IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 30 October 2024, accepted 14 November 2024, date of publication 18 November 2024,
date of current version 29 November 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3501775

SURVEY

Emerging Trends in Machine Learning Assisted Optimization Techniques Across Intelligent Transportation Systems

BLESSING ITORO AFOLAYAN[1], ARKA GHOSH[1], JENNY FAJARDO CALDERIN[1], AND ANTONIO D. MASEGOSA[1,2]

[1]DeustoTech, Faculty of Engineering, University of Deusto, 48007 Bilbao, Spain
[2]Ikerbasque, Basque Foundation for Science, 48009 Bilbao, Spain

Corresponding author: Arka Ghosh (arka.ghosh@deusto.es)

This work was supported in part by the Spanish Ministry of Science and Innovation through the REsearch on New Artificial Intelligence techniques to Improve Sustainability, SAfety and resilieNCE of mobility (RENAISSANCE) project under Grant PID2022-140612OB-I00; in part by Basque Government through Research under Grant IT1564-22, Grant KK-2023/00012, and Grant KK-2023/00038; and in part by the Horizon Europe Research and Innovation Program through the project SYNCHROMODE (Advanced Traffic Management Solutions for Synchronized and Resilient Multimodal Transport Services) under Grant 101104171.

ABSTRACT Artificial intelligence (AI) plays a critical role in Intelligent Transport Systems (ITS) as urban areas grow by processing data for safety enhancements, predictive analysis, and traffic management. This results in better traffic control, lower emissions, and preventative actions to lessen the effects of accidents. Despite these developments, there isn't a thorough academic analysis that covers a variety of optimization strategies for transportation AI models. By presenting an in-depth analysis of AI optimization methods and their uses in ITSs, this work seeks to close this knowledge gap and give academics important new information on possible directions for future research. Model-based optimization approaches, reinforcement learning techniques, model-predictive control techniques, and generative AI techniques are the four areas into which this study divides AI optimization techniques for the sake of structure, clarity, and comparative analysis. Subcategories of optimization techniques and their corresponding applications are explored, and each category is thoroughly addressed. Researchers will be better able to comprehend the state of AI optimization for transportation management today and in the future thanks to this methodical methodology. The most cutting-edge optimization methods created in the last five years are summarized in this review. This work acts as a compass for future research initiatives targeted at developing scalable and adaptable AI solutions for transportation management by identifying common approaches and highlighting research needs.

INDEX TERMS Artificial intelligence, generative AI, model predictive control, model-based optimization, reinforcement learning, intelligent transport systems.

I. INTRODUCTION

The transportation industry, being an integral part of economic and social development, has significantly benefited from AI advancements. Optimizing transport management is crucial for enhancing efficiency, reducing costs, and improving user experiences. This review paper aims to provide a comprehensive overview of the emerging trends in optimization techniques applied in transportation systems. In the last decade, due to the advance in technologies, various aspects of transportation model optimization have been published. Reference [1] conducted a comprehensive review on noteworthy contributions made in the applications of smart technologies in improving logistics operations and transportation network efficiency. 84 papers were reviewed and classified into Autonomous, Smart Logistics and Smart Warehouse. The challenges in optimization with the applications of smart technology were explained alongside future

The associate editor coordinating the review of this manuscript and approving it for publication was Razi Iqbal.

direction. Also, [2] conducted a systematic literature review synthesizing the state-of-the-art of AI applied in railway traffic planning and management from four perspectives, i.e., the intersection between AI research fields, techniques, applications, related disciplines and rail Traffic Planning and Management (TPM). Furthermore, [3] discussed on the Simulation, Optimization, and Machine Learning in Sustainable Transportation Systems (STS). Based on the planning horizon, the application of optimization in STS was divided into short-term (operational), medium-term (tactical), and long-term (strategic) decisions. The authors also categorized STS into urban transportation systems and regional/global transportation systems based on the size of the area. For supply chain model optimization, [4] gave a comprehensive review that synthesizes current practices and future potentials of leveraging AI for supply chain optimization. For surveys on the optimization models for electric vehicle service operations and shared mobility optimization algorithms, we have [5] and [6] respectively.

To sum up, Abduljabbar et al. [7] gave a holistic overview of AI techniques applied worldwide to address transportation problems mainly in traffic management, traffic safety, public transportation, and urban mobility. The overview concluded by addressing challenges and limitations as well. However, it lack the analysis of the optimization methods used. Also, In terms of technological advancement, the period from 2019 to 2024 is highly significant. There is a dearth of a good literature review paper that should cover the literature published in these years regarding AI transport model optimization. Furthermore, this review is different from the reviews that only considered specific area optimization [1], [5] and [6] focusing mainly on model optimization for supply chain, electric vehicles and shared mobility respectively. These surveys concentrates on a single objective rather than concurrently optimizing across a variety of characteristics. As a result of this, it is abundantly evident that there is a need for an all-encompassing review that takes into account a range of optimization models in a manner that is consistent, clear and organized.

In response to the above points, this paper proposes a comprehensive review torching across different four categories of transport AI model optimization, as well as detailing the challenge or motivation, proposed solution, model or optimization techniques used, limitations and future research of some reviewed papers on each of these categories (Model-based optimization methods, Reinforcement learning methods, Model-Predictive Control method, and Generative AI method) are also discussed. We conducted a search in Google Scholar by using these keywords: transportation management, transportation model optimization, model based optimization, reinforcement learning in transportation, transportation and model predictive control, generative AI in transportation and have identified several related papers within the last 5 years. Categorizing these optimization techniques into four groups helps to organize the different optimization methods based on their underlying principles and applications within ITS, making it easier for readers to understand the different approaches without getting overwhelmed by a large amount of detail. By grouping similar techniques together, it also serves as a guide in comparing different techniques, highlighting their strengths, weaknesses, and specific contributions to transportation optimization.

Lastly, this review paper addresses key research questions, including: (1) What are the most effective AI techniques currently being used for transport network optimization?. (2) How do these different AI methodologies compare in terms of efficiency, scalability, and optimization methods?. (3) What are the existing research gaps, and what directions should future research take to address these gaps?

Fig. 1 shows the structure of this paper, it is organized as follows. Section II presents the four optimization techniques categories. Followed by Section III where we have the applications of these techniques in ITS. This section has 2 subsections where we categorized major ITS problems into 5 groups and then we also mapped these 5 groups to it's corresponding applied optimization class. In Section IV, we discussed the scalability and computational trade-off for AI optimization methods. Section V highlights it's challenges in Real-World applications. Section VI discussed about the metrics for scalability and efficiency for these four categories of optimization methods and finally, lastly, Section VII concludes the article by summarizing the review process and by discussing how the research question that we presented in last paragraph were satisfied by this review work.

II. CATEGORIES OF OPTIMIZATION TECHNIQUES

In this sections, we delve into the four categories of AI model optimization techniques in detail: model-based optimization techniques, reinforcement learning methods, model predictive control methods, and generative AI methods. For each category, we discuss the methodologies, advantages, and disadvantages comprehensively.

A. MODEL-BASED OPTIMIZATION TECHNIQUES

Model-based optimization (MBO) is a critical component in contemporary modelling, simulation, and optimization activities. Recognized as an exceptionally effective method for addressing real world optimization challenges that are both cost-intensive and time-consuming, MBO leverages the use of a more efficient surrogate model to evaluate scenarios, which can lead to substantial savings in time, space, and computational resources [8]. MBO is particularly relevant in engineering fields [9]. The importance of model-based optimization lies in its ability to reduce computational time for problems where the solution space and the complexity of the model make traditional recursive methods too computationally expensive and, hence, infeasible. For example, meta-models can provide quick solutions to operational problems that require decision-making in a short time-frame or promote a greater understanding of the optimization process. This makes meta model-based optimization particularly valuable

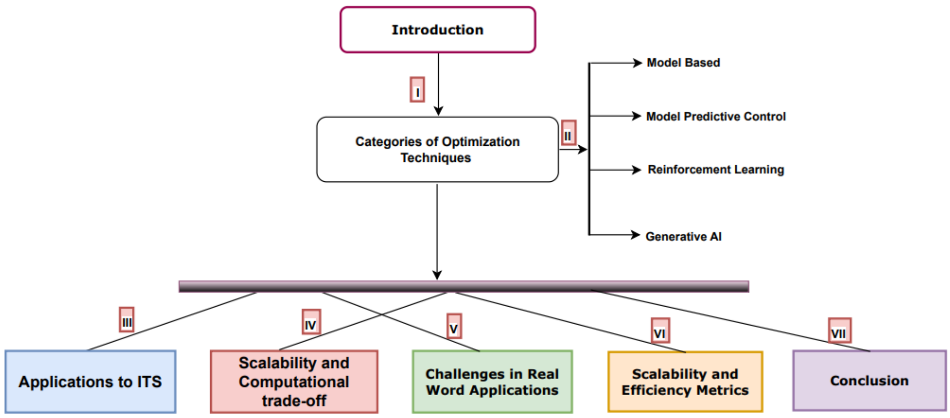


FIGURE 1. Flow of review summary.

in scenarios where quick, efficient, and accurate decision-making is crucial, such as in industrial processes, healthcare, logistics, and traffic management.

1) MAIN CATEGORIES OF METHODOLOGIES WITHIN MODEL-BASED OPTIMIZATION

In order to describe the main categories of model-based optimization methodologies that can be found in the literature, we will base ourselves on the taxonomy proposed in [8]. According to this work, we can differentiate between the following categories and sub-categories:

- Distribution-based optimization
- Surrogate model-based optimization
  - Single Surrogate
  - Multi-fidelity surrogate
  - Ensemble surrogate

Below are the detailed explanations of each categories

1) Distribution-based optimization algorithms: This algorithms are a class of algorithms used in optimization and machine learning that focus on modelling and sampling from probability distributions. Essentially, these algorithms work by maintaining and updating a probability distribution over the search space. Over time, this distribution evolves to increasingly favour regions of the search space that contain high quality solutions. The key principle behind distribution-based algorithms is that they do not just search for a single optimal solution. Instead, they explore a range of solutions by sampling from a probability distribution. This approach allows them to navigate large, complex search spaces effectively compared to other methods. There are several types of distribution-based algorithms, each with its own specific mechanisms and applications. Some of the most common examples of distribution-based algorithms are:
   - Estimation of Distribution Algorithms [10]: They explicitly build and update probabilistic models of promising solutions. They use these models to generate new candidate solutions for the optimization problem
   - Bayesian Optimization [11]: This method uses Bayesian techniques to model the objective function and makes decisions based on this probabilistic model. It's particularly useful for optimization problems where function evaluations are expensive.
2) Surrogate-based optimization algorithms: Surrogate-based optimization algorithms are advanced computational methods used to efficiently solve complex optimization problems, particularly those where direct evaluations of the objective function are computationally expensive or time-consuming [12], [13], [14], [15]. The core idea of surrogate-based optimization is to create a surrogate model, an approximate representation of the actual objective function. This surrogate model is computationally cheaper to evaluate and is constructed using a limited number of evaluations of the true objective function. The optimization process in surrogate-based algorithms typically involves two main steps: building the surrogate model and then optimizing this model. Initially, a few samples of the objective function are taken, often using design of experiments techniques. These samples are used to construct the initial surrogate. The optimization algorithm then iterates between updating the surrogate model with new samples and finding the optimum of the current surrogate. Three variants of surrogate-based optimization algorithms have become increasingly popular in recent years [8]: single-surrogate models, multi-fidelity surrogate models and ensemble models. We review each of these variants below:
   - Single-surrogate models: These optimization algorithms are a type of surrogate model-based optimization where only one model is used to accelerate the search process. One of the most known approaches within single-surrogate models is Surrogate Assisted Evolutionary Algorithms (SAEAs) [16], which is a variation of evolutionary algorithms that utilise meta-models to approximate

the fitness function. The surrogate models or meta-models that have been used in SAEAs can be classified into the following categories:

- Absolute Fitness Models: These models aim to directly predict the fitness function values of candidate solutions. They are further divided into:
  (1) Regression-based Models: These models use regression techniques to model the relationship between inputs (candidate solutions) and outputs (fitness values) [16]. (2) Similarity-based Models: They approximate fitness based on the similarity or correlation between unevaluated and evaluated individuals [17].
- Relative Fitness Models: These models focus on estimating the rank or preference of candidates rather than their absolute fitness values. They are subdivided into:
  (1) Rank-based Models: These models predict the relative rank of individuals in a population based on evaluated samples [18]. (2) Classification-based Models: These models categorise individuals based on a comparison result with a reference solution, determining whether they are 'better' or 'worse' [19].

- Multi-fidelity surrogate models: Multi-fidelity surrogate models [20] are algorithms that integrate different levels of fidelity, i.e., detail and accuracy, to achieve a balance between computational efficiency and accuracy in predictions. These models combine high-fidelity models (HFMs), which offer detailed and accurate results but are computationally expensive, with low-fidelity models (LFMs), which are less detailed but computationally cheaper. MFMs are particularly useful in applications where high-precision modelling is required but are restricted by computational resources. There are two main types of MFMs: Multi-Fidelity Surrogate Models (MFSMs) [21] and Multi-Fidelity Hierarchical Models (MFHMs) [22]. MFSMs use algebraic surrogates to correct LFMs based on HFM predictions, while MFHMs combine different fidelities based on specific criteria without constructing a surrogate model. Within these categories, various methods like additive and multiplicative corrections, comprehensive corrections, and space mapping are used to integrate different fidelity levels. The choice of method depends on the specific characteristics of the problem at hand. Some of the most commonly used models for Multi-fidelity surrogate optimization are Support Vector Regression Models [23], which map high and low-fidelity samples into a high-dimensional space using a kernel function and then apply a linear model to determine the input-output relationship, co-kriging models [21], that extend the concept of Kriging or Gaussian process regression to multiple levels of fidelity, modeling the correlation between different fidelity levels; or deep learning approaches [22], [24], [25], particularly multi-fidelity deep neural networks (MFDNN), are used to capture complex, nonlinear relationships between different fidelity levels.
- Ensemble-based surrogate models: An ensemble surrogate model is a composite model that combines multiple surrogate (or proxy) models to make predictions or perform analyzes. Each surrogate model in the ensemble is designed to approximate the same underlying system but may use different modelling techniques, assumptions, or data subsets [8], [12]. These surrogates might include various types of models like polynomial regressions, kriging models, radial basis functions, neural networks, or any other statistical or machine learning method capable of capturing the behaviour of the system of interest. Unlike multi-fidelity models, which integrate models of varying levels of fidelity, classical ensemble methods typically combine data-driven models of similar or identical fidelity. The key idea behind ensemble surrogate models is to leverage the strengths of multiple modelling approaches to achieve greater accuracy and robustness than what might be obtained from any single surrogate model. By combining models, the ensemble can compensate for individual weaknesses and capitalise on unique strengths. Ensemble-base surrogate models can be categorised according to the type of methodology used to build the ensemble. In this way, we can find the following categories: 1) Bagging [26], [27], which generates multiple data subsets through bootstrapping, training a model on each, and averaging their outputs; 2) Boosting [28], which sequentially trains models to focus on previous errors, with the final prediction being a weighted sum of all models; 3) Stacking [29] which trains multiple models on the same data, then uses a new meta-model to combine these predictions; 4) Wrapping [30] which employs a wrapper algorithm to select the best combination of models and features, often using cross-validation; and 5) Gradient Boosting [31], which builds models sequentially, with each new model correcting the errors of its predecessor, guided by the gradient of the loss function.

2) DISCUSSION ON MODEL-BASED OPTIMIZATION METHODS

One of the key strengths of MBO approach is its adaptability. The ability to update and adapt

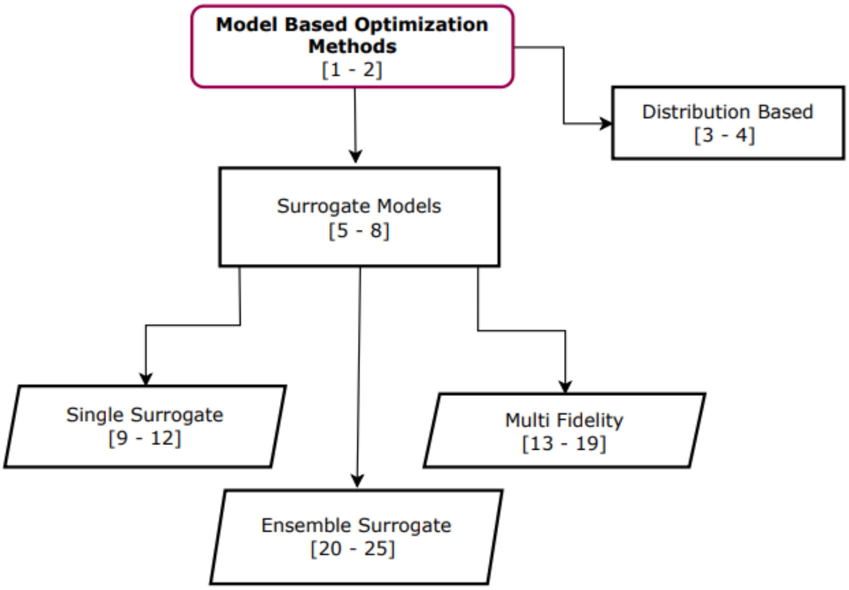


FIGURE 2. Taxonomy for model based optimization methods.

the model to reflect changes in the environment ensures that the optimization process remains relevant and effective over time. Moreover, the process of creating these models often leads to a deeper understanding of the system being optimised. This not only generates valuable insights into the system's internal dynamics but also helps in understanding the interrelationships within it. Another significant benefit is the robustness of a well-designed model. Such models are adept at finding solutions that are resilient to uncertainties and variations in the environment, providing a structured way to handle unpredictability and enhancing the reliability of the optimization results. However, the success of model-based optimization heavily depends on the accuracy of the model. If the model doesn't accurately reflect the real-world system, the results might be sub-optimal. Building an accurate model is particularly challenging for complex systems, as capturing all relevant factors becomes increasingly difficult with the complexity of the system. Furthermore, the computational cost associated with this approach can be considerable, especially for intricate models and optimization problems, which might limit its feasibility in time-sensitive or resource-limited scenarios. Regularly updating the model to mirror changes in the system is also essential for maintaining its effectiveness, but ensuring a smooth and accurate updating process without disrupting ongoing operations can be a challenge. Additionally, since models often rely on assumptions about the system, there is a risk that if these assumptions are incorrect or oversimplified, and the optimization results may not be applicable or reliable in real-world scenarios. Fig. 2 shows a taxonomy of model-based optimization techniques.

B. REINFORCEMENT LEARNING OPTIMIZATION TECHNIQUES

Reinforcement Learning (RL) [32] is a paradigm of machine learning that focuses on how agents should take actions in an environment to maximise some notion of cumulative reward. The fundamental principle of RL is centred on the concept of agents learning to make decisions by interacting with their environment, where these interactions are framed in terms of states, actions, and rewards. RL has been successfully applied in various domains, including robotics, gaming, healthcare, finance, and autonomous vehicles. The objective in RL is to find a policy, which is a mapping from states to actions, that maximises the expected cumulative reward over time. This objective often involves balancing the exploration of uncharted territory in the state-action space (exploration) against the exploitation of current knowledge to gain higher rewards (exploitation). Key components of RL include [32]: 1) Agent: the learner or decision maker; 2) Environment: The external system with which the agent interacts; 3) State: A representation of the status of the environment; 4) Action: An intervention or decision made by the agent; and 5) Reward: A scalar feedback signal indicating the success of an action. In RL, the agent learns a policy through trial-and-error interactions with the environment. The learning process involves evaluating the consequences of actions and refining the policy to improve the expected rewards.

1) MAIN CATEGORIES OF METHODOLOGIES WITHIN REINFORCEMENT LEARNING OPTIMIZATION

Although other categories of RL methods can be found in the literature, we have focused on the four categories that, in our opinion, have the best applicability to the area of Transport Network optimization. In the following subsections, we will review each of these categories:

- Continual RL
- Multi-agent RL
- Off-line RL
- Hierarchical RL

1) Continual Reinforcement Learning: Continual Reinforcement Learning (CRL) [33] is an advanced paradigm within the field of machine learning that focuses on the development of learning agents capable of adapting over a continuous stream of experiences. CRL extends the principles of traditional reinforcement learning by emphasising the agent's ability to learn incrementally across a non-stationary distribution of tasks. This approach is particularly aimed at addressing real-world scenarios where the environment's dynamics are subject to change over time, and the learning process itself cannot be confined to a fixed set of tasks or datasets. CRL encompasses strategies for retaining previously acquired knowledge, minimising catastrophic forgetting, and efficiently adapting to new situations and tasks as they arise. Continual Reinforcement Learning (CRL) approaches can be broadly categorised into three main categories [33]:
   - Explicit Knowledge Retention: This approach emphasises the importance of preserving knowledge gained from previous experiences. This is critical in continual learning environments to mitigate the issue of catastrophic forgetting, where the acquisition of new knowledge leads to the erosion of previously learned information. Within this strategy, various techniques are employed: 1) Latent Parameter Storage [34], which involves maintaining essential parameters in a latent form that encapsulates past learning; 2) Distillation-Based Methods, which focus on extracting and preserving the crucial knowledge components from a model; and 3) Rehearsal-Based Techniques [35] simulate or replay past experiences to reinforce previous learning.
   - Leverage Shared Structure strategy: This approach is focused on the identification and utilisation of common structures across different tasks or environments and promotes retention and transfer across the lifetime of an agent. Some of the methodologies proposed to leverage this shared structure are 1) Modularity and Composition [36], which aims to address the challenge of creating machines capable of compositional generalisation. Compositional generalisation involves using past experiences to solve variations of previous problems or even more complex issues that are new to the agent; 2) State Abstractions [37], which aims at identifying and harnessing common structures across different tasks, thereby potentially enabling the effective forward transfer of knowledge among related tasks; 3) Skill-Focused Approaches [38], which represent learning strategies that bypass the necessity of making decisions at every single time step; 4) Goal-Focused Methods [39], which aims to deal with complex problems by concentrating on reasoning based on goals, which can be defined as a state the agent want to reach, a specific reward target for the agent, or the endpoint of a skill; and 5) Auxiliary Task Focused Techniques [40], which aim at learning representations that encapsulate the fundamental, task-independent dynamics of the world. Learning to Learn: Also known as meta-learning, it is about enhancing the agent's ability to learn more effectively over time. This category includes Context Detection [41], where the agent learns to recognise and adapt to different contexts or environments; Learning to Adapt [42], which aims at imparting a bias that enhances the agent's efficiency in learning new behaviours with fewer samples; and Learning to Explore [43], which involves developing strategies to provide the agent with the intrinsic motivation to explore and acquire new knowledge efficiently.

2) Multi-agent Reinforcement Learning: Multi-agent reinforcement learning (MARL) [44] is characterised by the interaction of multiple agents within a shared environment. These agents simultaneously learn to make decisions, often with varying objectives. The environment in MARL scenarios is typically non-stationary from the perspective of any individual agent due to the concurrent learning and actions of other agents. This non-stationarity poses a significant challenge for algorithm design and convergence to stable policies. MARL can be categorised based on the nature of interactions among agents [44], which can be fully cooperative, fully competitive, or cooperative-competitive. Below, we describe each of them:
   - Fully cooperative multi-agent RL [45] involves scenarios where multiple agents work together towards a common goal. In this context, agents are designed to collaborate, sharing information and strategies to optimise a collective reward function. The challenge in fully cooperative MARL lies in efficiently coordinating the actions of all agents to achieve optimal outcomes. This requires agents to not only learn from their individual experiences but also to consider the actions and learning processes of their peers.
   - Fully competitive multi-agent RL [46] involves scenarios where multiple agents operate in an environment with the goal of maximising their

own individual rewards, often at the expense of other agents. In this setting, agents are in direct competition with each other, leading to a zero-sum or adversarial situation. Each agent aims to optimise its own performance without cooperation or communication with other agents.
- Cooperative-competitive multi-agent RL [47] is a hybrid approach where agents in a shared environment exhibit both cooperative and competitive behaviours. In this framework, agents may form alliances or compete against each other, depending on the context and objectives of the scenario. This approach models complex real-world situations where entities have overlapping but not identical goals. Agents must learn to optimise their strategies through both collaboration with some agents and competition against others.

3) Off-line Reinforcement Learning: Offline Reinforcement Learning (ORL) [48] is defined as a data-driven formulation of the traditional reinforcement learning (RL) problem. In ORL, the primary objective remains the optimization of a specified goal, similar to standard RL. However, a significant deviation is that the agent in ORL does not interact with the environment for additional data collection through a behaviour policy. Instead, ORL employs a static dataset of transitions D, which the learning algorithm utilises to derive the best possible policy. This approach aligns more closely with the framework of standard supervised learning, where D serves as the training set for the policy. The core challenge in ORL is for the learning algorithm to acquire a comprehensive understanding of the dynamics of the Markov Decision Process (MDP) solely from this fixed dataset. Subsequently, the algorithm must construct a policy π that aims to achieve the highest cumulative reward when interacting with the MDP. The Dataset D is presumed to consist of state-action tuples that are sampled according to the distribution of states and actions. The categories of methodologies proposed so far in literature for ORL are the following [48]:
- Offline Evaluation and RL via Importance Sampling [49]: These methods are important for evaluating the return of a given policy or for estimating the policy gradient in offline variants of policy gradient methods. A common approach in RL involves the use of importance sampling to estimate the expected return of a policy. This estimation is carried out using trajectories sampled from a behaviour policy. This process is termed as "off-policy evaluation."
- Offline RL via Dynamic Programming [50]: These methods focus on learning a value function to derive the optimal policy or estimate policy returns. For example, offline adaptations of basic Q-learning and policy iteration methods have been adapted by initialising with a non-empty buffer and setting collection steps to zero. Such approaches, including deep RL variants, have been effective, but additional online fine-tuning can significantly enhance performance over training solely on logged data. This class of ORL methods faces challenges due to distributional shifts when only offline data is used. Solutions are categorised into policy constraint methods, which keep the learned policy close to the behaviour policy to reduce distributional shift, and uncertainty-based methods, which estimate the uncertainty of Q-values to identify distributional shifts.
- Off-Policy Model-Based RL [51]: Model-based RL algorithms can be adapted for offline settings by training with offline data, with minimal changes to the algorithm itself. These methods, which include uncertainty estimation to limit model exploitation, are effective in ORL. Notably, model-based methods have demonstrated excellent performance in both conventional off-policy and offline RL settings. Recent works have also explored the use of high-capacity predictive models and hybrid approaches combining model-free and model-based elements for complex tasks like mobile robot navigation. Additionally, recent advancements in model-based ORL include conservative model-based algorithms. These methods aim to provide performance bounds by adjusting the learned model to encourage conservative behaviour. This involves penalising the policy for exploring areas where the model's accuracy is questionable.

4) Hierarchical Reinforcement Learning: Hierarchical Reinforcement Learning (HRL) [52] is a variant of RL that focuses on decomposing complex, long-horizon tasks into simpler subtasks. This is achieved by organising the learning process into a hierarchy of policies, each operating at a different level of abstraction and timescale. The higher-level policies in this hierarchy select subtasks, which themselves can be RL problems to be solved by lower-level policies. This structuring into a hierarchy allows for more efficient learning and problem-solving in environments where considering the entire action space at once would be impractical or infeasible due to its complexity or size. The concept of HRL is beneficial for dealing with long-horizon tasks and large state-action spaces, where traditional reinforcement learning approaches might struggle due to their inherent complexity. The main typologies of HRL methods that we can find in the literature are the following:
- Learning Hierarchical Policies (LHP) [53]: In this HRL approach, the lower-level policies are typically predefined and handle specific, detailed tasks. The upper levels are responsible for broader

strategic decisions. This methodology is effective in scenarios where basic tasks are well-understood but complex, higher-order planning is required. The primary challenge in LHP is to orchestrate these predefined tasks efficiently to achieve complex objectives. It is particularly useful in environments where foundational actions or decisions are established, but coordinating them towards complicated goals presents a significant challenge.

- Learning Hierarchical Policies in Unification with Sub-task Discovery (UNI) [54]: It is a comprehensive HRL methodology that seeks to concurrently identify sub-tasks and develop hierarchical policies. This dual process of discovery and learning is pivotal in situations where the division of tasks into sub-tasks is not predefined. The UNI approach facilitates an integrated learning experience, enabling a system to dynamically uncover task structures while simultaneously developing strategies to solve them. It is particularly beneficial in complex environments where task decomposition and strategic learning need to occur in an adaptive and cohesive manner.
- Independent Sub-task Discovery (ISD) [55]: This HRL approach focuses on the autonomous identification of sub-tasks independent of any specific main task. This approach primarily involves a pre-training phase, where useful sub-tasks are discovered through exploration and analysis of the environment. These sub-tasks are task-agnostic, meaning they are not tailored to any particular task but are rather general in nature. Once these sub-tasks are identified, they can be applied in learning hierarchical policies for different tasks. The key advantage of ISD lies in its ability to identify versatile and reusable sub-tasks, which can facilitate learning across various scenarios and domains. This strategy contributes to the flexibility and adaptability of HRL systems, allowing them to efficiently handle a range of tasks by leveraging pre-identified sub-tasks.
- Multi-Agent HRL (MAHRL) [56]: This category extends HRL into the realm of cooperative multi-agent systems. Therefore, this methodology also focuses on learning to coordinate among multiple agents, each potentially possessing its own hierarchical structure. The fundamental challenge in MAHRL lies in enabling these agents to effectively collaborate and learn in a shared environment, often towards a common goal or in pursuit of aligned objectives. Each agent in MAHRL is typically equipped with a hierarchy of policies, where decisions and actions at various levels of abstraction must be synchronised with those of other agents. This synchronisation is crucial to ensure coherent and efficient behaviour across the multi-agent system. MAHRL is highly relevant in complex, real-world scenarios where multiple entities must work in unison, such as in robotics, autonomous vehicles, and distributed systems.
- Transfer Learning with HRL (TransferHRL) [57]: This approach addresses the challenge of transferring knowledge and skills acquired in one task to improve or accelerate the learning in another. This approach leverages the hierarchical organisation of tasks and sub-tasks in HRL to transfer learned policies, sub-tasks, or other relevant information across different but related tasks. The objective is to enable the HRL system to utilise previously gained experience, thereby reducing the need for learning from scratch in new tasks. This is particularly useful in complex environments where similar tasks or structures recur, allowing for more efficient adaptation and learning across varied challenges. Fig. 3 shows a taxonomy of reinforcements learning optimization techniques alongside each of it's classifications and sub categories.

2) DISCUSSION ON REINFORCEMENT LEARNING OPTIMIZATION METHODS

The core of its advantages of RL lies in its flexibility and adaptability, as RL empowers systems to learn from experiences and adapt to new circumstances without the need for explicit programming. This characteristic renders RL particularly suitable for dynamic and ever-changing scenarios. Moreover, RL excels in optimizing complex decisions. It proves highly effective in situations where decisions are sequential and interconnected, optimizing decision-making processes in intricate scenarios. Another key benefit of RL is its ability to handle uncertainty and incomplete information. By continuously learning and updating strategies based on feedback from the environment, RL models demonstrate a robust capability in managing unpredictable situations. This adaptability extends to its versatility, with RL finding applications across various domains. This diversity in application showcases RL's potential in addressing a wide range of problems. However, alongside these benefits, RL faces several challenges. One significant challenge is sample efficiency. RL often requires extensive data or experience to learn effectively, posing a hurdle in real-world scenarios where data collection is costly or time-consuming. Additionally, the exploration versus exploitation trade-off presents a critical balancing act in RL. Determining the optimal balance between trying new actions to discover better strategies (exploration) and leveraging known successful strategies (exploitation) is especially challenging in complex environments. The design of reward functions and the issue of sparse rewards also pose significant challenges. Creating an appropriate reward function that accurately reflects the objectives of a task is essential. Sparse or poorly designed

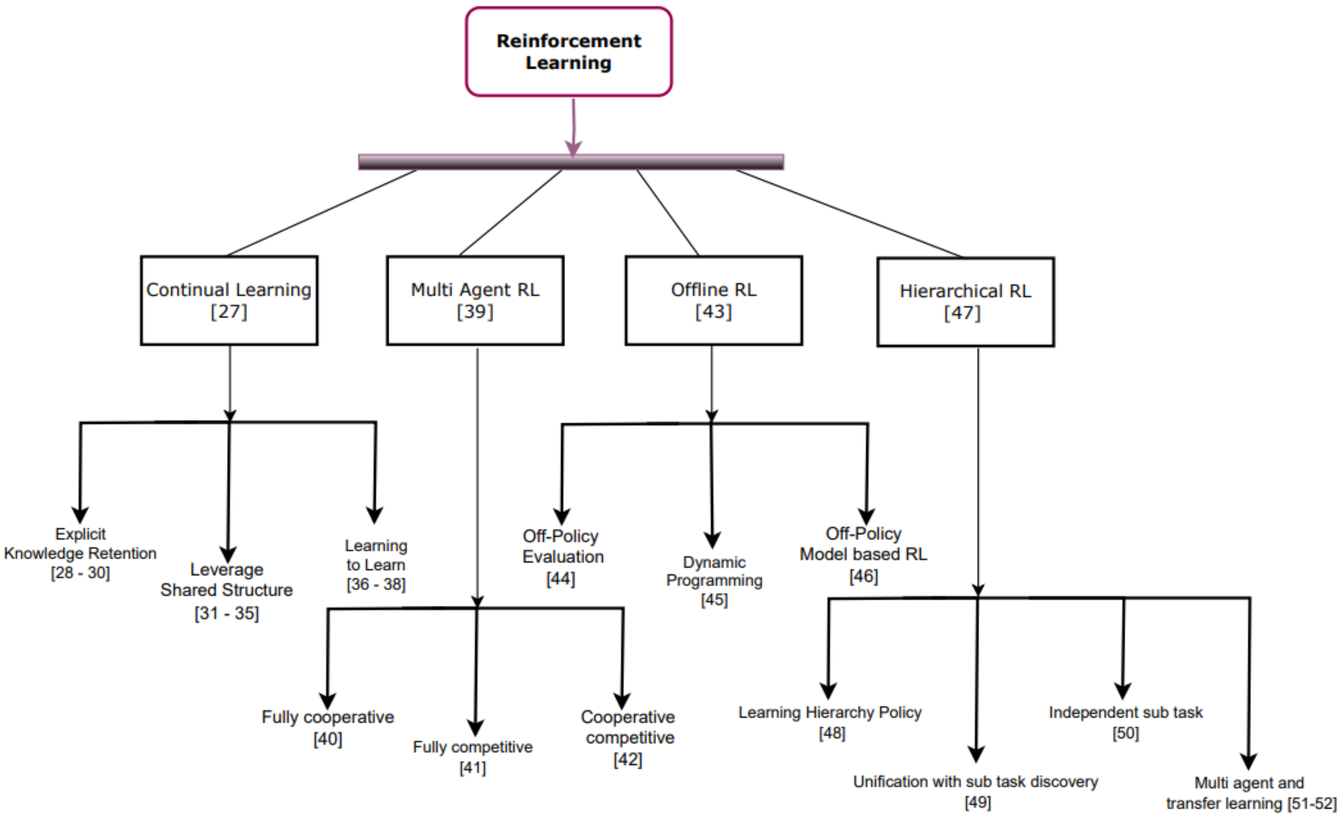


FIGURE 3. Taxonomy for reinforcements learning optimization methods.

rewards can hinder learning, making it inefficient or even ineffective. Furthermore, generalisation and transfer learning are challenging aspects of RL. Generalising knowledge learned in one environment to perform well in new, unseen environments remains a hurdle. Transfer learning, where knowledge from one task is applied to enhance learning in another, is a key area of ongoing research. Safety and ethical concerns are also paramount when deploying RL systems in the real world. Lastly, the computational complexity of RL algorithms, particularly in large-scale or intricate environments, can limit their applicability in real-time or resource-constrained settings. This complexity presents a barrier to the broader adoption of RL in various applications. Despite these challenges, the field of RL, in our opinion, offers a wide range of advantages and a high potential for applicability in Transport Network Management.

C. MODEL PREDICTIVE CONTROL OPTIMIZATION TECHNIQUES

Learning-Based Model Predictive Control (LBMPC) [58] is a novel control strategy that combines the robustness of model predictive control (MPC) with the adaptability of learning mechanisms. At its core, LBMPC is defined by its utilisation of a predictive model that is continuously updated based on data acquired during the system's operation. This fusion of predictive control with learning mechanisms allows LBMPC to effectively handle complex, nonlinear systems that are subject to uncertainties and changing environments. However, learning and data-based adaptations in MPC extend beyond improving the predictive models. On the one hand, they also use this learning capability for designing and enhancing controllers. This enables it to ensure safety and stability in control applications, which is critical in many industrial and technological contexts.

1) MAIN CATEGORIES OF METHODOLOGIES WITHIN LEARNING-BASED MODEL PREDICTIVE CONTROL

To describe the main categories of LBMPC methodologies, we will use the taxonomy outlined in [58]. According to this reference, the following categories of methodologies can be distinguished:

- Learning the System Dynamics within MPC
- Learning Controller Design within MPC
- MPC for Safe Learning

Below, we describe each of these categories in details

1) Learning the System Dynamics within MPC: Traditional MPC relies on preestablished parametric prediction models derived offline using physical principles and system identification techniques. This approach includes robust or stochastic MPC to handle uncertainties and disturbances, with models and uncertainties predefined and fixed offline. In contrast, learning-

based MPC [59] adapts dynamically, incorporating real-time data such as state measurements. Learning-based MPC techniques often differentiate between a nominal system model and an additive term that accounts for uncertainty. They usually assume that control based on the nominal model is feasible if the appropriate data can be safely collected from the real system. Furthermore, these methods typically do not differentiate between real system dynamics and predictive dynamics, assuming no model mismatch when the appropriate parametrization of the model is established. This category of LBMPC can, in turn, be classified into those using robust paradigms and stochastic techniques [58]. A more detailed definition of both is provided below:

- Robust Models: Robust MPC [60] schemes ensure that constraints are satisfied under uncertainty. These schemes often split the model into a nominal part and an additive uncertainty component, assuming the uncertainty lies within a defined set. Then, the controllers are designed to be robust against this uncertainty. Many LBMPC approaches focus on estimating model uncertainty directly from data. These techniques aim to adjust the uncertainty set over time to reduce conservatism. In this line, we can distinguish between parametric models [61], which seek parameter values consistent with observations, and non-parametric models [62], which estimate the function directly from observed data points and establish function bounds.
- Stochastic models. Unlike Robust MPC, which relies on hard bounds covering all uncertainty, which can be conservative. In contrast, stochastic MPC [63] uses distributional information about uncertainty without relying on hard bounds, although this makes theoretical analysis more challenging and often leads to less rigorous results. Again, we can differentiate between parametric and non-parametric models. In this case, parametric approaches are usually based on scenario optimization techniques [64] and non-parametric on Gaussian Process Regression [65].

2) Learning Controller Design within MPC: This type of LBMPC technique uses learning mechanisms to design the MPC problem to achieve a desired controller behaviour by using models for approximating the cost function and/or the constraints. These approaches can be divided into two categories [58]:

- Performance-driven controller learning: This approach aims to enhance closed-loop performance by inferring the parametrisation of the MPC problem that minimises the difference between the closed-loop performance and the optimal control problem [66]. This inference can be addressed in two ways. The first one [67] consists of solving a black-box optimization problem in which the controller is a smooth function (e.g. Gaussian Processes) and a black box optimizer iteratively adapts the parameters of the smooth function to optimise the true closed-loop cost. Bayesian optimization is one of the most common approaches [68]. The second one addresses the learning of the terminal components making use of collected data [67], [69]. Terminal components refer to the terminal cost and constraints used at the final prediction step. They guide the system toward a desired terminal state or behaviour, addressing the limitation of a short prediction horizon. By incorporating these terminal components, MPC can effectively control the system over a finite prediction horizon while considering long-term objectives and promoting stability and convergence in the system.
- Learning from Demonstration with Inverse Optimal Control: This approach leverages observed data from demonstrations to design an automatic controller [70]. In MPC, traditional controllers are often designed based on mathematical system models. However, inverse optimal control takes a different path by learning from real-world demonstrations of desired control behaviours. In this context, demonstrations involve observing how an expert or desired control system performs a specific task or responds to varying conditions. Instead of explicitly modelling the system dynamics, inverse optimal control aims to capture the underlying control strategy from these demonstrations and translate it into the parameterisation of the MPC problem. A related field is Inverse Reinforcement Learning [71], a technique for identifying cost or reward functions in probabilistic decision-making. This method helps interpret optimality conditions probabilistically, typically using likelihood maximisation, but is usually limited to discrete state and action spaces without considering system constraints. However, recently, they have been expanded to continuous state spaces by introducing a probabilistic control objective [72].

3) MPC for Safe Learning: Ensuring safety in learning-based control is very challenging, particularly in the context of RL, where safety constraints may not always be guaranteed during the learning process [73]. To address this, safety frameworks have emerged, combining model-based control for safety assurance (model predictive safety filters) with learning-based controller methods to optimise overall performance. This approach separates constraint satisfaction handled by MPC from performance optimization through learning-based methods, primarily due to the complexity of stochastic optimal control problems. MPC techniques

can then be used as safety filters to transform a safety-critical system into a safe one. The main function of a Model Predictive Safety Filter [74] is to assess the control inputs generated by the learning-based controller and verify that they adhere to predefined safety constraints. If the proposed control inputs are deemed unsafe, the safety filter modifies them or computes a safe backup trajectory to bring the system back into a safe state. This adaptive approach allows the system to operate robustly even in the presence of disturbances or uncertainties, providing an added layer of safety assurance. Fig. 4. shows a taxonomy of MPC optimization techniques alongside each of it's classifications and sub categories.

2) DISCUSSION ON LEARNING-BASED MODEL PREDICTIVE CONTROL

Learning-based MPC is a promising approach at the intersection of control theory and machine learning, offering several significant benefits and addressing unique challenges in control system design. One of the main advantages of learning-based MPC is its adaptability. Traditional MPC relies on accurate system models, which can be challenging to obtain for complex and dynamic systems. In contrast, learning-based MPC leverages data-driven models, allowing it to adapt to system changes and uncertainties. This adaptability enables the controller to perform effectively in real-world scenarios where model accuracy may vary. Another benefit is the ability to handle high-dimensional and nonlinear systems. Learning-based MPC can effectively control systems with complex dynamics, making it suitable for applications such as robotics, autonomous vehicles, and industrial processes. It can learn intricate control policies that might be challenging to derive analytically. Furthermore, learning-based MPC can incorporate safety constraints and optimise performance simultaneously. By learning from data and adapting to real-world conditions, it can strike a balance between achieving control objectives and ensuring safety. This feature is vital in applications where safety is critical. However, learning-based MPC faces its share of challenges. One of the primary issues is the need for large amounts of data for training, which can be impractical in some cases. Additionally, ensuring the controller's safety during the learning process and handling exploration of the control space without violating constraints remains a challenge. Moreover, interpretability and guaranteeing theoretical properties are less straightforward in learning-based MPC compared to traditional control methods. Understanding the learned control policies and providing formal guarantees on system stability and performance can be complex.

D. GENERATIVE AI OPTIMIZATION TECHNIQUES

A subclass of artificial intelligence systems known as "Generative AI" or "Generative Artificial Intelligence" is defined by its ability to create original content that is similar to the training data. These systems are based on generative models, which are algorithms designed to identify and reproduce patterns in data, allowing for the production of creative results. In optimization theory, generative AI-based techniques are essential instruments because they provide creative solutions for the complexities of high-dimensional and challenging optimization issues. Neural network-based Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) exhibit impressive potential for examining various solution spaces and identifying non-linear patterns in data. The use of generative models is noteworthy in the field of combinatorial optimization, where GANs show promise in producing excellent solutions for challenging combinatorial problems. Generative AI is important because it can improve global optimization by minimising the chance of getting stuck in local optima through effective navigation of intricate terrain. This capacity satisfies the requirements of a number of fields, including supply chain management and logistics, where traditional approaches might have trouble locating global optima. Furthermore, hybrid methods that combine the best features of both paradigms for more reliable and adaptable optimization processes are produced by fusing generative AI with conventional optimization techniques. Recent studies have shown how generative AI improves sample efficiency, resilience, and flexibility in optimization, highlighting the technology's significance in developing the area.

1) MAIN CATEGORIES OF METHODOLOGIES WITHIN GENERATIVE AI OPTIMIZATION

- Generative AI applied to surrogate optimization
- Generative AI as sample generation for optimization

1) Generative AI methods for surrogate creation - (Building the surrogate model): This presents a novel approach to the optimization of difficult-to-manage complicated simulations, particularly those that exhibit random or uncertain behaviour. In domains such as engineering and physics, these simulations are frequently difficult to comprehend or control. This approach makes use of sophisticated mathematical models that are able to imitate those simulations in tiny, controllable regions through learning. The researchers assert that by doing this, they can more effectively determine the ideal simulation parameters, even in situations when other widely used approaches falter. In this section, we discuss the application of generative-based methods in the creation of the black-box function of such sophisticated mathematical models or surrogates. The article [75] offers a novel strategy for overcoming the excessive variance observed in score function gradient estimators while utilising the benefits of gradient-based optimization. By parameterizing the surrogate model with pertinent simulator parameters, this approach uses deep generative models as differentiable surrogate models to approximate non-differentiable simulators. This allows for the approximation of the stochastic behaviour of

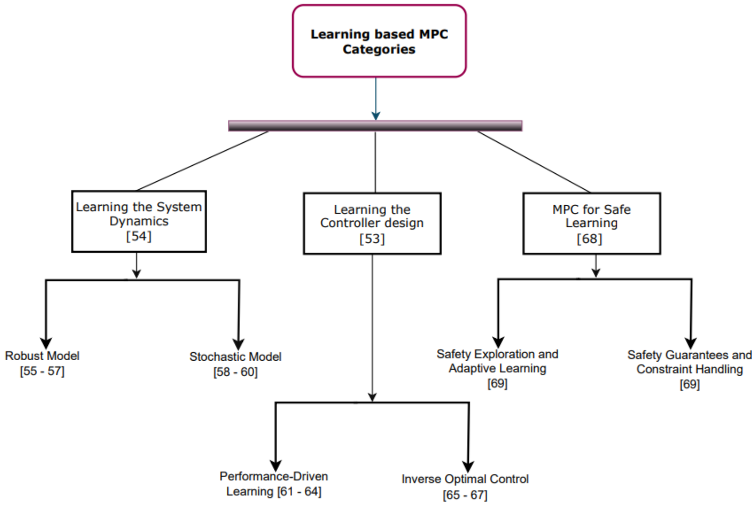


FIGURE 4. Taxonomy for model predictive control optimization methods.

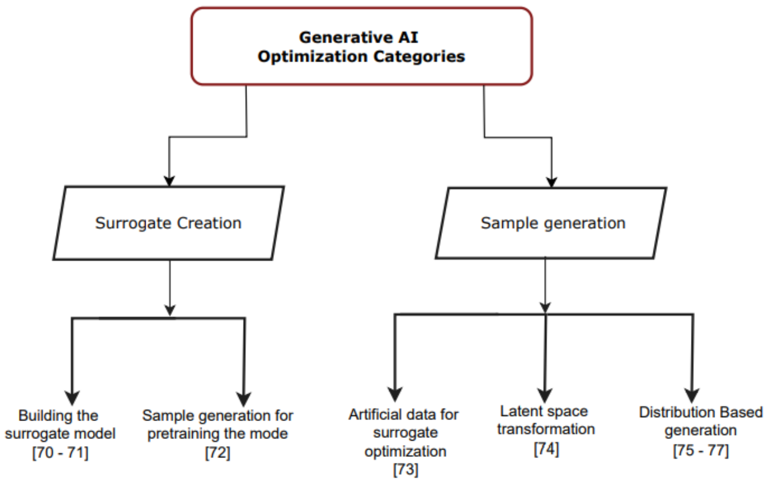


FIGURE 5. Taxonomy for generative AI optimization methods.

the simulator and allows for direct gradient-based optimization of an objective. The proposed technique, called local generative surrogate optimization (L-GSO), makes strategic use of successive local neighbourhoods of the parameter space at each step of parameter optimization. This is especially useful in dealing with high-dimensional parameter spaces, where training surrogates over the entire parameter space can be computationally expensive if follows the work flow of the generative adversarial networks to conduct the optimization process. The article [76] explains a novel regression model based on beta-variational Autoencoders (betaVAEs) that optimises oilfield development decisions. optimization studies based on reservoir modelling are commonly used to guide decisions in the oilfield development environment. The computing cost of these simulations, which compare various production scenarios and well controls, is high. Surrogate models are frequently employed to expedite this research. Even if they work well, traditional deep-learning models for creating surrogates have drawbacks, such as an inability to quantify prediction uncertainty and a lack of interpretability. These problems are addressed by the suggested beta-VAE-based regression model. Regression is performed using the interpreted, factorised representation of choice variables in a latent space that is provided by Beta-VAE. The model includes probabilistic dense layers that help with approximate Bayesian inference and measure prediction uncertainty.

2) Generative AI methods for surrogate creation - (Sample generation for pretraining of the model): Apart from the objective function creation for the underlying surrogate, generative AI-based methods can also aid in synthetic data point or sample generation for the creation of the surrogate process. In simple terms, when

a lack of original samples affects the pre-training phase of the surrogate machine learning model, generative AI-based methods can be used to generate synthetic samples for pretraining purposes. In this section, we discuss such endeavors of generative AI-based methods. An empirical study [77] on the applicability of key dimensionality reduction methods for building surrogate models in high-dimensional optimization situations is included in the text. The goal of the research is to construct surrogate models more effectively by condensing intricate, high dimensional design spaces into compact, low-dimensional representations. This work investigates four methods for reducing dimensionality: Principal Component Analysis (PCA), Autoencoders (AEs), Kernel Principal Component Analysis (Kernel PCA), and Variational Autoencoders (VAEs). The quality of the low-dimensional surrogate models generated by these methods is assessed, and they are tested in various situations with varying dimensionalities, benchmark problems from continuous optimization, and surrogate modelling methods such as Kriging and Polynomials. The study's findings demonstrate the efficacy of PCA and autoencoders in terms of global optimality and modelling accuracy, respectively. Due to their historical importance in machine learning applications such as feature learning, dimensionality reduction, data compression, and data visualisation, PCA and AEs were selected. In order to create effective surrogate models, kernel PCA, a non-linear extension of PCA, and VAEs, non-linear stochastic encoding of the data space, were included. With the support of a thorough quality evaluation of the associated low-dimensional surrogate models (LDSMs) across a wide range of test scenarios, the paper attempts to offer a fresh viewpoint on the suitability of these dimensionality reduction techniques (DRTs) in surrogate-assisted optimization (SAO).

3) Generative AI methods for sample generation: Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) are two examples of generative AI techniques that are skilled in producing artificial data samples for surrogate optimization. Real data is used to train these models so they may discover the underlying trends. After being taught, they can produce new, artificial samples that replicate the dataset's statistical characteristics. These artificial samples are essential for surrogate optimization. They are employed in the creation of a surrogate model that mimics a sophisticated, frequently computationally demanding simulation. Because it uses readily created synthetic data rather than constantly executing the actual simulation, this model training is significantly more efficient. After that, the surrogate model aids in optimization by offering fast estimates for a range of input parameters and assisting in the discovery of the best configurations. To ensure the accuracy of the surrogate, genuine data from the actual simulation is periodically used to modify this process. For this reason, surrogate optimization becomes much more efficient thanks to generative AI techniques, which allow for quick and affordable exploration and optimization of complicated systems without requiring a lot of simulation or real-world testing. In this section, we discuss the application of a generative AI based method for the synthetic sample generation process to aid the surrogate validation and training phases. It can be seen from the literature that this sample generation can be performed in two main ways, first by latent space transformation and second, distribution-based. In the coming subsections, we discuss these two approaches for sample generation for the surrogate optimization process. The article [78] talks about the use of Variational Autoencoders (VAE) in offline model-based optimization, especially in situations where the purpose is to maximise a black-box objective function with a static dataset of designs and the scores that go along with them. Numerous fields, including material design, robotics, DNA sequencing, and protein engineering, frequently encounter this optimization dilemma. The traditional method uses the static dataset to train a Deep Neural Network (DNN) as a proxy function. Gradient ascent is then applied to the current designs to find possibly high-scoring ones. Unfortunately, the out-of-distribution problem frequently arises with this approach, which results in inaccurate design predictions. BiDirectional learning for offline Infinite-width model-based optimization (BDI) is the suggested remedy to this problem. The two mappings that make up the BDI are the forward mapping, which predicts scores for high-scoring designs using the static dataset, and the reverse mapping, which predicts scores for the static dataset using the high-scoring designs. In contrast to earlier research that frequently disregarded backward mapping, BDI recognises its importance in enhancing the high-scoring designs by extracting additional information from the static dataset, thus reducing the out-of-distribution issue. The article highlights that there is a considerable decline in design quality for finite-width DNN models because the loss function of the backward mapping is intractable and only has an approximate shape. In order to get over this restriction, an infinite-width DNN model is used, and a closed-form loss for more precise design updates is produced by using the associated neural tangent kernel. This decision is meant to improve the optimization process's accuracy in offline model-based settings.

4) Generative AI methods for sample generation - (based on latent space transformation): By altering the latent space, generative AI techniques such as variational autoencoders (VAEs) and generative adversarial networks (GANs) can be used in surrogate optimization

to produce synthetic examples. The most important aspects of the data are captured in a compressed, lower-dimensional form known as the latent space. These generative models acquire the ability to map actual data to and from the latent space through training on real data. The generative model explores this latent space to generate new data samples for surrogate optimization. In order to create new, synthetic samples, this is accomplished by slightly altering points in the latent space and then converting these points back into the high-dimensional space of the original data. These samples are unique in their details, yet they preserve the original dataset's statistical characteristics. These artificial samples are then used to train the surrogate model, which enables an effective approximation of a resource intensive, complex process. Generative AI techniques provide more effective and efficient optimization procedures by generating diverse and informative samples for surrogate models through the exploration and modification of the latent space. With the use of latent space manipulation, this method produces a large range of synthetic samples that can greatly increase the surrogate model's capacity for learning and generalisation, which in turn improves the optimization process. In this section, we discuss the application of generative-based methods for latent space transformation-based sample generation for surrogate optimization. This paper [79] presents a generative solution to a classic mechanical materials design problem by integrating Long Short-Term Memory (LSTM) neural networks with Variational Autoencoders (VAE). Virtual Autoencoders (VAEs) are known for their ability to extract low-dimensional representations from large and complicated datasets, especially for generative applications like rebuilding complex data like images. On the other hand, LSTM neural networks are quite good at figuring out logical trajectory correlations inside datasets. The developers of this combination method, known as VAELSTM, concentrate on a cantilever design-related compliance optimization challenge. An LSTM is used to learn trajectories inside this latent space that correlate to the optimization process after a VAE is used to encode cantilever structures into a 2D latent space. The cantilever design space can be clearly shown thanks to the final model, which also makes it possible to create new, incredibly low-density ideas outside of the initial dataset. Additionally, the method makes it easier to extract the best cantilever structures that are modelled after natural phenomena. Crucially, 3D printing can be used to produce the created designs, offering a quick turnaround time from concept to prototype. According to the article, this technique can be used for other image-based datasets that record changes due to various circumstances. Intelligent design and manufacturing of material structure difficulties are noted as areas where the interpretability of complicated behaviour through representations in a simplified space is recognised as having substantial potential applications.

5) Generative AI methods for sample generation - (Distribution based): Generative AI-based models can comprehend how data is distributed across features since they are initially trained to grasp the underlying probability distribution of a specific dataset. Following this learning phase, the model's sample points from this learnt distribution to create fresh, synthetic samples, thereby producing data that is a statistical mirror of the training set. The diverse and representative samples produced by this procedure are essential for developing a reliable and accurate surrogate model. In this section, we discuss the application of Generative AI methods for distribution-based learning for synthetic data generation for the surrogate optimization process. In data-driven multi-objective optimization problems (DD-MOPs), where the amount of data accessible from actual engineering trials is constrained by time and expense, the article [80] presents a novel solution to these difficulties. Traditionally, using trained surrogate models, evolutionary algorithms indirectly solve DD-MOPs. Unfortunately, the accuracy of the estimated Pareto front that is obtained frequently deteriorates significantly due to a lack of practical data. The study presents two cutting-edge tactics that a Generative Adversarial Network (GAN) uses to combat this. The first technique, called critical fitness, presents a new critical fitness that is made up of the prediction value of the surrogate model and the critical score obtained from the discriminator of the GAN. The second tactic is data augmentation, in which fresh samples are produced by the GAN to enhance the surrogate model's training. Through the integration of both techniques, the GAN fulfils the dual function of augmenting data and improving critical fitness, hence effectively overcoming the issues associated with DD-MOPs. In this article [81], a unique way to use Generative Adversarial Networks (GANs) to improve the performance of model-based evolutionary algorithms is presented. To operate at their best, traditional model-based evolutionary algorithms mostly depend on the quality of the training set. But when the problem gets bigger, the curse of dimensionality becomes a concern, and performance deteriorates quickly. The suggested approach makes use of a multiobjective evolutionary algorithm powered by GANs to solve this problem. To train the GANs, parent solutions are first divided into real and false samples in each generation. Then, using the trained GANs, progeny solutions are sampled. Because GANs are powerful generative algorithms, the suggested technique can produce plausible offspring solutions even in high-dimensional decision spaces with a small amount

of training data. Experiments on ten benchmark issues with up to 200 choice variables demonstrate the algorithm's usefulness and indicate that it can overcome the problems brought on by the curse of dimensionality while still exhibiting increased performance. The article [82] presents a brand-new image-based route planning algorithm that aims to overcome the drawbacks of the conventional sampling-based approaches used in robot path planning. This method makes use of a generative adversarial network (GAN), in contrast to uniform sampling strategies that explore the state space without complex geometric modelling of the configuration space. Without any further pre-processing, the environment map–represented as an Red Green Blue (RGB) image is fed into the GAN. The result is also an RGB image with a segmented promising section highlighted that probably contains a workable path. This promising region addresses the issues of delayed convergence to the optimal solution and initial solution quality by acting as a heuristic for attaining non-uniform sampling in the path planner. The suggested method's higher performance in terms of initial solution quality and convergence speed as compared to conventional approaches is validated by simulation experiments. Notably, the approach demonstrates strong performance in settings that differ markedly from the training context, underscoring its adaptability and versatility. Fig. 5 shows a taxonomy of model generative AI optimization techniques.

2) DISCUSSION ON GENERATIVE AI APPLIED TO OPTIMIZATION

There are various benefits of using generative AI-based techniques for creating surrogates. Synthetic sample generation is fast and efficient, allowing for a quick process of creating surrogates and minimising the need for expensive experiments or resource-intensive simulations. This affordability is especially useful in fields where it may not be feasible to conduct real-world testing. Because generative models are naturally able to identify a wide range of patterns in the training data, the resulting surrogate model is guaranteed to generalise effectively over a wide range of scenarios and input changes. Furthermore, these techniques provide a useful approximation for complicated systems that are difficult to represent explicitly. Because the procedure is iterative, the surrogate model can be continuously improved, increasing its relevance and accuracy over time. In addition to enabling focused parameter space exploration that supports effective optimization, generative AI offers a controlled environment for the creation of a variety of synthetic samples in situations involving restrictions or control challenges. Finally, the capacity of surrogate models trained on synthetic data to be transferred to other contexts improves their versatility, which makes generative AI a powerful tool for optimizing a range of intricate systems and procedures. Utilising strategies such as variational autoencoders (VAEs) and generative adversarial networks (GANs), these approaches enable the creation of synthetic samples by exploring the latent space, a lower-dimensional representation of the data that captures important properties. This method offers a sophisticated investigation of the underlying data distribution by adjusting points in the latent space, which makes it easier to create representative and varied samples. The capacity to produce samples in the latent space offers a way to methodically investigate various configurations, enabling the generation of more focused and educational data. Additionally, it provides a productive means of enhancing datasets for surrogate model training, especially in situations where real-world data is scarce. Latent space transformation's versatility makes it possible to create fresh, insightful samples, which helps in the development of surrogate models that can faithfully represent intricate relationships found in the data. Using generative AI in latent space transformation improves synthetic sample creation's overall effectiveness, diversity, and informativeness, which helps with surrogate model training and optimization procedures. Although using latent space transformation to generate samples for surrogate optimization via generative AI presents significant benefits, there are still several obstacles to overcome. One challenge is the intricate nature of generative models, such as GANs and VAEs, which require careful tuning to appropriately represent the complex data distribution because of their complexity and training requirements. It is still difficult to guarantee that produced samples are diverse and accurately reflect the features of real data; problems such as mode collapse in GANs result in surrogate models that are biased. Interpretability and explainability of the models are limited by the black-box character of some generative techniques. Two persistent issues are avoiding over-reliance on synthetic samples and generalisation to unexpected events. Furthermore, the ethical issues surrounding the unintentional replication of biases in generated data highlight the need for ongoing research to improve the robustness and moral application of generative AI in surrogate optimization. The computational resources required for training complex generative models can also be a limiting factor.

III. APPLICATIONS OF THE FOUR OPTIMIZATION CATEGORIES IN ITS

In Section II we lay the theoretical foundation and categorization of various AI optimization techniques, to build on this foundation by demonstrating practical applications of these techniques in transportation management, we introduce section III. In this section, we essentially provide real-world applications and examples of the concepts and methods introduced in section II, making the transition from theory to practice. Therefore, this section provides a detailed summary of each reviewed research article across the four categories explored in transportation. This can be seen in Table 1 - 4. For each article, we explored the research objectives, the optimization methods employed,

TABLE 1. Summary of model based applications in ITS.

References	Motivation/Objective	Contribution/Results	Algorithm or Optimization Methods	Evaluation	Future Research Direction/Research Gaps
[83]	To improve the reliability of buses and alleviate traffic congestion in urban settings using ML-based surrogate models for Dedicated Bus Lanes allocation	Enhanced accuracy and reduced computational complexity surrogate models and Jackknife uncertainty estimator in SBO framework.	Gaussian processes, Light Gradient Boosting Model , Jackknife uncertainty estimator.	DynusT simulation tool. Large-scale network in Guiyang, China.	Lift restrictions on bus frequencies, explore dynamic lane restrictions, apply ML-based SBO to continuous transportation problems.
[84]	Addressing the computational challenges in the Toll Design Problem (TDP) by developing an efficient road pricing schemes.	Introduced SAGA and RSNS heuristics for multi objective TDP using GP within a bi-level optimization framework.	Gaussian processes, surrogate-assisted genetic algorithm (SAGA), random sampling neighborhood search (RSNS).	Sioux Falls Network, Chicago Sketch Network.	Adaptive parameter setting, adaptive surrogate management, refine TDP model formulations.
[85]	Addressed the combined problem of locating and pricing charging links, considering EV drivers' routing and recharging behaviors	Bi-level optimization for locating and pricing dynamic charging links using logit-based Stochastic User Equilibrium (SUE) model and LP sub-problem.	Modified surrogate optimization mixed integer, Edge-conditioned Convolutional Network, Deep Belief Network.	Sioux Falls network.	Maintenance economy of dynamic charging links, multi-period charging/discharging decisions.
[86]	Verifying the safety of Highly Automated Vehicles (HAVs) in interaction with human-driven vehicles (HDVs)	Surrogate model-based testing framework for HAV safety under interaction with HDVs using Radial Basis Function.	Radial Basis Function, gradient descent algorithm.	Three-vehicle following scenario using MATLAB simulation.	Surrogate models for high dimensional problems, multi dimensional performance indicators for HAV evaluation.
[87]	To address the computational challenges in modeling electric vehicle (EV) charging demand and traffic user equilibrium (TUE)	Integrating edge conditioned convolutional network with deep belief network to effectively model spatial features for optimal EV charging demand.	Edge-conditioned Convolutional Network, Deep Belief Network.	6-node, 24-node, and real-world 124-node systems in Nanjing. Mean squared error (MSE) comparison.	Discharging facilities, multi-period charging/discharging decisions, optimization in transportation and power distribution networks.
[88]	To addresses the challenge of train timetable rescheduling (TTR) in high-speed railway (HSR) networks during disruptions	Surrogate models using Random Forests and Neural Networks for multi-line train rescheduling in HSR networks.	Random Forests, Neural Networks.	iROSE simulation system. Realistic iROSE simulator data.	Knowledge transfer for optimal rescheduling schemes, scalability across different lines.
[89]	To explored the growing inter dependency between transportation and power distribution networks due to the increased use of EV	Surrogate model based algorithm with adaptive particle swarm optimization for Multi period user Equilibrium in traffic power systems.	Radial Basis Function solved using a Bi-level programming method.	Sioux Falls network.	Dynamic scheduling methods, integration with renewable energy systems.
[90]	To address the rising use of Agent-Based and Activity-Based modeling in transportation, particularly for complex applications.	Surrogate-based Bayesian Optimization for calibrating behavioral parameters in complex transportation models.	Gaussian Processes.	Tested in the city of Tallinn, Estonia, using Sim-Mobility MT software to calibrate a model with 477 parameters.	Applying the algorithm to more disaggregated data sets to reduce local discrepancies and handling of model uncertainty.
[91]	To explore the challenge of evaluating and optimizing urban transportation network capacities, necessary for effective traffic planning.	KSBO algorithm for multimodal network capacity problem (MNCP) and multimodal network design (MNDP-MNCP).	Kriging-surrogate-based optimization (KSBO).	Sioux-Falls network, Nanjing network.	Integration of additional transport modes, enhancement of KSBO algorithm.

the results obtained, the identified research gaps, or the suggested directions for future work. Table 5 encapsulates the core aspects of discussions and future trends in model-based, MPC, Generative AI, and RL in transport optimization.

After the execution of a thorough review of the literature about the above-discussed four classes of AI-based optimization techniques we have found that all applied intelligent transport systems problems can be grouped into five major classes of problems namely, Congestion and Traffic Capacity

TABLE 2. Summary of model predictive control applications in ITS.

References	Motivation/Objective	Contribution/Results	Algorithm or Optimization Methods	Evaluation	Future Research Direction/Research Gaps
[92]	Minimize hydrogen costs and account for fuel cell/battery degradation in hybrid electric buses using MPC framework.	Novel strategy evaluated under prediction horizons and uncertainties; explores different driving and pricing scenarios.	MPC, simulation-based data, quasi-static modeling.	Predicting hydrogen, fuel cells, and battery price evolution; integrating factors into energy management framework.	Develop advanced models for predicting hydrogen, fuel cells, and battery price evolution.
[93]	Reduce harm in unavoidable accidents for autonomous vehicles using MPC-based motion planning.	Integrates crash severity potential and artificial potential field into MPC framework; validated via simulations.	MPC, comparative path smoothing techniques.	Field testing, application in complex urban environments, emergency situations at traffic intersections.	Explore application in more complex urban environments; evaluate during emergency situations at intersections.
[94]	Enhance energy management strategies for plug-in hybrid electric vehicles (PHEVs) using Pontryagin's Minimum Principle based model predictive control (PMP-MPC) framework.	Computational advantages over DP-MPC; integrates Markov chain-based speed predictor.	Pontryagin's Minimum Principle-based Model Predictive Control (PMP-MPC)	Simulation with Chinese City Bus Driving Cycle (CCBDC)	Extend model to incorporate vehicle-to-vehicle and vehicle-to-infrastructure communications.
[95]	Coordinate EVs, heat pumps, and thermal energy storage systems for energy community flexibility using MPC.	Two stage energy management strategy; MPC for home energy management systems (HEMS) and predictive coordination for grid flexibility community energy management.	MPC, nonlinear AC optimal power flow, simulation with actual weather data and electricity pricing in Sweden.	Four energy flexibility indicators: cost-saving, flexibility factor, self-consumption, and shedding flexibility.	The need for more precise thermal building models and deeper analysis of household costs for comprehensive benefit evaluation.
[96]	Improve traffic congestion at freeway bottlenecks using dynamic cycle Variable Speed Limit (VSL) strategy with predictive control.	Adjusts VSL based on predictive traffic flow parameters; optimized via an optimization algorithm.	Predictive control, cell transmission model, numerical analysis, simulation experiments.	Through sensitivity analysis via numerical analysis and simulation experiments, different control strategies were compared under various road bottleneck structures.	Consider environmental benefits and traffic safety; refine predictive control strategy for real highway scenarios.
[97]	Improve path tracking for autonomous vehicles using NN-based MPC with learned vehicle dynamics.	Data-driven approach with NN for accurate vehicle state prediction; evaluated under various road conditions.	Deep Neural Networks (DNN), MPC	Simulation-based evaluation, employing Simulink and CarSim, validates the effectiveness of the proposed controller	The paper assumes uniform obstacle motion and suggests future work should consider the randomness of obstacle movement for improved collision avoidance.
[98]	Enhance energy management in serial-parallel Plug-In Hybrid Electric Vehicle (PHEVs) using novel learning-based model predictive control (LMPC) strategy with GP modeling.	Reference tracking based approach; To enhance control effectiveness, an online learning process using Gaussian process (GP) modeling was integrated into MPC to address uncertainties during state estimation	Learning-based Model Predictive Control (LMPC), microscopic traffic flow analysis (MTFA), simulation with microscopic traffic flow analysis.	The evaluation compares the LMPC strategy with other algorithms like Long short-term memory (LSTM) networks and Support vector machines (SVM).	Improve velocity prediction accuracy using intelligent methods; integrate multi-source information.
[99]	Enhance target tracking performance in UAVs using Hammerstein model-based MPC for gimbal control.	Robust target tracking in the presence of disturbances.	Hammerstein model-based MPC, simulation in MATLAB.	Collaboration control strategies for multi-UAV scenarios, tracking moving targets in the air.	Explore collaboration control strategies for multi-UAV scenarios.

based, Traffic Scheduling based, Transport Safety based, Vehicle Dynamics Based and Traffic State Estimation & Prediction based. Fig. 6 shows the grouping of individual application directions in intelligent transport systems to these five classes of problems. Fig. 7 shows the mapping between these five classes of problems and the four AI-based optimization methods that we have considered for this review. From this figure, one can comprehend that model-based optimization is being used in congestion and traffic capacity-based problems and a few instances for transport

TABLE 3. Summary of generative AI applications in ITS.

References	Motivation/Objective	Contribution/Results	Algorithm or Optimization Methods	Evaluation	Future Research Direction/Research Gaps
[100]	Generate urban vehicle trajectories using generative adversarial imitation learning (GAIL).	Novelty in formulating trajectory learning as an imitation learning problem within a Partially Observable Markov Decision Process (POMDP) framework.	Generative Adversarial Imitation Learning (GAIL), POMDP framework.	Evaluated using AIMSUN simulation and real taxi trajectory data from Seoul, South Korea.	Incorporate additional variables such as traffic conditions and temporal factors; explore attention mechanisms and conditional GANs.
[101]	Enable high-speed trains to autonomously adapt to dynamic conditions for real-time decision-making, focusing on trajectory planning and motion control.	GAN-based data augmentation scheme for overcoming data insufficiency; hybrid learning network predicts speed trajectories.	MPC, LSTM, GAN, Time-Stamp Conditional GAN (TSCGAN).	Evaluated using RMSE and MAE for the Beijing-Shanghai high-speed railway.	Address challenges with different sampling periods; enhance optimization for real-time autonomous decision-making.
[102]	Predict taxi-passenger demand considering spatial, temporal, and external dependencies using a CGAN model.	Modified density-based spatial clustering algorithm with noise (DBSCAN) captures spatial dependencies; LSTM captures temporal patterns	Conditional Generative Adversarial Network (CGAN), LSTM, DBSCAN.	Evaluated using two-week data from Beijing and two-year data from New York City.	Incorporate pick-up and drop-off information; optimize training algorithms; explore cluster algorithms and time intervals.
[103]	Accurately estimate the expected time of arrival (ETA) in public transit systems, especially for bus routes.	Proposed model uses pixel-CNN and mask-CNN to learn ETA probability distributions; updates ETA in real-time.	Deep learning techniques (pixel-CNN, mask-CNN), generative modeling.	Compared with Autoregressive Integrated Moving Average (ARIMA) and LSTM using real-world data from Delhi, India.	Explore model capabilities in handling unexpected events and assessing prediction uncertainties.
[104]	Overcome imbalance in crash vs. non-crash traffic data using DCGAN for real-time crash prediction.	Introduces DCGAN for balancing dataset; integrates CNN models to capture variable correlations.	Deep Convolutional Generative Adversarial Network (DCGAN), CNN.	Evaluated on expressway SR 408 in Orlando using sensitivity, specificity, and Area Under the Curve (AUC) metrics.	Collect more extensive data; integrate data balancing and model estimation procedures to enhance prediction performance
[105]	Address traffic congestion with accurate traffic flow estimation and congestion identification.	Introduces GAN-based augmentation for traffic congestion dataset; compares CNN with ResNet-50 and DenseNet-121.	Generative Adversarial Network (GAN), CNN.	Evaluated using real traffic data from Karachi; metrics include Accuracy, Sensitivity/Recall, Prec*ision, and F1 Score.	Enhance dataset generation and model performance evaluation methodologies.
[106]	Develop a self-powered smart transportation infrastructure for traffic classification.	Proposes self-powered smart transportation infrastructure skin (SSTIS) with by Triboelectric Nanogenerator (TENG) sensors and AI analysis	TENG, Wasserstein GAN - Gradient Penalty (WGAN-GP) for data augmentation, ResNet architecture.	Evaluated with AUC and ROC metrics.	Bluetooth with commercial 5G wireless connection for improved efficiency; explore further applications in smart city infrastructures.

safety-based problems. MPC can be seen majorly beings used in transport safety and vehicle dynamics based problems. Generative AI based methods are mostly used in vehicle dynamics and traffic state prediction based problems. Lastly, Traffic scheduling and vehicle dynamics based problems are among the major class of problems that uses RL based solution.

IV. SCALABILITY AND COMPUTATIONAL TRADE-OFFS FOR AI OPTIMIZATION METHODS

There are a number of particular difficulties with using AI optimization techniques–MBO, RL, MPC, and Generative AI–to bigger transportation networks. To enable real-time, network-wide application, each method must handle the computational trade-offs between responsiveness, accuracy, and efficiency. The application of MBO to large-scale networks presents substantial problems since realistic surrogate models that simulate complicated scenarios without imposing an undue processing strain are required. Building accurate surrogate models is more costly and time-consuming as networks get bigger, particularly in high-dimensional environments with complex variable dependencies. Single-fidelity or even multi-fidelity surrogate models may be used in traditional MBO techniques to strike a compromise between model accuracy and computational expense. When scaled, these surrogates necessitate significant tweaking and frequent re-calibration, which raises processing requirements and raises the possibility of decreased model fidelity.

TABLE 4. Summary of reinforcement learning applications in ITS.

References	Motivation/Objective	Contribution/Results	Base Algorithm and Methods	Evaluation	Future Research Direction/Research Gaps
[107]	Minimize probability of delay occurrence in transportation	Improved accuracy and computational efficiency in urban traffic scenarios	Q-learning approach with dynamic neural networks	Empirical evaluations in Beijing, Munich, and Singapore traffic scenarios	Explore deep Q-network, consider weather conditions, traffic light dynamics
[108]	Optimize passenger and goods transportation	Enhanced service levels and reduced operational costs in urban environments	FlexPool: Distributed model-free deep reinforcement learning algorithm	Simulations using NYC taxi-cab operations and Google Maps data	Develop cost-efficient transit methods, multi-hop transfers, multi-agent frameworks
[109]	Flexible job shop scheduling with crane transportation	Efficient optimization of makespan and energy consumption	Knowledge-based deep Q-network model for multi-objective FJSP	Simulations in urban transportation contexts	Incorporate uncertain factors, dynamic scheduling, combination of RL and Evolutionary Algorithms (EA) approaches
[110]	Traffic light scheduling in smart transportation	Improved traffic flow in urban networks	Software-Defined Deep Reinforcement Learning (SDDRL) framework	Simulations on Indian city road network using SUMO	Extend to emergency services, Social Internet of Vehicles (SIoVs)
[111]	Real-time vehicle routing for last-mile delivery	Faster online route generation in transportation networks	Deep reinforcement learning-based neural combinatorial optimization	Simulations on Cologne transportation network	Explore advanced neural network and RL techniques, large-scale vehicle routing
[112]	Solve multi-vehicle routing problem with soft time windows	Efficient routing solutions for urban logistics distribution systems	Multi-Agent Attention Model (MAAM)	Simulations showing MAAM effectiveness	Extend to larger-scale problems, heterogeneous vehicle fleets, stochastic traffic conditions
[113]	EV charging navigation in smart grid and ITS integration	Feasible and effective EV charging strategies without prior uncertainty data	Deep Reinforcement Learning (DRL) and deterministic shortest charging route model (DSCRM)	Case studies demonstrating feasibility	Extend to multi-agent DRL, coordination of multiple EVs, data cleaning and mining for scalability
[114]	Adaptive traffic signal control in transportation networks	Joint control of traffic signals at network level	Deep Q-network (DQN) based on traffic images and CNN	Simulations showing out performance over various strategies	Measure delay with sensory data, test algorithm in real-time environments
[115]	Safe decision-making in autonomous driving	Eliminate need for labelled driving data	Hierarchical reinforcement learning (H-RL)	Simulations validating effectiveness	Validate in complex traffic scenarios like highways, intersections, urban environments

TABLE 5. Discussions and future trends in AI transport optimization categories in ITS.

Category	Common Methods Used	Common Approach Followed	Research Gaps
Model-Based	Surrogate models, Bayesian Optimization, Ant Colony Optimization, Particle Swarm Optimization, Bi-level Optimization	Multi-level modeling (bi-level, tri-level), data-driven approaches with historical and simulation data, integration of transport modes	Model generalizability, incorporation of realistic factors, data limitations (incomplete, imbalanced), efficiency and scalability challenges
Model Predictive Control	MPC, Neural Networks, Gaussian Processes, Pontryagin's Minimum Principle (PMP), Hammerstein models	Tailored MPC strategies for non-linear dynamics, uncertainties, and disturbances, extensive simulations for validation	Uncertainties (parametric, non-parametric), sophisticated model development, computational efficiency, collaborative control strategies
Generative AI	Generative Adversarial Networks (GANs), Deep Learning (CNNs, RNNs, LSTM), synthetic data generation, data augmentation techniques	Data-driven optimization with large datasets, generative AI methods for enhanced dataset quality, performance optimization	Comprehensive datasets, bridge between simulation and real-time application, model interpretability, external factors integration
Reinforcement Learning	RL, GANs for data augmentation, CNNs for traffic analysis and control, simulation-based validation	Data augmentation for improved model performance, CNNs for robust classification, simulation testing in traffic scenarios	Scalability challenges, model generalization to diverse conditions, real-time implementation gaps

MBO frequently uses ensemble surrogate models to address these issues, combining several surrogate algorithms to manage the complexity and variety of a wider network. Accuracy and computing efficiency are traded off in this integration, with more accuracy frequently resulting in the need for more computer power. This can be lessened by adaptive techniques like ensemble learning and parameter tweaking, which give computationally effective surrogates

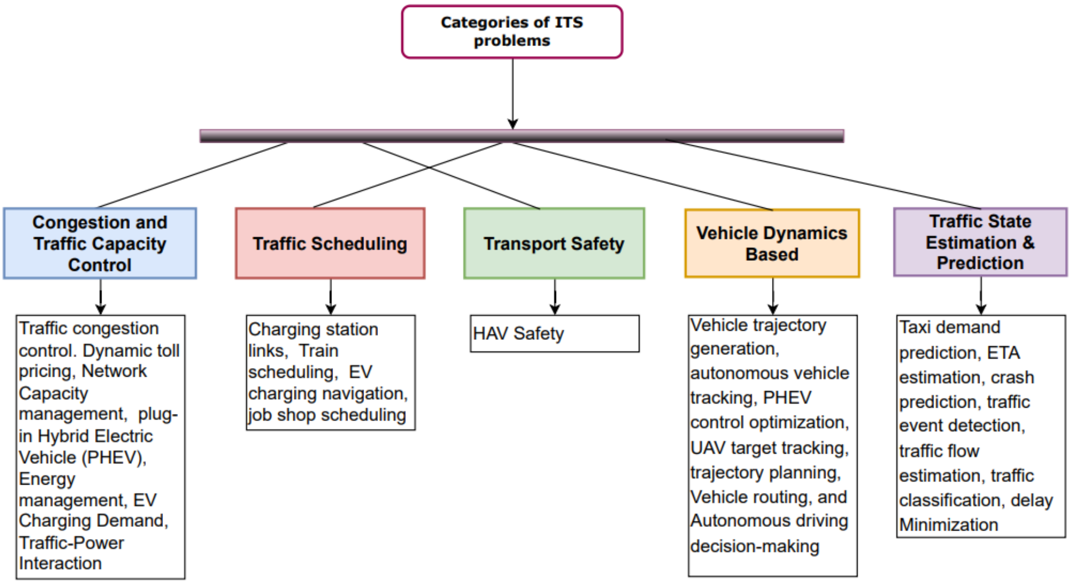


FIGURE 6. Grouping of common ITS problems.

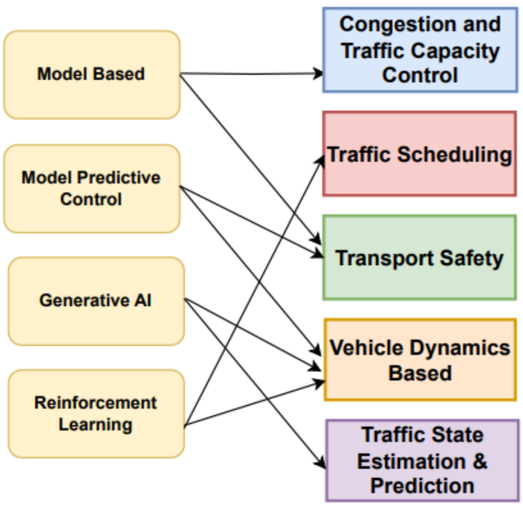


FIGURE 7. Mapping between common ITS problem groups and applied optimization solution class.

priority in real-time processes while maintaining accuracy at crucial decision points. Nevertheless, this introduces additional levels of complexity, necessitating effective model selection to provide reliable scalability.

Because of the expanded state-action space and the requirement for thorough exploration to identify the best policies, extending RL to bigger transportation networks presents difficulties. The computational requirements for training agents increase exponentially with the size of the network, which indicates more states and possible actions. Policy convergence may be delayed by RL's reliance on trial-and-error learning techniques, especially in larger settings where agent interaction or coordination may be required (e.g., in multi-agent RL setups).

Using hierarchical or multi-agent RL frameworks is one method of addressing these computational trade-offs in RL. Large issues are divided into smaller, more manageable sub-tasks via hierarchical reinforcement learning, each of which is subject to a lower-level policy. This lessens the computing effort by enabling a high-level policy to effectively direct sub-policies. Moreover, RL algorithms can be scaled with the aid of function approximators such as neural networks in Deep RL or off-policy learning, where pre-gathered data can speed up training. Nevertheless, these methods may have drawbacks: subtasks may not always match real-world situations exactly, and neural networks may be hard to understand, which makes it hard to guarantee optimal policies in big networks.

In order to respond swiftly to shifting network conditions, MPC, which is frequently employed in transportation systems for real-time decision-making, needs dynamic optimization over brief time horizons. As the size of the optimization problem increases, scaling MPC to bigger networks presents difficulties that make it harder to satisfy real-time requirements. For instance, the computational complexity increases with the number of controlled vehicles or junctions, making it more difficult for MPC to analyze in real time. High-fidelity updates are also less feasible due to the average model-update frequency in bigger networks, which might put a burden on system resources.

MPC frequently breaks down bigger networks into smaller, independently controllable segments or uses approximation techniques like reduced-order models in larger applications. These methods introduce trade-offs between local precision and global network performance, but they can control computational costs. By coordinating controllers across smaller subsystems, a distributed MPC arrangement can balance computing efficiency; nevertheless, this can be difficult to implement and necessitates cooperation to avoid suboptimal global performance.

For sample generation and surrogate model training in complex networks, generative AI techniques–specifically, Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs)–have become more and more popular. It can be difficult to scale generative models to larger transportation networks since it takes a lot of data to properly reflect the intricacies of many scenarios. For example, it takes a lot of computing power to train GANs, especially as the models and data dimensionality increase.

By providing flexible sampling and data augmentation capabilities, generative models help to mitigate computational trade-offs. This allows for the creation of synthetic data to improve simulation variety or pre-train surrogate models. But doing so comes at the expense of more processing power, particularly when dealing with high-dimensional issues like traffic forecast for a whole metropolis. Models can be trained on lower-resolution data to maximize efficiency, sacrificing precision in favor of scalability. By generalizing learned features across tasks, latent space exploration and transfer learning approaches also aid in minimizing the need for intensive retraining in every new situation. However, controlling model generalization and convergence in big networks is still a difficult part of generative AI scaling.

V. CHALLENGES IN REAL-WORLD APPLICATIONS OF AI OPTIMIZATION TECHNIQUES IN ITS

Because urban transportation settings are complex and dynamic, there are significant problems when using AI optimization models to real-world ITS. Although obtaining this degree of model accuracy is computationally demanding and frequently limited by insufficient real-time data, Model-Based Optimization necessitates highly flexible, high-fidelity surrogate models that can manage enormous, complex datasets. For real-time applications in live traffic systems, where agents must learn without creating interruptions, RL can be challenging because to its large state-action space, sluggish convergence, and sample inefficiency. Although MPC has benefits for short-term optimization, it has scalability issues in big metropolitan networks, particularly when erratic occurrences like accidents or severe weather affect real-time accuracy. Although generative AI holds promise for scenario creation and data augmentation, it requires large datasets to correctly represent the complexity of the real world and, if not trained properly, may produce biased or inaccurate data. Furthermore, rigorous synchronization is necessary when combining generative outputs with real decision-making models in ITS systems to guarantee that they reflect the state of affairs at the moment. Hybrid techniques are crucial for creating reliable and scalable ITS solutions since the effective use of each technique depends on striking a balance between computational efficiency, data reliance, and adaptability.

VI. SCALABILITY AND EFFICIENCY METRICS FOR ASSESSING THE EFFECTIVENESS OF OPTIMIZATION TECHNIQUES IN ITS

Certain criteria are necessary to evaluate how well optimization strategies in ITS function under growing network sizes and computing demands in order to gauge their scalability and efficiency. Computation time, solution correctness, and resource utilization (such as memory and CPU usage) are frequently the main determinants of efficiency. The rate of convergence to an optimal or nearly optimal solution can also be used to assess efficiency; this is especially important for methods like MBO and MPC, where making decisions quickly is crucial.

Usually, scalability is evaluated by tracking resource usage or performance deterioration as network size or problem complexity increases. For instance, in RL, decision-making throughput and latency are crucial because they show how well an algorithm can manage larger, more intricate state-action spaces without sacrificing efficiency. Metrics like synthetic data quality and realism in different traffic situations are crucial for generative AI in order to verify that model performance holds up well when the dataset expands or adapts to new circumstances. In this regard, generalizability measurements that demonstrate how well models can adjust to increasingly complicated or unknown traffic patterns also frequently reflect scalability.

VII. CONCLUSION

In this review, we have explored the role of artificial Intelligence in intelligent transport systems. The categorization of AI optimization techniques into four distinct groups, MBO, RL, MPC, and Generative AI, provides a structured framework to understand and compare the diverse methodologies employed in ITS. This review synthesizes recent advancements and identifies emerging trends within these categories, revealing the innovative approaches and technological strides made over the past five years. Furthermore, by synthesizing current research and identifying common approaches and research gaps, this paper provides a valuable road map for future advancements in AI-driven transportation optimization. This resource facilitates a deeper understanding of how different AI techniques can be applied to specific transportation problems, offering insights into the motivations behind various approaches, the solutions proposed, and their respective outcomes and research gaps.

In this article we have addressed the three research questions by categorizing AI techniques–MBO, RL, MPC, and Generative AI–and analyzing their effectiveness in optimizing ITS. For Research Question 1 on identifying the most effective techniques, we have evaluated each category's strengths, detailing how MBO excels in surrogate modeling for complex traffic environments, RL adapts through interaction-driven learning for dynamic control, MPC handles real-time short-horizon decision-making in vehicle and safety applications, and Generative AI enhances data-driven optimizations by generating realistic synthetic data. In answering Research Question 2 on efficiency and scalability, we have explored how these techniques balance computational trade-offs by measuring efficiency through metrics like computation time, resource usage, and scalability in network expansion. For instance, MBO's surrogate model accuracy, RL's sample efficiency, MPC's real-time performance, and Generative AI's data adaptability reflect each method's capacity to scale. Finally, Research Question 3 is addressed by identifying gaps and future research directions, such as improving RL's sample efficiency, enhancing MBO's model adaptability, resolving MPC's scalability constraints, and addressing Generative AI's data dependency, thus highlighting areas where further innovation is essential. This structured approach not only evaluates each technique's suitability for specific ITS applications but also establishes a roadmap for overcoming limitations, ensuring the development of robust, scalable solutions for complex transportation challenges.

REFERENCES

[1] S.-H. Chung, "Applications of smart technologies in logistics and transport: A review," *Transp. Res. E, Logistics Transp. Rev.*, vol. 153, Sep. 2021, Art. no. 102455.

[2] J. Zhang and J. Zhang, "Artificial intelligence applied on traffic planning and management for rail transport: A review and perspective," *Discrete Dyn. Nature Soc.*, vol. 2023, pp. 1–17, Jul. 2023.

[3] R. de la Torre, C. G. Corlu, J. Faulin, B. S. Onggo, and A. A. Juan, "Simulation, optimization, and machine learning in sustainable transportation systems: Models and applications," *Sustainability*, vol. 13, no. 3, p. 1551, Feb. 2021.

[4] O. S. Joel, A. T. Oyewole, O. G. Odunaiya, and O. T. Soyombo, "Leveraging artificial intelligence for enhanced supply chain optimization: A comprehensive review of current practices and future potentials," *Int. J. Manage. Entrepreneurship Res.*, vol. 6, no. 3, pp. 707–721, Mar. 2024.

[5] Z.-J.-M. Shen, B. Feng, C. Mao, and L. Ran, "Optimization models for electric vehicle service operations: A literature review," *Transp. Res. B, Methodol.*, vol. 128, pp. 462–477, Oct. 2019.

[6] A. Mourad, J. Puchinger, and C. Chu, "A survey of models and algorithms for optimizing shared mobility," *Transp. Res. B, Methodol.*, vol. 123, pp. 323–346, May 2019.

[7] R. Abduljabbar, H. Dia, S. Liyanage, and S. A. Bagloee, "Applications of artificial intelligence in transport: An overview," *Sustainability*, vol. 11, no. 1, p. 189, Jan. 2019.

[8] T. Bartz-Beielstein and M. Zaefferer, "Model-based methods for continuous and discrete global optimization," *Appl. Soft Comput.*, vol. 55, pp. 154–167, Jun. 2017, doi: 10.1016/j.asoc.2017.01.039.

[9] J. V. Soares do Amaral, J. A. B. Montevechi, R. D. C. Miranda, and W. T. D. S. Junior, "Metamodel-based simulation optimization: A systematic literature review," *Simul. Model. Pract. Theory*, vol. 114, Jan. 2022, Art. no. 102403, doi: 10.1016/j.simpat.2021.102403.

[10] J. Ceberio, A. Mendiburu, and J. A. Lozano, "A roadmap for solving optimization problems with estimation of distribution algorithms," *Natural Comput.*, vol. 23, no. 1, pp. 99–113, Sep. 2022, doi: 10.1007/s11047-022-09913-2.

[11] X. Wang, Y. Jin, S. Schmitt, and M. Olhofer, "Recent advances in Bayesian optimization," *ACM Comput. Surv.*, vol. 55, no. 13s, pp. 1–36, Jul. 2023, doi: 10.1145/3582078.

[12] T. Chatterjee, S. Chakraborty, and R. Chowdhury, "A critical review of surrogate assisted robust design optimization," *Arch. Comput. Methods Eng.*, vol. 26, no. 1, pp. 245–274, Jan. 2019, doi: 10.1007/s11831-017-9240-5.

[13] P. Jiang, Q. Zhou, and X. Shao, "Surrogate model-based engineering design and optimization," in *Springer Tracts in Mechanical Engineering*. Singapore: Springer, 2020, doi: 10.1007/978-981-15-0731-1.

[14] M. Moustapha and B. Sudret, "Surrogate-assisted reliability-based design optimization: A survey and a unified modular framework," *Struct. Multidisciplinary Optim.*, vol. 60, no. 5, pp. 2157–2176, Nov. 2019, doi: 10.1007/s00158-019-02290-y.

[15] P. S. Palar, R. P. Liem, L. R. Zuhal, and K. Shimoyama, "On the use of surrogate models in engineering design optimization and exploration: The key issues," in *Proc. Genetic Evol. Comput. Conf. Companion*. New York, NY, USA: Association for Computing Machinery, Jul. 2019, pp. 1592–1602, doi: 10.1145/3319619.3326813.

[16] H. Tong, C. Huang, L. L. Minku, and X. Yao, "Surrogate models in evolutionary single-objective optimization: A new taxonomy and experimental study," *Inf. Sci.*, vol. 562, pp. 414–437, Jul. 2021, doi: 10.1016/j.ins.2021.03.002.

[17] Z. Kus, C. Akkan, and A. Gülcü, "Novel surrogate measures based on a similarity network for neural architecture search," *IEEE Access*, vol. 11, pp. 22596–22613, 2023, doi: 10.1109/ACCESS.2023.3252887.

[18] F. Li, L. Gao, and W. Shen, "Surrogate-assisted multi-objective evolutionary optimization with Pareto front model-based local search method," *IEEE Trans. Cybern.*, vol. 54, no. 1, pp. 173–186, Jan. 2024, doi: 10.1109/TCYB.2022.3186591.

[19] M. Zhao, K. Zhang, G. Chen, X. Zhao, J. Yao, C. Yao, L. Zhang, and Y. Yang, "A classification-based surrogate-assisted multiobjective evolutionary algorithm for production optimization under geological uncertainty," *SPE J.*, vol. 25, no. 5, pp. 2450–2469, Oct. 2020. [Online]. Available: https://onepetro.org/SJ/articleabstract/25/05/2450/454132/A-Classification-Based-Surrogate-Assisted

[20] M. G. Fernández-Godino, "Review of multi-fidelity models," *Adv. Comput. Sci. Eng.*, vol. 1, no. 4, pp. 351–400, Dec. 2023, doi: 10.3934/acse.2023015.

[21] X. Shang, L. Su, H. Fang, B. Zeng, and Z. Zhang, "An efficient multi-fidelity Kriging surrogate model-based method for global sensitivity analysis," *Rel. Eng. Syst. Saf.*, vol. 229, Jan. 2023, Art. no. 108858, doi: 10.1016/j.ress.2022.108858.

[22] I. G. Farcas, B. Peherstorfer, T. Neckel, F. Jenko, and H. J. Bungartz, "Context-aware learning of hierarchies of low-fidelity models for multi-fidelity uncertainty quantification," *Comput. Methods Appl. Mech. Eng.*, vol. 406, Mar. 2023, Art. no. 115908, doi: 10.1016/j.cma.2023.115908.

[23] M. Shi, L. Lv, W. Sun, and X. Song, "A multi-fidelity surrogate model based on support vector regression," *Struct. Multidisciplinary Optim.*, vol. 61, no. 6, pp. 2363–2375, Jun. 2020, doi: 10.1007/s00158-020-02522-6.

[24] J. Tao and G. Sun, "Application of deep learning based multi-fidelity surrogate model to robust aerodynamic design optimization," *Aerosp. Sci. Technol.*, vol. 92, pp. 722–737, Sep. 2019, doi: 10.1016/j.ast.2019.07.002.

[25] X. Zhang, F. Xie, T. Ji, Z. Zhu, and Y. Zheng, "Multi-fidelity deep neural network surrogate model for aerodynamic shape optimization," *Comput. Methods Appl. Mech. Eng.*, vol. 373, Jan. 2021, Art. no. 113485, doi: 10.1016/j.cma.2020.113485.

[26] Q. Lin, X. Wu, L. Ma, J. Li, M. Gong, and C. A. C. Coello, "An ensemble surrogate-based framework for expensive multiobjective evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 26, no. 4, pp. 631–645, Aug. 2022, doi: 10.1109/TEVC.2021.3103936.

[27] M. Salehian, M. H. Sefat, and K. Muradov, "Multi-solution well placement optimization using ensemble learning of surrogate models," *J. Petroleum Sci. Eng.*, vol. 210, Mar. 2022, Art. no. 110076, doi: 10.1016/j.petrol.2021.110076.

[28] M. Yousefi and M. Yousefi, "Human resource allocation in an emergency department: A metamodel-based simulation optimization," *Kybernetes*, vol. 49, no. 3, pp. 779–796, Jun. 2019, doi: 10.1108/k-12-2018-0675.

[29] R. Shams, S. Alimohammadi, and J. Yazdi, "Optimized stacking, a new method for constructing ensemble surrogate models applied to DNAPL-contaminated aquifer remediation," *J. Contaminant Hydrol.*, vol. 243, Dec. 2021, Art. no. 103914, doi: 10.1016/j.jconhyd.2021.103914.

[30] J.-Y. Li, Z.-H. Zhan, H. Wang, and J. Zhang, "Data-driven evolutionary algorithm with perturbation-based ensemble surrogates," *IEEE Trans. Cybern.*, vol. 51, no. 8, pp. 3925–3937, Aug. 2021, doi: 10.1109/TCYB.2020.3008280.

[31] H. Kalayci, U. E. Ayten, and P. Mahouti, "Ensemble-based surrogate modeling of microwave antennas using XGBoost algorithm," *Int. J. Numer. Modelling: Electron. Netw., Devices Fields*, vol. 35, no. 2, p. e2950, Mar. 2022, doi: 10.1002/jnm.2950.

[32] A. Gosavi, "Reinforcement learning: A tutorial survey and recent advances," *INFORMS J. Comput.*, vol. 21, no. 2, pp. 178–192, May 2009, doi: 10.1287/ijoc.1080.0305.

[33] K. Khetarpal, M. Riemer, I. Rish, and D. Precup, "Towards continual reinforcement learning: A review and perspectives," *J. Artif. Intell. Res.*, vol. 75, pp. 1401–1476, Dec. 2022, doi: 10.1613/jair.1.13673.

[34] G. Shi, K. Azizzadenesheli, M. O'Connell, S.-J. Chung, and Y. Yue, "Meta-adaptive nonlinear control: Theory and algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2021, pp. 10013–10025. [Online]. Available: https://proceedings.neurips.cc/paper/paper/2021

[35] Z. Daniels, A. Raghavan, J. Hostetler, A. Rahman, I. Sur, M. Piacentino, and A. Divakaran, "Model-free generative replay for lifelong reinforcement learning: Application to Starcraft-2," in *Proc. 1st Conf. Lifelong Learn. Agents*, Nov. 2022, pp. 1120–1145. [Online]. Available: https://proceedings.mlr.press/v199/daniels22a.html

[36] J.-B. Gaya, T. Doan, L. Caccia, L. Soulier, L. Denoyer, and R. Raileanu, "Building a subspace of policies for scalable continual learning," in *Proc. 11th Int. Conf. Learn. Represent.*, Sep. 2022, pp. 1–28. [Online]. Available: https://openreview.net/forum?id=UKr0MwZM6fL

[37] A. Curtis, T. Silver, J. B. Tenenbaum, T. Lozano-Pérez, and L. Kaelbling, "Discovering state and action abstractions for generalized task and motion planning," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2022, vol. 36, no. 5, pp. 5377–5384, doi: 10.1609/aaai.v36i5.20475.

[38] K. Lu, A. Grover, P. Abbeel, and I. Mordatch, "Reset-free lifelong learning with skill-space planning," in *Proc. Int. Conf. Learn. Represent.*, Oct. 2020, pp. 1–20.

[39] L. Moro, A. Likmeta, E. Prati, and M. Restelli, "Goal-directed planning via hindsight experience replay," in *Proc. Int. Conf. Learn. Represent.*, Oct. 2021, pp. 1–16. [Online]. Available: https://openreview.net/forum?id=6NePxZwfae

[40] Z. D. Guo, B. A. Pires, B. Piot, J. B. Grill, F. Altché, R. Munos, and M. G. Azar, "Bootstrap latent-predictive representations for multitask reinforcement learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 3875–3886. [Online]. Available: https://proceedings.mlr.press/v119/guo20g.html

[41] F.-M. Luo, S. Jiang, Y. Yu, Z. Zhang, and Y.-F. Zhang, "Adapt to environment sudden changes by learning a context sensitive policy," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, Jun. 2022, pp. 7637–7646, doi: 10.1609/aaai.v36i7.20730.

[42] J. D. Co-Reyes, S. Feng, G. Berseth, J. Qui, and S. Levine, "Accelerating online reinforcement learning via model-based meta-learning," in *Proc. ICLR*, Mar. 2021, pp. 1–7. [Online]. Available: https://openreview.net/forum?id=XCJ5PEkuMkC

[43] C. A. Steinparz, T. Schmied, F. Paischer, M. C. Dinu, V. P. Patil, A. Bitto-Nemling, H. Eghbal-Zadeh, and S. Hochreiter, "Reactive exploration to cope with non-stationarity in lifelong reinforcement learning," in *Proc. 1st Conf. Lifelong Learn. Agents*, Nov. 2022, pp. 441–469. [Online]. Available: https://proceedings.mlr.press/v199/steinparz22a.html

[44] L. Canese, G. C. Cardarilli, L. Di Nunzio, R. Fazzolari, D. Giardino, M. Re, and S. Spanò, "Multi-agent reinforcement learning: A review of challenges and applications," *Appl. Sci.*, vol. 11, no. 11, p. 4948, May 2021, doi: 10.3390/app11114948.

[45] A. Oroojlooy and D. Hajinezhad, "A review of cooperative multi-agent deep reinforcement learning," *Int. J. Speech Technol.*, vol. 53, no. 11, pp. 13677–13722, Jun. 2023, doi: 10.1007/s10489-022-04105-y.

[46] E. Tampubolon, H. Ceribasic, and H. Boche, "On information asymmetry in competitive multi-agent reinforcement learning: Convergence and optimality," 2020, *arXiv:2010.10901*.

[47] T. Aotani, T. Kobayashi, and K. Sugimoto, "Bottom-up multi-agent reinforcement learning by reward shaping for cooperative-competitive tasks," *Int. J. Speech Technol.*, vol. 51, no. 7, pp. 4434–4452, Jul. 2021, doi: 10.1007/s10489-020-02034-2.

[48] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," 2020, *arXiv:2005.01643*.

[49] Y. Shen, Z. Fang, Y. Xu, Y. Cao, and J. Zhu, "A rank-based sampling framework for offline reinforcement learning," in *Proc. IEEE Int. Conf. Comput. Sci., Electron. Inf. Eng. Intell. Control Technol. (CEI)*, Sep. 2021, pp. 197–202, doi: 10.1109/CEI52496.2021.9574597.

[50] T. Yamagata, A. Khalil, and R. Santos-Rodriguez, "Q-learning decision transformer: Leveraging dynamic programming for conditional sequence modelling in offline RL," in *Proc. 40th Int. Conf. Mach. Learn.*, Jul. 2023, pp. 38989–39007. [Online]. Available: https://proceedings.mlr.press/v202/yamagata23a.html

[51] R. Kidambi, A. Rajeswaran, P. Netrapalli, and T. Joachims, "MOReL: Model-based offline reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2020, pp. 21810–21823.

[52] S. Pateria, B. Subagdja, A.-H. Tan, and C. Quek, "Hierarchical reinforcement learning: A comprehensive survey," *ACM Comput. Surv.*, vol. 54, no. 5, pp. 1–35, Jun. 2021, doi: 10.1145/3453160.

[53] D. Drexler, J. Seipp, and H. Geffner, "Learning hierarchical policies by iteratively reducing the width of sketch rules," in *Proc. KR*, vol. 19, Aug. 2023, pp. 208–218, doi: 10.24963/kr.2023/21.

[54] J. Zhang, H. Yu, and W. Xu, "Hierarchical reinforcement learning by discovering intrinsic options," in *Proc. Int. Conf. Learn. Represent.*, Oct. 2020, pp. 1–19. [Online]. Available: https://openreview.net/forum?id=rgPPHEjpmw

[55] W. Song, S. Jeon, H. Choi, K. Sohn, and D. Min, "Learning disentangled skills for hierarchical reinforcement learning through trajectory autoencoder with weak labels," *Exp. Syst. Appl.*, vol. 230, Nov. 2023, Art. no. 120625, doi: 10.1016/j.eswa.2023.120625.

[56] I. Jendoubi and F. Bouffard, "Multi-agent hierarchical reinforcement learning for energy management," *Appl. Energy*, vol. 332, Feb. 2023, Art. no. 120500, doi: 10.1016/j.apenergy.2022.120500.

[57] T. Saha, S. Saha, and P. Bhattacharyya, "Transfer learning based task-oriented dialogue policy for multiple domains using hierarchical reinforcement learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8, doi: 10.1109/IJCNN48605.2020.9206954.

[58] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annu. Rev. Control, Robot., Auton. Syst.*, vol. 3, no. 1, pp. 269–296, May 2020, doi: 10.1146/annurev-control-090419-075625.

[59] S. Lale, K. Azizzadenesheli, B. Hassibi, and A. Anandkumar, "Model learning predictive control in nonlinear dynamical systems," in *Proc. 60th IEEE Conf. Decis. Control (CDC)*, Dec. 2021, pp. 757–762, doi: 10.1109/CDC45484.2021.9683670.

[60] G. C. Calafiore and L. Fagiano, "Robust model predictive control via scenario optimization," *IEEE Trans. Autom. Control*, vol. 58, no. 1, pp. 219–224, Jan. 2013, doi: 10.1109/TAC.2012.2203054.

[61] G. A. A. Gonçalves and M. Guay, "Robust discrete-time set-based adaptive predictive control for nonlinear systems," *J. Process Control*, vol. 39, pp. 111–122, Mar. 2016, doi: 10.1016/j.jprocont.2015.12.006.

[62] E. T. Maddalena and C. N. Jones, "Learning non-parametric models with guarantees: A smooth Lipschitz regression approach," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 965–970, Jan. 2020, doi: 10.1016/j.ifacol.2020.12.1265.

[63] E. González, J. Sanchis, S. García-Nieto, and J. Salcedo, "A comparative study of stochastic model predictive controllers," *Electronics*, vol. 9, no. 12, p. 2078, Dec. 2020, doi: 10.3390/electronics9122078.

[64] L. Hewing and M. N. Zeilinger, "Scenario-based probabilistic reachable sets for recursively feasible stochastic model predictive control," *IEEE Control Syst. Lett.*, vol. 4, no. 2, pp. 450–455, Apr. 2020, doi: 10.1109/LCSYS.2019.2949194.

[65] R. Guzman, R. Oliveira, and F. Ramos, "Heteroscedastic Bayesian optimisation for stochastic model predictive control," *IEEE Robot. Autom. Lett.*, vol. 6, no. 1, pp. 56–63, Jan. 2021, doi: 10.1109/LRA.2020.3028830.

[66] D. Piga, M. Forgione, S. Formentin, and A. Bemporad, "Performance-oriented model learning for data-driven MPC design," *IEEE Control Syst. Lett.*, vol. 3, no. 3, pp. 577–582, Jul. 2019, doi: 10.1109/LCSYS.2019.2913347.

[67] M. Neumann-Brosig, A. Marco, D. Schwarzmann, and S. Trimpe, "Data-efficient autotuning with Bayesian optimization: An industrial control study," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 3, pp. 730–740, May 2020, doi: 10.1109/TCST.2018.2886159.

[68] D. Stenger, M. Ay, and D. Abel, "Robust parametrization of a model predictive controller for a CNC machining center using Bayesian optimization," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 10388–10394, Jan. 2020, doi: 10.1016/j.ifacol.2020.12.2778.

[69] U. Rosolia and F. Borrelli, "Learning how to autonomously race a car: A predictive control approach," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 6, pp. 2713–2719, Nov. 2020, doi: 10.1109/TCST.2019.2948135.

[70] N. Ab Azar, A. Shahmansoorian, and M. Davoudi, "From inverse optimal control to inverse reinforcement learning: A historical review," *Annu. Rev. Control*, vol. 50, pp. 119–138, Jan. 2020, doi: 10.1016/j.arcontrol.2020.06.001.

[71] S. Arora and P. Doshi, "A survey of inverse reinforcement learning: Challenges, methods and progress," *Artif. Intell.*, vol. 297, Aug. 2021, Art. no. 103500, doi: 10.1016/j.artint.2021.103500.

[72] M. Menner, K. Berntorp, M. N. Zeilinger, and S. Di Cairano, "Inverse learning for human-adaptive motion planning," in *Proc. IEEE 58th Conf. Decis. Control (CDC)*, Dec. 2019, pp. 809–815, doi: 10.1109/CDC40024.2019.9030020.

[73] J. H. Gillulay and C. J. Tomlin, "Guaranteed safe online learning of a bounded system," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2011, pp. 2979–2984, doi: 10.1109/IROS.2011.6095101.

[74] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, "Safe reinforcement learning via shielding," in *Proc. 32nd AAAI Conf. Artif. Intell. 13th Innov. Appl. Artif. Intell. Conf. 8th AAAI Symp. Educ. Adv. Artif. Intell.*, Feb. 2018, pp. 2669–2678.

[75] S. Shirobokov, V. Belavin, M. Kagan, A. Ustyuzhanin, and A. G. Baydin, "Black-box optimization with local generative surrogates," 2020, *arXiv:2002.04632*.

[76] A. Kumar, "Surrogate model for field optimization using beta-VAE based regression," in *Proc. IPTC*, Feb. 2022, Art. no. D031S073R002, doi: 10.2523/IPTC-22059-MS.

[77] S. Ullah, D. A. Nguyen, H. Wang, S. Menzel, B. Sendhoff, and T. Bäck, "Exploring dimensionality reduction techniques for efficient surrogate-assisted optimization," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2020, pp. 2965–2974, doi: 10.1109/SSCI47803.2020.9308465.

[78] C. Chen, Y. Zhang, J. Fu, X. Liu, and M. Coates, "Bidirectional learning for offline infinite-width model-based optimization," 2022, *arXiv:2209.07507*.

[79] A. J. Lew and M. J. Buehler, "Encoding and exploring latent design space of optimal material structures via a VAE-LSTM model," *Forces Mech.*, vol. 5, Nov. 2021, Art. no. 100054, doi: 10.1016/j.finmec.2021.100054.

[80] Y. Zhang, W. Hu, W. Yao, L. Lian, and G. G. Yen, "Offline data-driven multi-objective optimization evolutionary algorithm based on generative adversarial network," *IEEE Trans. Evol. Comput.*, vol. 28, no. 2, pp. 293–306, Apr. 2024, doi: 10.1109/TEVC.2022.3231493.

[81] C. He, S. Huang, R. Cheng, K. C. Tan, and Y. Jin, "Evolutionary multiobjective optimization driven by generative adversarial networks (GANs)," *IEEE Trans. Cybern.*, vol. 51, no. 6, pp. 3129–3142, Jun. 2021, doi: 10.1109/TCYB.2020.2985081.

[82] T. Zhang, J. Wang, and M. Q.-H. Meng, "Generative adversarial network based heuristics for sampling-based path planning," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 1, pp. 64–74, Jan. 2022, doi: 10.1109/JAS.2021.1004275.

[83] Z. Li, Y. Tian, J. Sun, X. Lu, and Y. Kan, "Simulation-based optimization of large-scale dedicated bus lanes allocation: Using efficient machine learning models as surrogates," *Transp. Res. C, Emerg. Technol.*, vol. 143, Oct. 2022, Art. no. 103827.

[84] D. Rodriguez-Roman and S. G. Ritchie, "Surrogate-based optimization for multi-objective toll design problems," *Transp. Res. A, Policy Pract.*, vol. 137, pp. 485–503, Jul. 2020.

[85] H. Liu, Y. Zou, Y. Chen, and J. Long, "Optimal locations and electricity prices for dynamic wireless charging links of electric vehicles for sustainable transportation," *Transp. Res. E, Logistics Transp. Rev.*, vol. 152, Aug. 2021, Art. no. 102187.

[86] Y. Wang, R. Yu, S. Qiu, J. Sun, and H. Farah, "Safety performance boundary identification of highly automated vehicles: A surrogate model-based gradient descent searching approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 23809–23820, Dec. 2022.

[87] Q. Yuan, Y. Ye, Y. Tang, Y. Liu, and G. Strbac, "A novel deep-learning based surrogate modeling of stochastic electric vehicle traffic user equilibrium in low-carbon electricity–transportation Nexus," *Appl. Energy*, vol. 315, Jun. 2022, Art. no. 118961.

[88] R. Liu, D. Cui, X. Dai, P. Yue, and Z. Yuan, "A data-driven surrogate modeling for train rescheduling in high-speed railway networks under wind-caused speed restrictions," *IEEE Trans. Autom. Sci. Eng.*, vol. 21, no. 2, pp. 1107–1121, Apr. 2024.

[89] Z. Zhou, Z. Liu, H. Su, and L. Zhang, "Optimal operation for cross-period scheduling of electric vehicles in traffic-power systems," *IEEE Trans. Transport. Electrific.*, vol. 10, no. 3, pp. 6676–6689, Sep. 2024.

[90] S. Agriesti, V. Kuzmanovski, J. Hollmén, C. Roncoli, and B.-H. Nahmias-Biran, "A Bayesian optimization approach for calibrating large-scale activity-based transport models," *IEEE Open J. Intell. Transp. Syst.*, vol. 4, pp. 740–754, 2023.

[91] Y. Wang, H. Liu, Y. Fan, J. Ding, and J. Long, "Large-scale multimodal transportation network models and algorithms—Part II: Network capacity and network design problem," *Transp. Res. E, Logistics Transp. Rev.*, vol. 167, Nov. 2022, Art. no. 102918.

[92] X. Hu, C. Zou, X. Tang, T. Liu, and L. Hu, "Cost-optimal energy management of hybrid electric vehicles using fuel cell/battery health-aware predictive control," *IEEE Trans. Power Electron.*, vol. 35, no. 1, pp. 382–392, Jan. 2020.

[93] H. Wang, Y. Huang, A. Khajepour, Y. Zhang, Y. Rasekhipour, and D. Cao, "Crash mitigation in motion planning for autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 9, pp. 3313–3323, Sep. 2019.

[94] S. Xie, X. Hu, Z. Xin, and J. Brighton, "Pontryagin's minimum principle based model predictive control of energy management for a plug-in hybrid electric bus," *Appl. Energy*, vol. 236, pp. 893–905, Feb. 2019.

[95] C. Srithapon and D. Månsson, "Predictive control and coordination for energy community flexibility with electric vehicles, heat pumps and thermal energy storage," *Appl. Energy*, vol. 347, Oct. 2023, Art. no. 121500.

[96] R. Lu, M. Ma, and S. Liang, "Variable speed limit strategy with dynamic control cycle based on predictive control," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2677, no. 3, pp. 620–638, Mar. 2023.

[97] M. Rokonuzzaman, N. Mohajer, S. Nahavandi, and S. Mohamed, "Model predictive control with learned vehicle dynamics for autonomous vehicle path tracking," *IEEE Access*, vol. 9, pp. 128233–128249, 2021.

[98] Y. Zhang, Y. Huang, Z. Chen, G. Li, and Y. Liu, "A novel learning-based model predictive control strategy for plug-in hybrid electric vehicle," *IEEE Trans. Transport. Electrific.*, vol. 8, no. 1, pp. 23–35, Mar. 2022.

[99] A. Altan and R. Hacioğlu, "Model predictive control of three-axis gimbal system mounted on UAV for real-time target tracking under external disturbances," *Mech. Syst. Signal Process.*, vol. 138, Apr. 2020, Art. no. 106548.

[100] S. Choi, J. Kim, and H. Yeo, "TrajGAIL: Generating urban vehicle trajectories using generative adversarial imitation learning," *Transp. Res. C, Emerg. Technol.*, vol. 128, Jul. 2021, Art. no. 103091.

[101] X. Wang, T. Xin, H. Wang, L. Zhu, and D. Cui, "A generative adversarial network based learning approach to the autonomous decision making of high-speed trains," *IEEE Trans. Veh. Technol.*, vol. 71, no. 3, pp. 2399–2412, Mar. 2022.

[102] H. Yu, X. Chen, Z. Li, G. Zhang, P. Liu, J. Yang, and Y. Yang, "Taxi-based mobility demand formulation and prediction using conditional generative adversarial network-driven learning approaches," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3888–3899, Oct. 2019.

[103] C. Paliwal and P. Biyani, "To each route its own ETA: A generative modeling framework for ETA prediction," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 3076–3081.

[104] Q. Cai, M. Abdel-Aty, J. Yuan, J. Lee, and Y. Wu, "Real-time crash prediction on expressways using deep generative models," *Transp. Res. C, Emerg. Technol.*, vol. 117, Aug. 2020, Art. no. 102697.

[105] H. Yijing, W. Wei, Y. He, W. Qihong, and X. Kaiming, "Intelligent algorithms for incident detection and management in smart transportation systems," *Comput. Electr. Eng.*, vol. 110, Sep. 2023, Art. no. 108839.

[106] U. Jilani, M. Asif, M. Rashid, A. A. Siddique, S. M. U. Talha, and M. Aamir, "Traffic congestion classification using GAN-based synthetic data augmentation and a novel 5-layer convolutional neural network model," *Electronics*, vol. 11, no. 15, p. 2290, Jul. 2022.

[107] Z. Cao, H. Guo, W. Song, K. Gao, Z. Chen, L. Zhang, and X. Zhang, "Using reinforcement learning to minimize the probability of delay occurrence in transportation," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 2424–2436, Mar. 2020.

[108] K. Manchella, A. K. Umrawal, and V. Aggarwal, "FlexPool: A distributed model-free deep reinforcement learning algorithm for joint passengers and goods transportation," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2035–2047, Apr. 2021.

[109] Y. Du, J. Li, C. Li, and P. Duan, "A reinforcement learning approach for flexible job shop scheduling problem with crane transportation and setup times," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 4, pp. 5695–5709, Apr. 2024.

[110] N. Kumar, S. Mittal, V. Garg, and N. Kumar, "Deep reinforcement learning-based traffic light scheduling framework for SDN-enabled smart transportation system," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2411–2421, Mar. 2022.

[111] J. J. Q. Yu, W. Yu, and J. Gu, "Online vehicle routing with neural combinatorial optimization and deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3806–3817, Oct. 2019.

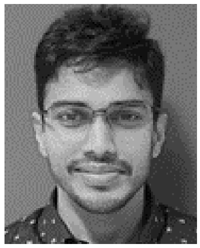
[112] K. Zhang, F. He, Z. Zhang, X. Lin, and M. Li, "Multi-vehicle routing problems with soft time windows: A multi-agent reinforcement learning approach," *Transp. Res. C, Emerg. Technol.*, vol. 121, Dec. 2020, Art. no. 102861.

[113] T. Qian, C. Shao, X. Wang, and M. Shahidehpour, "Deep reinforcement learning for EV charging navigation by coordinating smart grid and intelligent transportation system," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1714–1723, Mar. 2020.

[114] J. Lee, J. Chung, and K. Sohn, "Reinforcement learning for joint control of traffic signals in a transportation network," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 1375–1387, Feb. 2020.

[115] J. Duan, S. Eben Li, Y. Guan, Q. Sun, and B. Cheng, "Hierarchical reinforcement learning for self-driving decision-making without reliance on labelled driving data," *IET Intell. Transp. Syst.*, vol. 14, no. 5, pp. 297–305, May 2020.

BLESSING ITORO AFOLAYAN received the degree from the University of Ilorin, Nigeria, the master's degree in mathematical sciences from AIMS, Cameroon, and the African Masters in Machine Intelligence (AMMI) degree, specialized in machine learning from Ghana. She is a Researcher with a strong foundation in applied mathematics. Her academic journey was driven by a passion for solving real-world problems, exemplified by her work as an Intern in South Korea, where she designed a machine learning-based sensor clustering system for farmers. In the USA, she further honed her skills as an Intern with Carnegie Mellon University, focusing on image and text data analysis. Currently, she applies her expertise in AI with DeustoTech, Spain, where she is a Research Assistant, focusing on transportation model optimization. She received the scholarships from Google and Facebook to pursue a master's degrees.



ARKA GHOSH received the Ph.D. degree in computer science and technology from the Indian Institute of Engineering Science and Technology, Shibpur, in 2021. He is an Associate Researcher with DeustoTech, University of Deusto, Bilbao, Spain. He has authored several high-impact publications in premier journals. He has contributed to several EU Horizon projects, such as TANGENT and CONDUCTOR. His research focuses on smart mobility, multi-modal transport optimization, differential evolution, black-box optimization, and evolutionary adversarial learning. He serves as a reviewer for leading journals in his field.

JENNY FAJARDO CALDERIN received the university degree in computer engineering and the master's degree in applied informatics from the University of CUJAE (Universidad Tecnologica de la Habana), Cuba, in 2008 and 2010, respectively, and the Ph.D. degree in information and communication technologies in Granada, in 2015. Since then, she has worked in both teaching and research. She has more than ten years of teaching experience in both undergraduate and graduate lessons with CUJAE. Since 2019, she has been as a Researcher with Deusto Institute of Technology, Bilbao, Spain. In research, she has focused on the field of artificial intelligence, the main research topics are focused on intelligent systems based on soft computing techniques for decision and optimization problems. In particular, her main research interests include artificial intelligence, intelligent systems, soft computing, hybrid metaheuristics, machine learning, and intelligent transportation systems.

ANTONIO D. MASEGOSA received the university degree in computer engineering and the Ph.D. degree in computer sciences from the University of Granada, Spain, in 2005 and 2010, respectively. From June 2010 to November 2014, he was a Postdoctoral Researcher with the Research Center for ICT, University of Granada. In 2014, he received an IKERBASQUE Research Fellowship to work in the Mobility Unit, Deusto Institute of Technology, in Bilbao, Spain. In 2019, he received the IKERBASQUE Research Associate Mention and currently coordinates the Artificial Intelligence area with DeustoTech. His main research interests include artificial intelligence, intelligent systems, soft computing, hybrid metaheuristics, machine learning, deep learning, intelligent transportation systems, logistic networks, travel behaviour analysis, and traffic forecasting.