (industry) can be public, private, or hybrid, which provides services like SaaS, PaaS, and IaaS to end users. An essential service of cloud computing is infrastructure, which involves the provision of machines for computing and storage. Cloud providers also offer data platform services that span the different available databases.

Some of the benefits of AI and cloud or edge technologies are derivable with access to huge data sets, including refining the data (in an intelligent way), cost-effectiveness, increased productivity, reliability, and availability of advanced infrastructure. For example, an AI-powered pricing module can ensure that a company's pricing will always be optimized. It's not just about making better use of data, it's conducting the analysis and then putting it into action without the need for human intervention. Some other useful benefits of AI integration with cloud and edge computing include powering a self-Managing cloud with AI; improving data management with AI at the cloud level; delivering more value-added service with AI–SaaS integration; and utilizing dynamic cloud-AI integrated services. Thus, many organizations are now improving their commitment to investing in AI-based strategies or cognitive technologies.

Therefore, it is our pleasure to present to you this book: *Artificial Intelligence for Cloud and Edge Computing: Concepts and Paradigms*. The book provides useful in-depth information regarding current research in artificial intelligence on various cybersecurity challenges. The main goal of this book is to conduct analyses, implementation, and discussion of many tools (artificial intelligence, machine learning, and deep learning and cloud computing, fog computing, and edge computing, including concepts of cyber security) towards understanding the integration of these technologies. The book also contains self-assessment problems for increasing the knowledge of readers and case studies about real-world issues, that is, how AI and cloud/edge computing can change business forever. The book comprises 15 chapters from authors around the world. Below is a summary of each chapter.

Queiroz et al. in their chapter titled "An Optimization View to the Design of Edge Computing Infrastructures for IoT Applications" modeled the service placement issue using a multi-objective optimization aiming at minimizing two aspects: the response time for data transmission and processing in the sensors-edge cloud path, and the (energy or monetary) cost related to the number of turned-on edge nodes. They also proposed two heuristics, based on variable neighborhood search and genetic algorithms, and evaluated over a wide range of scenarios, considering a practical smart city application with 100 sensors and up to 10 edge nodes.

Rijal et al., in their chapter titled "AIOps: A Multivocal Literature Review," presented multivocal literature review to define AIOps, the benefits gained from it, the challenges an organization might face, and, finally, what lies in the foreseen future of the AIOps. Their findings revealed that adopting AIOps helps monitor IT work, optimizes time saving, improves human-AI collaboration, makes IT work proactive, and provides faster mean time to recovery (MTTR).

In the chapter titled "Deep Learning-Based Facial Recognition on Hybrid Architecture for Financial Services," Granados and Garcia-Bedoya explored a mechanism for deep learning–based facial recognition to improve customer identification in financial institutions on a hybrid architecture. They proposed a model that integrates traditional classifiers with convolutional neural networks (CNN) to facial recognition involving visual sequences or different image sources as social media as a tool for financial institutions. Thus, a hybrid architecture between edge-cloud computing, which catalyzes these face detection tools in personalized services, was employed.

In the chapter "Classification of Swine Disease Using K-Nearest Neighbor Algorithm on Cloud-Based Framework," Adeniji et al. used supervised learning to conduct real-time classification along with elucidating the device architecture that classifies photographs of diseased animals using k-nearest neighbor (KNN) algorithm. They developed a web-based expert system for the detection of swine diseases in pigs. The developed system is able to predict the likely occurrence of swine disease using the ML algorithm.

Alabi et al., in their chapter titled "Privacy and Trust Models for Cloud-Based EHRs Using Multilevel Cryptography and Artificial Intelligence," proposed a multi-level cryptography approach for achieving security (privacy) at both local and cloud-based electronic medical records (EMR) systems. They also applied subjective logic-based belief or reasoning models for measuring the trustworthiness of the medical personnel.

In the chapter titled "Utilizing an Agent-Based Auction Protocol for Resource Allocation and Load Balancing in Grid Computing," Ali et al. explored resource management challenges for grid systems. They proposed an economic-based approach for effective allocation of resources and scheduling of application. The proposed architecture uses distributed artificial intelligence. The proposed model aims to optimize the usage of resources to satisfy the demands of grid users.

Al-Janabi and Salman, in their chapter titled "Optimization Model of Smartphone and Smart Watch Based on Multi Level of Elitism (OMSPW-MLE)," proposed a model based on two artificial intelligent techniques, namely deterministic selection and ant optimization. The multi-level optimization model was aimed at performing a clustered process of large volume data streams and consisted of five stages which include: data collection, preprocessing of data, finding the most influential nodes in complex networks, finding the best subgraph, and result evaluation using three different measures.

In the chapter titled "K-Nearest Neighbour Algorithm for Classification of IoT-Based Edge Computing Device," Arowolo et al. proposed a method that guarantees the implementation of low-performance ML techniques on hardware in the Internet of Things model, creating means for IoT awareness. The study used the KNN ML algorithm for the implementation, and a confusion matrix in terms of accuracy was used to evaluate the system. The experiment results displayed an accuracy of 85%, outperforming other methods that have been suggested and compared within the literature.

Awotunde et al., in their chapter titled "Big Data Analytics of IoT-Based Cloud System Framework: Smart Healthcare Monitoring Systems," presented a BDA IoT-based cloud system storage for real-time data generated from IoT sensors and analysis of the stored data from the smart devices. The framework is tested using a big data Cloudera platform for database storage and Python for the system design. The framework's applicability is tested in real-time analysis of healthcare monitoring of patients' data for automatic managing of body temperature, blood glucose, and blood pressure. The integration of the system shows improvement in patients' health monitoring situations.

Sudeepa et al., in their chapter titled "Genetic Algorithm-Based Pseudo Random Number Generation for Cloud Security," proposed an innovative, intelligent framework to spawn a pseudo-random number secret key utilizing the combination of the feedback shift register and a genetic algorithm to intensify the key's strength in terms of its length. The proposed system is intended to enhance the sequence's length by the utilization of the genetic algorithm. They carried out various statistical tests to uphold the strength of the generated key sequences.

Khan et al. in "Anomaly Detection in IOT Using Machine Learning" proposed a comparison between different machine learning algorithms that can be used to identify and predict any malicious or anomalous data for a data set of environmental characteristics created by them. The data set is developed from the data exchanged between the sensors in an IoT environment of Bangladesh. The evaluation was carried out to determine which algorithm performs the best in terms of detecting anomalies in the given data set.

In the chapter titled "System Level Knowledge Representation for Edge Intelligence," Maio provided an overview of critical issues from a systems research viewpoint. They also presented novel constructors for AI, the system-level knowledge representation, and its relevance to IoT.

Narasimhan et al., in their paper titled "AI-Based Enhanced Time Cost-Effective Cloud Workflow Scheduling," presented an extension of an earlier work where time-effective scheduling algorithms were discussed. They presented two scheduling algorithms: time-constrained early-deadline cost-effective algorithm (TECA) to schedule these time-critical workflows with minimum cost, and versatile time-cost algorithm (VTCA) which consider both time and cost constraints; these algorithms considerably enhance the earlier algorithms.

In the paper titled "AI-JasCon: An Artificial Intelligent Containerization System for Bayesian Fraud Determination in Complex Networks," Okafor et al. presented AI containerization API system based on JAVA-SQL container (JSR-233) for fraud prediction and prevention in telecommunication networks. Pipeline modeling using Bayesian software implementation paradigm is introduced using AI-JasCon model. A demonstration of how AI engine works with a complex network system for observation of the number of calls, call frequency, and hidden activities for predictive classification (analytics) at the network backend is also discussed.

Finally, in the chapter titled "Performance Improvement of Intrusion Detection System for Detecting Attacks on Internet of Things and Edge of Things," Saheed proposed an improved IDS model for the classification of attacks on IoT and EoT. To protect EoT and IoT appliances and devices and improve IDSs-IoT, implementation of ten different machine learning models is proposed. They carried out normalization using the minimum-maximum (min-max) method. Then they performed dimensionality reduction using principal component analysis (PCA). The light gradient boosting machine, decision tree, gradient boosting machine, k-nearest neighbor, and extreme gradient boosting algorithms were used for classification.

In conclusion, cloud computing utilizing AI is certainly not a radical or progressive change. In numerous regards, it's a transformative one. We require "test and learn" abilities of AI and the cloud. We are sure that the fusion of cloud computing services and AI technology will significantly change the technology industry. In the near future, AI will become a factor of production with large storage capacity. Researchers and scientists will find the topics and examples given in this book very useful and instructive in the domain of AI and cloud and edge computing. Furthermore, investment in AI for cloud and edge computing will improve the profits and productivity of businesses in no small measure. The main goal is to conduct analyses, implementation, and discussion of many tools (artificial intelligence, machine learning, and deep learning and cloud computing, fog computing, and edge computing, including cyber security concepts) to understand the integration of these technologies.

On behalf of all editors,

Ota, Nigeria

Sanjay Misra

# Contents

# Editors' Biography

**Sanjay Misra** is a professor at Ostfold University College(HIOF), Halden Norway. Before coming to HIOF, he was a professor in Covenant University (400–500 ranked by THE(2019)) Nigeria for 9 years. He received his PhD. in Inf. & Know. Engg (Software Engg) from the University of Alcala, Spain and M.Tech. (Software Engg) from MLN National Institute of Tech, India. As per SciVal (SCOPUS – Elsevier) analysis (on 01.09.2021), he is the most productive researcher (Number 1) https://t.co/fBYnVxbmiL in Nigeria since 2017 *(in all disciplines)*, in comp science no 1 in the country & no 2 in the whole of Africa. Total more than 500 articles (SCOPUS/WoS) with 500 coauthors worldwide (-110 JCR/SCIE) in the core & appl. area of Soft Engg, Web engineering, Health Informatics, Cybersecurity, Intelligent systems, AI, etc. He got several awards for outstanding publications (2014 IET Software Premium Award(UK)), and from TUBITAK-Turkish Higher Education and Atilim University). He has delivered more than 100 keynote/invited talks/public lectures in reputed conferences and institutes (traveled to more than 60 countries). He is one of the editors of 58 LNCSs, 4 LNEEs, 1 LNNSs, 3 CCISs & 10 IEEE proceedings and 6 books, and editor in chief of *IT Personnel and Project Management* and *Int J of Human Capital & Inf Technology Professionals* – IGI Global and editor in various SCIE journals.

**Amit Kumar Tyagi** is an assistant professor (senior grade) and senior researcher at Vellore Institute of Technology (VIT), Chennai Campus, India. His current research focuses on machine learning with big data, blockchain technology, data science, cyber physical systems, smart and secure computing, and privacy. He has contributed to several projects such as "AARIN" and "P3-Block" to address some of the open issues related to the privacy breaches in vehicular applications (such as parking) and medical cyber physical systems. He received his Ph.D. degree from Pondicherry Central University, India. He is a member of the IEEE.

**Vincenzo Piuri** has received his Ph.D. in computer engineering from Politecnico di Milano, Italy (1989). He is full professor of computer engineering at the Università Degli Studi di Milano, Italy (since 2000). He has been associate professor at Politecnico di Milano, Italy, and visiting professor at the University of Texas at

Austin and at George Mason University, USA. His main research interests are: artificial intelligence, computational intelligence, intelligent systems, machine learning, pattern analysis and recognition, signal and image processing, biometrics, intelligent measurement systems, industrial applications, digital processing architectures, fault tolerance, dependability, and cloud computing infrastructures. Original results have been published in more than four hundred papers in international journals, proceedings of international conferences, books, and book chapters.

He is fellow of the IEEE, distinguished scientist of ACM, and senior member of INNS. He is president of the IEEE Systems Council (2020–21) and has been IEEE vice president for technical activities (2015), IEEE director, president of the IEEE Computational Intelligence Society, vice president for education of the IEEE Biometrics Council, vice president for publications of the IEEE Instrumentation and Measurement Society and the IEEE Systems Council, and vice president for membership of the IEEE Computational Intelligence Society.

Prof. Puiri has been editor-in-chief of the *IEEE Systems Journal* (2013–19) and associate editor of the *IEEE Transactions on Cloud Computing*, and has been associate editor of the *IEEE Transactions on Computers*, the *IEEE Transactions on Neural Networks*, the *IEEE Transactions on Instrumentation and Measurement*, and *IEEE Access*. He received the IEEE Instrumentation and Measurement Society Technical Award (2002) and the IEEE TAB Hall of Honor (2019). He is honorary professor at: Obuda University, Hungary; Guangdong University of Petrochemical Technology, China; Northeastern University, China; Muroran Institute of Technology, Japan; and Amity University, India.

**Lalit Garg** is a senior lecturer in computer information systems at the University of Malta, Malta, and an honorary lecturer at the University of Liverpool, UK. He has also worked as a researcher at the Nanyang Technological University, Singapore, and Ulster University, UK. His supervision experience includes more than hundred ninety masters' dissertations and four PhD thesis. He has edited two books and published over a hundred papers in refereed high-impact journals, conferences, and books, and some of his articles awarded best paper awards. He has delivered more than twenty keynote speeches in different countries, organized/chaired/co-chaired a similar number of international conferences. He was awarded research studentship in healthcare modelling to carry out his Ph.D. research studies in the faculty of computing and engineering at Ulster University, UK. His doctoral research was nominated for the Operational Research Society Doctoral Prize "Most Distinguished Body of Research leading to the Award of a Doctorate in the field of OR." He has also been consulted by numerous public and private organizations for their information system implementations. His research interests are missing data handling, machine learning, data mining, mathematical and stochastic modelling, and operational research, and their applications, especially in the healthcare domain.

# An Optimization View to the Design of Edge Computing Infrastructures for IoT Applications

**Thiago Alves de Queiroz, Claudia Canali, Manuel Iori, and Riccardo Lancellotti**

## 1 Introduction

A great variety of Internet of Things (IoT) applications have been developed in the last few years and are getting more and more popularity thanks to their capability of providing society and citizens with improved services and lifestyle. Many applications, ranging from autonomous driving, healthcare, smart cities up to industrial environments, increasingly exploit IoT-based services to support decision-making systems and take advantage of data-driven services [2, 45]. IoT applications are based on the presence of geographically distributed sensors to collect heterogeneous kinds of data about the surrounding environment. Such data are typically sent to a cloud computing data center to be processed using machine learning and Artificial Intelligence (AI) algorithms [31]. However, IoT applications are characterized by very different requirements, which will be pointed out in Sect. 2.

For some applications, the traditional cloud-based approach may not represent the most convenient choice. For example, for bandwidth-bound applications, the increasing volume of produced data is likely to make transferring and processing

T. A. de Queiroz
Institute of Mathematics and Technology, Federal University of Catalão, Catalão, Goiás, Brazil
e-mail: taq@ufg.br

C. Canali (✉) · R. Lancellotti
Department of Engineering "Enzo Ferrari", University of Modena and Reggio Emilia, Modena, Italy
e-mail: claudia.canali@unimore.it; riccardo.lancellotti@unimore.it

M. Iori
Department of Science and Methods for Engineering, University of Modena and Reggio Emilia, Reggio Emilia, Italy
e-mail: manuel.iori@unimore.it

 
S. Misra et al. (eds.), *Artificial Intelligence for Cloud and Edge Computing*, Internet of Things, https://doi.org/10.1007/978-3-030-80821-1_1

data at a remote cloud data center too expensive or not convenient for network constraints. Indeed, a high network utilization may be unwanted due to costs related to the cloud pricing options, even when the network load does not lead to a performance degradation. Other IoT applications have essential requirements related to latency and response time (e.g., real-time contexts) and cannot cope with a remote cloud data center's high network delays. In these cases, an *Edge Computing* paradigm is likely to represent a preferable solution [25].

The main feature of an edge computing infrastructure is a layer of edge nodes located on the network edge, close to the sensors, to host tasks aimed at pre-processing, filtering, and aggregating the data coming from the distributed sensors. The layer of edge nodes, placed in an intermediate position between sensors and the remote cloud data center, presents a twofold advantage. First, it may reduce the data volume transferred to the cloud through pre-processing and filtering performed on the network edge. Second, the intermediate layer of edge nodes reduces latency and response times for latency-bound applications. However, the increased complexity due to the introduction of the edge nodes layer opens novel issues concerning the infrastructure design.

The allocation of the data flows coming from sensors over the edge nodes and the choice related to the number and location of edge nodes to be activated represent critical aspects for the design and the management process of an edge computing infrastructure. While the problem of an optimized service placement has been largely investigated in terms of resource management within cloud computing data centers [3, 26], it received scarce attention in the edge computing area. This chapter focuses on this critical issue, proposing modeling the service placement based on a multi-objective optimization problem aiming at minimizing two aspects. First, the response time for the data transmission and processing along the path from sensors to edge nodes and then to the cloud data center; second, the (energy or monetary) costs related the number of turned on edge nodes. Due to the complexity related to the optimized design of an edge computing infrastructure that should cope with a non-linear objective function, we propose two approaches based on meta-heuristics, namely Variable Neighborhood Search (VNS) and Generic Algorithms (GA). We evaluate the performance of the two approaches over a range of different scenarios, considering the realistic setting of a smart city application developed in the medium-sized Italian city of Modena. Our experiments show that both heuristics represent promising approaches for designing an edge computing infrastructure, representing effective solutions for supporting IoT applications.

The rest of this chapter is organized as follows. Section 2 describes IoT applications and their classifications based on key requirements. Section 3 focuses on the literature review. Section 4 describes the main performance metrics used to evaluate the behavior of an edge computing infrastructure supporting IoT applications. In contrast, Sect. 5 formalizes the problem of designing and operating an edge computing infrastructure. Section 6 presents the two considered heuristics based on VNS and GA. Section 7 evaluates how the proposed algorithms can cope with the problem of designing and managing an edge computing infrastructure. Finally, Sect. 8 presents some concluding remarks.

# 2 IoT Applications: Classification and Challenges

IoT applications have recently experienced a remarkable diffusion in very heterogeneous fields, ranging from automotive to industry, health, and smart city applications. The reasons motivating this increasing popularity of applications based on a distributed set of sensors collecting data from the environment are multiple. First of all, thanks to the enhancements of the underlying technologies, the required technical equipment (principally the IoT sensors) is becoming increasingly powerful and efficient in terms of energy consumption, reducing its size at the same time, making possible its use in many contexts before not even conceivable. Simultaneously, the capability of sensors of storing and transmitting data has increased along with the availability of wireless connectivity in many physical environments.

In this section, we initially classify IoT applications based on their field of application. Then, we characterize their essential requirements to understand the challenges from the supporting infrastructure's point of view.

## *2.1 A Classification by Field of Application*

### 2.1.1 Automotive

The global automotive market around artificial intelligence is expected to grow up to $ 8887.6 million by 2025, with a compound annual growth rate of 45% from 2018 to 2025.[1] *Autonomous driving* holds the promise of reducing traffic fatalities, reducing congestion as well as curbing our carbon footprint. According to ABI, a marketing firm, roughly 8 million (10% of global output) vehicles with self-driving capabilities of level 3 or higher, will be shipped in 2025.[2] Connected cars and related city infrastructures also ensure fleet and vehicle health solutions based on collection of data coming from the vehicle, data management at the cloud level, and application of advanced analytics. Furthermore, the concept of *predictive maintenance* is progressively transforming into a practical and straightforward solution integrating vehicles sensors, hardware and software modules, and data transmitters, allowing the tracking of performance and eventual failure factors. Finally, value-added user services and applications are growing in the field of *infotainment*. A wide range of advanced services are offered by companies such as car manufacturers, insurance, service companies, and infotainment providers.

[1] https://www.alliedmarketresearch.com/automotive-artificial-intelligence-market.

[2] https://www.goldmansachs.com/insights/technology-driving-innovation/cars-2025/.

### 2.1.2 Industry 4.0

The interconnection of machines and IoT sensors able to gather useful data to provide real-time valuable information increases productivity, quality, and worker safety. It is expected that industrial markets will capture more than 70% of IoT value—estimated to exceed $4.6 trillion by 2025.[3] For example, leading institutions and firms in Europe have proposed the Reference Architecture Model Industry 4.0 (RAMI 4.0), describing key concepts and standards for this emerging paradigm [43]. *Industrial IoT* (IIoT) and Artificial Intelligence constitute the fundamental basis for the development and success of the so-called Industry 4.0: IIoT allows to continuously collect data from several and heterogeneous sensors and devices, and to forward the collected data to cloud computing data centers in a secure way. For these reasons, IIoT is able of supporting many Industry 4.0 applications, such as safety and security of industrial processes, quality control, inventory management, optimization of packing, logistics and supply chain, maintenance processes monitoring and effective detection of failures through the use of machine learning predictive techniques [45].

### 2.1.3 E-Health

The global market size related to IoT-based equipment for healthcare is projected to reach $ 534.3 billion by 2025, with at an annual growth rate of 19.9% over the period 2019–2025.[4] IoT sensors technologies have penetrated various healthcare applications, ranging from monitoring and management of chronic disease, support for home health, education for patients and professionals, up to applications for disaster management. In the field of *medicare interaction*, smart wearable devices are increasingly used to collect data about patients' health status. Significant examples of data that are likely to be collected in these applications are heartbeat, glucose level, and blood pressure. Such data are collected through sensors located on wearable technologies and are then sent to smartphones to be collected and analyzed. In this way, critical information about the medical condition of the patients is collected and afterward transmitted to a remote provider to be analyzed in order to provide the required care support. This approach allows the health care provider to remotely monitor a patient in his/her own home or in a care facility, thus reducing the readmission rates. Furthermore, the interactions between individuals and a provider can be aided with live video streams for consultative, treatment, and diagnostic services. Another sub-field of application is related to *health tracking*, in terms of health practice and education, that has been increasingly supported by mobile devices. E-Health IoT-based applications can range from continuous monitoring of

[3] https://www.forbes.com/sites/louiscolumbus/2017/12/10/2017-roundup-of-internet-of-things-forecasts/.

[4] https://www.reportlinker.com/p05763769/?utm_source=PRN.

health parameters or conditions, to targeted text messages for medicare, up to wide-scale alerts about disease outbreaks.

### 2.1.4 Smart Cities Applications

IoT applications are enabling Smart City initiatives worldwide. The possibility of collecting data by distributed sensors and analyzing them through artificial intelligence algorithms for predictive purpose or supporting decision-making processes opened a wide range of opportunities for creating innovative services to improve the lives of their residents [2]. Among the many smart cities' applications, we select the following ones as significant examples.

- *Mobility management*. Data collected from distributed sensors reveal patterns of public mobility and transportation. Mobility operators use these data to enhance the traveling experience and eventually improve safety and punctuality. Simultaneously, solutions for personal vehicles can determine the number, location, and speed of moving vehicles. Furthermore, smart traffic lights can automatically manage the lights according to real traffic conditions. Smart services for traffic management may also forecast, based on historical data, traffic evolution over time in order to prevent potential congestion. Finally, smart parking solutions can predict the occupation or the availability of parking spots for creating dynamic parking maps.
- *Energy and waste management*. The sustainable management of energy and waste in our cities is one of the most important field of IoT-based applications. The data collected by smart meters connected to the network can be sent directly to a public utility for further processing and analysis. In this way, utility companies are able to bill for the exact amount of energy, water, and gas consumed by each citizen. Consumption patterns of an entire city can be easily monitored too. Furthermore, waste management applications may facilitate the optimization of waste collection schedules through the accurate tracking of waste levels and analytics allowing route optimization.
- *Public safety*. Applications for public safety for smart cities involve collection, processing, and transmission of a heterogeneous and ever increasing amount of video sources, ranging from home surveillance systems, traffic monitoring, up to individual videos. In general, smart technologies deployment can provide surveillance with the capability to identify different entities, ranging from human beings, objects, events and detect patterns of behavior and anomalies, either real-time or post-events, from large bodies of video and audio streams. Another application is related to Public Warning Systems to alert citizen in a certain area and communicate them to take precautions or actions. In this case, IoT devices are connected to IoT service platforms to convey information to the citizens.

### 2.1.5 Retail

In the retail industry, IoT and artificial intelligence are applied with the final goal to support, guide, and improve the production of products for stores and businesses. Remote and accurate monitoring, and sales forecast represent the main benefits that IoT-based innovative solutions may provide to the retail business. Tracking and finalizing consumers' purchases and their product history and specific demographics provide retail consumers with heightened convenience. This trend also benefits retailers by achieving an improved understanding of their audience, thanks to the use of artificial intelligence algorithms on the collected data. A typical application in this field is related to *location-aware advertisement and navigation*. Bluetooth beacons provide shoppers with indoor and outdoor localization and information about shops nearby and help vendors focus their marketing campaigns on actual customer behavior. On the same line, IoT-led data tracking allows retail firms to get advanced metrics on the flow of people within the stores and the best selling points for individual customers. An innovative application consists of *smart mirrors*, which can be adopted within fitting rooms for suggesting other items based on what others have bought using an RFID label-scanning system. Smart mirrors are also used in conjunction with augmented reality, allowing customers to dress virtually without physically doing it.

### 2.1.6 Smart Agriculture

*Smart agriculture*, also called precision agriculture may significantly help farmers in order to maximize their productivity while minimizing costs and use of resources (e.g., water, seeds, and fertilizer). The deployment of sensors-based fields maps allows the farmers to better understand their farms at a micro-scale level of detail and to reduce the related environmental impacts. The core building block for the development of such smart agriculture are represented by Internet of Things (IoT) and artificial intelligence [8]. Specifically, the role of IoT is fundamental for the automatic collection of data that are subsequently transmitted to cloud data centers for processing. On the other hand, artificial intelligence solutions typically involving artificial neural networks and clustering techniques are then applied to process data to support decision-making. A typical example is related to the identification of the more appropriate amount of water necessary to irrigate fields, that is determined through the analysis of the collected agricultural data. Specifically, an action is taken in order to provide the correct quantity of water as soon as the irrigation need is detected, that is when the field lacks water.

## 2.2 Challenges of IoT Applications

The previously described applications show significantly different characteristics in terms of collected data, type of required processing, and purpose of the data analysis results. The application characteristics imply very different requirements from computational and networking points of view, leading to consequent challenges for the underlying supporting infrastructure. As an example, autonomous driving capabilities require real-time data transmission with strictly controlled latency margins to be safe: a typical requirement of the so-called level 4–5, that is high-level autonomous driving, is that each status message is delivered within 10 ms at maximum to enable appropriate and safe vehicle reactions [38]. On the other hand, medicare applications offering video streaming are more likely to be constrained by bandwidth availability. The near coming transition between closed-world, independent applications, and general-purpose infrastructures capable of supporting diverse applications without intervention requires hardware and software tools capable of finding adaptive trade-offs among different requirements. In the following, we introduce a taxonomy based on four categories, including the majority of IoT applications. Table 1 summarizes the results.

- **Latency-bound**: applications with strong requirements in terms of response times, as in real-time contexts. Examples of applications: mobility management, public surveillance, autonomous driving, and location-aware advertisement and navigation belong to this category.
- **CPU-bound**: applications characterized by computationally heavy tasks with high processing times where the CPU of the processing nodes is likely to become a bottleneck. Examples of applications: public safety, autonomous driving, infotainment systems, smart mirrors, and medicare applications are in this category.

**Table 1** IoT applications essential requirements

| Application | Latency | CPU | Bandwidth | Reliability | Statefulness |
|---|---|---|---|---|---|
| Autonomous driving | ✓ | ✓ | ✓ | ✓ | – |
| Predictive maintenance | – | – | – | – | ✓ |
| Infotainment | – | ✓ | ✓ | – | – |
| Industrial IoT | – | – | – | – | ✓ |
| Medicare interaction | – | ✓ | ✓ | ✓ | – |
| Health tracking | – | ✓ | – | ✓ | – |
| Mobility management | ✓ | – | – | – | ✓ |
| Utility/Waste | – | – | – | – | – |
| Public safety | ✓ | ✓ | ✓ | – | – |
| Location-aware Ads/Nav | ✓ | – | – | – | – |
| Smart Mirrors | – | ✓ | ✓ | – | ✓ |
| Smart agriculture | – | – | – | – | ✓ |

- **Bandwidth-bound**: applications characterized by a significant amount of data to be transferred, where the network bandwidth is an essential requirement for delivering services. Examples of applications: public safety, autonomous driving, infotainment, smart mirrors, and remote medicare interaction.
- **Reliability-bound**: applications where the reliability—intended as integrity, security, privacy, and availability—of data is a crucial requirement. Examples of applications: autonomous driving and e-health applications.
- **Stateful**: applications need to access past information for processing incoming data, hence data flows cannot be distributed over multiple nodes. Typically, applications in which aggregated views of the data are needed. Examples are mobility management, predictive maintenance, and smart mirrors.

## 3 Literature Review

As discussed in the previous section, an edge computing paradigm may efficiently handle data volume to be processed in IoT applications. In general, these applications require low latency and highly scalable services, motivating edge-based solutions in substituting the standard cloud computing paradigm. There is a vast literature showing the benefits of the edge computing infrastructure, especially when there is a large volume of data coming from geographically distributed devices [33, 35, 42, 44]. One considerable advantage of this architecture is decentralization, which allows pre-process data (for example, filtering and aggregation processing) before they reach the cloud data centers [19, 35, 39].

A survey on edge computing infrastructures was given in [39], who commented on applications and their challenges. Regarding IoT services, Wen et al. [35] discussed concepts and issues that may appear in complex scenarios, as, e.g., smart cities and marine monitoring. These authors handled a generic scenario by using genetic algorithms. Recently, a survey on the integration of edge computing with IoT was given by [28]. The authors discussed many challenges related to the heterogeneity, complexity, and dynamics of applications that need to be taken into account to deliver precise and reliable services. The authors also gave some insights into resiliency and the modeling by game theory. These studies foster the smart city application we are investigating in terms of cost reduction, load balancing, and latency reduction.

In general, the project of edge computing infrastructures is concerned with allocating services over the infrastructure. This was the case of [15], who handled issues related to the power consumption and transmission delay in the edge nodes to cloud data centers communication. On the other hand, the work of [40] modeled the load sharing issues in an edge-to-edge communication. In the context of industrial applications, Foukalas [18] presented a distributed intelligent IoT platform. The author used the cognitive IoT concept, with machine learning classifiers having an overall knowledge of the application, for monitoring and control purposes. Experiments were conducted on a scenario of predictive maintenance in smart factories, demonstrating the effectiveness of the proposed approach. Recently, Caiza et al. [9] discussed the main issues related to architecture, security, latency, and energy consumption in the context of IIoT. The authors commented on the vital role that edge computing plays to deal with these issues, which certainly include the fast processing and real-time storage of large volumes of data. Although our application may carry some similarities with these studies, our initial assumption considers a long-range communication network where multi-hop links exist between sensors. Then, each edge node can serve any sensor.

Our study considers an application to support smart city services and other studies also support this kind of application. For example, Tang et al. [33] proposed a system structured on four layers to handle it. Wang et al. [34] examined the coupling resource management problem, which is related to the multiple requisitions a sensor node is required to attend and then may result in failures of services. With the aim of an edge computing paradigm, the authors could efficiently introduce buffer and controller operations to obtain a sustainable system. In the survey of Zahmatkesh and Al-Turjman [42], the benefits of buffer/caching are discussed when edge computing is used in IoT communications. The authors investigated caching techniques using artificial intelligence and machine learning approaches, showing their potential and future challenges. Concerning the smart traffic monitoring, Dhingra et al. [16] used edge computing in a framework for congestion monitoring and traffic light management. The final results demonstrated a reduction in response time and an increase in the bandwidth. The proposed approach is also flexible, because it does not impose a fixed number of layers and let the number of edge nodes be determined according to the application's requirements.

Determining the number of edge nodes turned on and which sensors each of these nodes will serve is a typical facility location problem. This problem has been extensively explored in the operational research literature, proposing different algorithms to handle it. This problem aims at determining the number and where to open facilities (location decisions), besides which customers each open facility will serve (allocation decisions). The objective is to minimize the location and allocation decisions while satisfying the customers' demand [14].

Facility location problems were surveyed in [1, 13, 17, 22]. Farahani et al. [17] focused their review on multi-criteria problems, including bi-objective, multi-objective, and multi-attribute problems. They also commented on solutions methods. In the review of [22], the problems were classified according to the use of aggregation rules to handle situations with a large number of customers. The authors described aggregation error measurements, besides commenting on conditional location problems. On the other hand, Ahmadi-Javid et al. [1] focused their review on healthcare applications, presenting a framework to classify the problems following the emergency type and location management. The recent survey of Turkoglu and Genevois [13] considers service facility location problems, with a detailed classification of them based on purpose, space, distance, time, parameters, capacity, facilities, objectives, competition, application field, solution method, type of experiment, and other features.

Regarding IoT applications, Klinkowski et al. [24] handled the problem of data center locations with light-path provisioning in elastic optical networks. Silva and Fonseca [32] considered the optimization of human mobility using an edge computing infrastructure. The objective is to reduce the processing in the cloud data centers and the latency of users. Both [24] and [32] started dealing with mathematical models, but due to their inefficiency, they derived heuristic approaches. Recently, Canali, and Lancellotti [10, 11] considered the optimization of a service placement problem with the presence of edge infrastructures. In [10], these authors outlined the problem, while in [11], they presented a GA to examine a smart city application. On the other hand, the GA-based heuristic proposed in this chapter is a clear step ahead concerning these studies, because we introduce the possibility of locating edge nodes to reduce cost and power consumption, besides respecting a service level agreement.

## 4 Performance Modeling in Edge Computing Infrastructures

In this section, we present the main performance metrics that can be used to evaluate an edge computing infrastructure's behavior supporting IoT applications. We can distinguish two classes of metrics: efficiency metrics, related to the utilization of resources needed to operate, and performance metrics that concern the ability to operate effectively.

### *4.1 Efficiency Metrics*

An edge computing infrastructure needs resources to provide its services. Resources range from hardware components to energy to economic costs related to maintenance and administration. In the following of this analysis, we will refer in general to costs because all the resources needed to support IoT applications can be easily mapped into economic costs. Costs can occur both at deployment time or when running IoT applications on the infrastructure. An example of deployment time is the purchase of the hardware for the infrastructure and its installation. We usually refer to these costs as *Capital Expenditures* (CAPEX). Examples of costs related to the operation of the infrastructure are the energy cost and maintenance. We refer to them as to *Operating Expenditures* (OPEX).

Another way to classify the costs related to the edge computing infrastructure is to consider if the cost is global or related to its infrastructure. For example, costs related to setting up a control center for the edge computing infrastructure are not strongly related to the size of the infrastructure. On the other hand, the cost of the purchase and installation of edge nodes is directly correlated with the number of nodes to install. Similarly, energy costs are directly related to the number of active edge nodes.

The costs occur both when the infrastructure is deployed (usually, in this case, we consider the CAPEX related to the edge nodes that should be installed) and when the infrastructure operates (in this case, the focus is on OPEX such as energy costs for the nodes). It is worth noting that, when the load is subject to variations over time, the infrastructure can be designed to cope with the peak load; however, during off-peak hours, a fraction of the nodes can be turned off to reduce energy costs.

## *4.2 Performance Metrics*

Another class of metrics is related to the performance experienced by the IoT application. These metrics typically can be related to the response time when data flows are processed in the edge infrastructure or can concern the infrastructure's availability.

Among response time related performance metrics, the most common approach is to consider the average response time in a stationary scenario (when there is no transient effect due to sudden variations of the incoming load). An alternative metric is to consider a different estimation of the response time, typically taking into account 90-percentile or 95-percentile, representing a worst-case scenario excluding pathologically long request processing, for example, due to timeouts of transmission errors.

Availability-related metrics can take into account metrics such as service availability over a while (typical values range from 95% availability up to the 5-nines availability for critical services that must be online for 99.999% of the considered period). Another metric to measure a service's ability to process requests is considering the number of requests processed and the number of requests dropped, for example, due to the infrastructure's failures or due to overload in the edge nodes. To this aim, we define the *Throughput* as the number of requests processed in a considered period and the *Goodput* as the number of requests correctly processed in the same time frame. The difference between Throughput and Goodput measures the requests not processed (that is dropped) due to overload or other infrastructure problems. From this metric, we can infer the request drop rate, which is the percentage of requests that cannot be satisfied.

All the considered performance metrics can be used to define some *Service Level Agreement* (SLA) that defines the contract between the provider of the service and its users. An SLA can define a maximum acceptable response time (defined as a limit on the average or on some percentile of the response time), a minimum service availability that must be guaranteed, or a minimum drop rate. Failing to respect an SLA for the service provider usually results in a penalty that reduces the revenues.

# 5 Problem Formulation

We now formalize the problem of designing and operating an edge computing infrastructure supporting an IoT application. We assume the problem to be both CPU-bound and network-bound, where the network concerns may be either due to network latency or bandwidth issues. Furthermore, we assume our IoT application is a stateful application, where all data from an IoT device (sensor) must be sent to the same edge node for processing. We use as the basis of our model a facility location problem. We aim at (1) identifying a subset of potential edge nodes that should serve requests from IoT devices (referred to as sensors) and (2) mapping data flows from sensors to edge nodes and from edge nodes to cloud data centers to reduce cost and optimize performance.

If the problem concerns the infrastructure (infrastructure deployment) design phase, we consider a set of potential locations where edge nodes can be installed. We know an expected incoming load. Based on a performance model, we determine how many locations should be selected to host an edge node to satisfy SLA requirements while minimizing the infrastructure CAPEX. Similarly, during the edge infrastructure operations (infrastructure management), we can solve the same problem with different input data. In this case, we have a set of edge nodes, and we want to decide the minimum set of nodes to turn on to satisfy the SLA while minimizing the OPEX.

## 5.1 Model Parameters

To provide a model for the problem, we assume a stationary scenario. Let $\mathcal{S}$ be a set of geographically distributed IoT devices (sensors) producing data at a constant rate. The generic sensor $i$ produces data at a rate $\lambda_i$. An intermediate layer of edge nodes will process the data from the sensors. The edge nodes can perform operations such as filtering, aggregation, or anomaly detection with low latency. In our problem, we consider a set $\mathcal{N}$ of potential edge nodes locations. A decision variable $E_j$ is used to decide if edge node $j$ is to be used. If we consider the problem of infrastructure deployment, $E_j$ is used to decide if a potential edge node is to be actually deployed. In the problem of infrastructure management, the variable $E_j$ is used to decide if the edge node is to be turned on or off, based on present load conditions. Each edge node $j$ can process data at a rate $\mu_j$ and is associated with a cost $c_j$ that is either a representation of OPEX or CAPEX, depending on the scenario in which the problem is applied. Furthermore, the transmission of data between the sensors and the edge nodes occurs with a delay $\delta_{ij}$, with $i$ being the sensor and $j$ the edge node.

Finally, we consider the third layer of our architecture, that consists in set of cloud data centers $\mathcal{C}$. When data is sent from an edge node $j$ to a cloud data center $k$, it incurs in a delay $\delta_{jk}$.

**Table 2** Summary of notations used in the proposed model

| | |
|---|---|
| *Parameters of the model* | |
| $\mathcal{S}$ | Sensors set |
| $\mathcal{N}$ | Edge nodes set |
| $\mathcal{C}$ | Cloud data centers set |
| $\lambda_i$ | Outgoing data rate from sensor $i$ |
| $\lambda_j$ | Incoming data rate at edge node $j$; $\lambda_j = \sum_{i \in \mathcal{S}} x_{ij} \lambda_i$ |
| $1/\mu_j$ | Processing time at edge node $j$ |
| $c_j$ | Cost for deploying a potential edge node $j$ (or for keeping an edge node powered on) |
| $\delta_{ij}$ | Communication delay from sensor $i$ to edge node $j$ |
| $\delta_{jk}$ | Communication delay from edge $j$ to cloud $k$ |
| *Indices used in notation* | |
| $i$ | Index of a sensor |
| $j$ | Index of an edge node |
| $k$ | Index of a cloud data center |
| *Decision variables* | |
| $E_j$ | Enabling of edge node $j$ |
| $x_{ij}$ | Communication occurs between sensor $i$ and edge node $j$ |
| $y_{jk}$ | Communication occurs between edge node $j$ and cloud data center $k$ |

To considered problem uses the following binary decision variables.

- $E_j$, to define if a (potential) edge node located at position $j$ is available to process data from sensors.
- $x_{ij}$, to define if sensor $i$ is sending data to edge node $j$.
- $y_{jk}$, to define if edge node $j$ is sending data to cloud data center $k$.

The main symbols of the model are summarized in Table 2.

## 5.2 Objective Functions and SLA

The considered problem is based on the problem in [10] and takes into account two different performance metrics. The first performance metric is related to the infrastructure cost that we aim to minimize. Ideally, the lower is the number of edge nodes that we use, the lower is the global cost. The second performance metric is related to the average response time, and it plays a double role in our problem formulation. On the one hand, we consider an SLA on the average response time to guarantee that the response time remains below a given value. On the other hand, as long as the infrastructure's cost remains the same, we prefer solutions that are characterized by a lower average response time.

Concerning the infrastructure cost, we can define the total cost as

$$C = \sum_{j \in \mathcal{N}} c_j E_j \tag{1}$$

For the second performance metric, we should remember that response time is the sum of three contributions: $T_{netSE}$ that is the network delay related to the sensor-to-edge latency, $T_{netEC}$ that is the delay related to the edge-to-cloud latency, and $T_{proc}$ that is the time due to data processing on the edge nodes.

The network delay components can be modeled as follows:

$$T_{netSE} = \frac{1}{\sum_{i \in \mathcal{S}} \lambda_i} \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{N}} \lambda_i x_{ij} \delta_{ij} \tag{2}$$

$$T_{netEC} = \frac{1}{\sum_{j \in \mathcal{N}} \lambda_j} \sum_{j \in \mathcal{N}} \sum_{k \in \mathcal{C}} \lambda_j y_{jk} \delta_{jk} \tag{3}$$

For the $T_{netSE}$ definition in Eq. (2), we weight the delays $\delta_{ij}$ between each sensor $i$ and each edge node $j$ by the amount of traffic passing through that link, which is $\lambda_{ij} x_{ij}$. The sum is then normalized dividing by the total traffic $\sum_i \lambda_i$. In a similar way, we define $T_{netEC}$ in Eq. (3). The main difference is the use of the term $\lambda_j$ to refer to the load incoming into each edge node $j$ and forwarded from the edge node to the cloud layer. We can define $\lambda_j$ as

$$\lambda_j = \sum_{i \in \mathcal{S}} x_{ij} \lambda_i, \quad \forall j \in \mathcal{N} \tag{4}$$

The component concerning the processing time $T_{proc}$ is modeled using the queuing theory considering an M/G/1 system. According to the PASTA theorem [36] and to the Pollaczek-Khintchine formula used to describe the M/G/1 average response time [21], we have

$$T_{proc} = \frac{1}{\sum_{j \in \mathcal{N}} \lambda_j} \sum_{j \in \mathcal{N}} \lambda_j \left( \frac{1}{\mu_j} + \frac{\lambda_j \mu_j V_j^2}{2(\mu_j - \lambda_j)} \right) \tag{5}$$

where $V_j^2$ is the variance of the service time process. Assuming a variance of the service time close to the one of the inter-arrival time $(2/\mu_j^2)$, as in other example sin literature [4, 11], we can simplify Eq. (5) in

$$T_{proc} = \frac{1}{\sum_{j \in \mathcal{N}} \lambda_j} \sum_{j \in \mathcal{N}} \lambda_j \frac{1}{\mu_j - \lambda_j} \tag{6}$$

It is worth mentioning that we do not consider the cloud layer's details (such as the computation time at the cloud data center level) in the model of our problem. Indeed, this aspect is not meaningful for optimizing the edge infrastructure.

This model for the processing time is used also to describe the SLA for the edge computing infrastructure. Specifically, we expect the average response time to stay below $T_{SLA}$:

$$T_{SLA} = K\frac{1}{\overline{\mu}} + \overline{\delta}_{SF} + \overline{\delta}_{FC} \tag{7}$$

where $K$ is constant value (in cloud systems it is common to consider $K = 10$ [4]); $1/\overline{\mu}$ is the average service time in an edge node; $\overline{\delta}_{SE}$ and $\overline{\delta}_{EC}$ are the average sensor-to-edge and edge-to-cloud delays, respectively.

### 5.3 *Optimization Problem*

We can define the model for the edge computing infrastructure problem supporting an IoT application as follows:

Minimize :

$$C = \sum_{j \in \mathcal{N}} c_j E_j \tag{8}$$

$$T_R = T_{netSE} + T_{netEC} + T_{proc} \tag{9}$$

Subject to :

$$T_R \leq T_{SLA} \tag{10}$$

$$\lambda_j < E_j \mu_j, \quad \forall j \in \mathcal{N} \tag{11}$$

$$\sum_{j \in \mathcal{N}} x_{ij} = 1, \qquad \forall i \in \mathcal{S} \tag{12}$$

$$\sum_{k \in \mathcal{C}} y_{jk} = E_j, \qquad \forall j \in \mathcal{N} \tag{13}$$

$$E_j \in \{0, 1\}, \quad \forall j \in \mathcal{N} \tag{14}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in \mathcal{S}, j \in \mathcal{N} \tag{15}$$

$$y_{jk} \in \{0, 1\}, \quad \forall j \in \mathcal{N}, k \in \mathcal{C} \tag{16}$$

The two objective functions, (8) and (9), are related, respectively, to the minimization of: **cost**, which depends on the number of used edge nodes; and **response time**, which is the delay in sensor-edge-cloud data transit expressed through the function introduced in the previous section. The response time objective is subordinated to the cost one, meaning that we aim to minimize (9) as long as the improvement for this objective function does not affect (8).

The model includes the following constraints. Constraint (10) places a limit for the average response time: this value must not violate the *Service Level Agreement* (SLA). Constraints (11) guarantee that no overload occurs on the edge nodes. Hence, the incoming data rate on every node must be lower than the processing rate. For a node that is powered down, no processing occurs. Constraints (12) ensure that exactly one edge node will process the data of each sensor. In a similar way constraints (13) guarantee that exactly one cloud data center receives the data for each edge node. Finally, constraints (14), (15) and (16) describe the binary nature of the decision variables.

## 6 Heuristics

Several options are available when we aim to solve a problem using a heuristic algorithm. On the one hand, greedy heuristics are usually quite fast. On the other hand, the performance of a greedy algorithms depends on the nature of the problem. Local minima and a non-convex domain, which may hinder their ability to find a solution for the problem. In designing an edge computing platform, the objective function is non-linear. Furthermore, the feasibility domain of the problem is not guaranteed to be convex. For this reason, we explore solutions that differ from the classical greedy approach.

Due to the number of edge nodes and sensors, the problem is characterized by a high dimensionality that may hinder the performance of branch and bound approaches, due to the large solution space to explore.

For these reasons we focus on meta-heuristics that provide a flexible approach to the problem. The ability of these meta-heuristics to solve successfully a broad and heterogeneous set of problems is known in literature [7]. In particular, we consider two approaches, namely Variable Neighborhood Search (VNS) and Generic Algorithms.

### *6.1 Variable Neighborhood Search*

The *Variable Neighborhood Search* (VNS) is a meta-heuristic used to solve hard non-linear and combinatorial optimization problems. Some examples include its application to vehicle routing [37], portfolio selection [5], cutting and packing [30], scheduling [29], among others.

VNS is a single solution-based method, where a solution passes through a shaking and local search phases to become globally optimal concerning all neighborhood structures. These structures represent how to explore the current solution's