

# Hybrid machine learning approaches for optimizing vehicle routing in moroccan urban logistics

*Marouane ELABBASSI*<sup>1\*</sup>, *Karim RHOFIR*<sup>1</sup>, and *Najib Mouhassine*<sup>1</sup>

<sup>1</sup>LaSTI Laboratory, National School of Applied Sciences, Sultan Moulay Slimane University, Khouribga, Morocco

**Abstract.** City vehicle routing is actually a challenge since individuals desire various things, and numerous regulations must be observed. To improve this, we developed a method that employs artificial intelligence. This new approach is a combination of several techniques: deep neural networks, reinforcement learning, and a special layer that is used to determine the optimal paths. Decisions are also made by this layer. Regulates such problems as the time spent by a driver on the road and the vehicle size. We tested this option with real data and fictitious data of a transportation company in Morocco, which has been in operation since 2012. Our results are very similar to what occurs in real life in large Moroccan cities. It considers all the regulations, and the complex trends of the time people desire things to be delivered, and the time it takes to receive them. We contrasted our approach with other general approaches. The findings indicate that our new way results in vehicles covering a shorter distance, spending less time on computers, and saving fuel. The advantage of this system is that it pre-processes most of the work, hence it can make quick decisions in real time. In general, our new system is quite effective in locating the routes of the vehicles in the Moroccan cities.

**Keywords:** Vehicle Routing Problem; Hybrid artificial intelligence; Deep neural networks; Reinforcement learning; Split algorithm.

## 1 Introduction

The Vehicle Routing Problem (VRP) is one of the fundamental optimization problems in supply chain management and logistics. To service a group of clients at the minimal cost possible and still satisfy the operational needs such as vehicle capacity and delivery time windows, it involves computing the most efficient paths of a fleet of vehicles [4]. When appropriately considered, VRP can reduce environmental impact, enhance service quality, and reduce transportation costs drastically [5]. This is more important in fast-growing cities

---

\* Corresponding author: [marouanelabbassi@gmail.com](mailto:marouanelabbassi@gmail.com)

where traffic congestion, legal limitations, and evolving consumer demands make routing choices more challenging [5].

In many real-life situations such as the Moroccan cities of Casablanca, Marrakech, and Tangiers, logistics operations have to contend with ever-changing traffic conditions, unpredictable customer demands, and diverse infrastructure [15]. The traditional optimization techniques and formulations of VRP tend to employ a deterministic and time-independent parameter set, limiting their capabilities to capture dynamic system behavior. [4]. Due to this, offline-calculated routes might soon become suboptimal or even unfeasible when implemented in practice. longer travel times, missed delivery times, and increased fuel usage can appear due to this [5].

Promising methods for solving dynamic routing problems are provided by recent developments in artificial intelligence, especially in machine learning and reinforcement learning [10]. While routing can be described as a series of actions under uncertainty thanks to reinforcement learning (RL), supervised and deep learning models can use past data to forecast demand patterns and journey times [1]. However, when combined with learning-based components, classical heuristics and metaheuristics continue to be useful for enforcing restrictions like vehicle capacity, time frames, and depot returns in huge combinatorial spaces [12].

To offer a smart routing architecture that is tailored to the dynamic urban logistics in the Moroccan setting, this article is concerned with hybridizing these strategies. The proposed algorithm relies on a split-based algorithm to ensure compliance with vehicle capacity and customer time window constraints, a reinforcement learning agent to decide in real-time, and deep neural networks (DNN) to learn complex routing patterns [2]. The framework is integrated into a dynamic VRP formulation that specifically takes demand and time-varying travel conditions into consideration. This plan aligns with the recent trends in the data-driven and dynamic route planning of transportation networks [16].

This work makes four important contributions:

- We construct an active vehicle routing model that reflects the main points of logistics in Moroccan cities, such as the changing customer demand, the travel conditions, and the realistic operational limitations [4].
- A hybrid artificial intelligence system is proposed, which involves the use of deep neural network (DNN)-based pattern extraction, a sequence of decision-making (RL)-based reinforcement learning with classical optimization using a split-based mechanism to generate adaptive and efficient routing solutions [2].
- The performance of the proposed hybrid model is evaluated using multiple criteria, such as total travel distance, number of delayed deliveries, and fuel consumption [15].
- Also, the suggested solution is contrasted with the current solutions to analyze its performance in a variety of situations and to point out its practical aspects in terms of logistics activities [4].

This is how the rest of the paper is set up. In Section II, we look at other research that has been done on VRP, machine learning for routing, and hybrid AI-optimization methods [10]. In Section III, we talk about the dynamic VRP problem formulation. In Section 4, we go into great detail about the proposed hybrid AI method. Here is where you can find the results and setup for the experiment. The last part of the paper, Section 6, talks about what future research should look into.

## 2 Related work

### 2.1 Classical VRP and dynamic extensions

Heuristic and metaheuristic approaches are classical methods that provide near-optimal solutions for the Vehicle Routing Problem (VRP) [5]. Exact methods, such as branch-and-cut and integer linear programming, have been applied to small- and medium-sized logistics problems, where their computational performance can be effectively evaluated. These methods, however, have promising results, but they are not easily scalable to large and complex problem settings [4]. Algorithms like the nearest neighbor and Clarke-Wright methods are heuristic methods and can be used to generate solutions fast. Moreover, metaheuristic algorithms, such as Particle Swarm Optimization, Ant Colony Optimization, and Genetic Algorithms, have been commonly utilized in solving large-scale logistics issues. They are often parameter sensitive to different problem settings, which highlights their practical limitation [5].

However, despite their promising results, these methods do not scale well to large and complex problem settings [4]. Heuristic techniques, such as the nearest neighbor and Clarke-Wright algorithms, can generate solutions quickly. Moreover, metaheuristic algorithms, such as Particle Swarm Optimization, Ant Colony Optimization, and Genetic Algorithms, have been commonly utilized in solving large-scale logistics issues. However, these approaches continue to find it difficult to ensure global optimum solutions in a real-life situation with restricted computing time. Moreover, they are frequently parameter sensitive to various problem settings, which emphasizes their practical constraint [5]. Even though heuristics and metaheuristics have been adapted to dynamic conditions, they often rely on handcrafted rules, and may not take full advantage of rich data sources, such as past demand trends or current traffic data [15].

### 2.2 Machine learning for routing and logistics

Logistics and route management are becoming more and more transformed by machine learning. Two such supervised learning models are regression and feed forward neural networks, which are starting to be applied more to predict client demand, estimate actual journey times, and predict traffic congestion, providing routing algorithms with more accurate, and valuable information [10]. More expressive deep learning models that may be used to characterize the temporal dynamics of logistics flows, and the spatial layout of road networks in real-world operational environments include convolutional neural networks (CNNs), graph neural networks (GNNs), and transformer-based models [11].

Reinforcement learning (RL) methods route reinforcement learning in a very specific and efficient way, and model it as a sequence of time-dependent decisions in a Markov decision process. Here, Deep Q-Networks, policy gradient and actor-critic are used on a static and a dynamic version of the VRP [1]. These methods acquire routing policies through experience with simulated or historical environments, instead of by using handcrafted policies [2]. Long-term reward indicators inform the strategies and take into account aspects of total travel distance, quality of services, and ability to withstand disruptions [3]. Recent research has shown that deep RL can occasionally be equal or even superior to state-of-the-art metaheuristics in solving benchmark routing problems. This performance is especially remarkable when it is accompanied by attention mechanisms or the graph-based representations of states that better reflect the organization and dynamics of actual transportation systems [7].

## 2.3 Hybrid AI and optimization for VRP

Combining combinatorial optimization and data-driven learning is the goal of hybrid approaches. Neural networks are frequently used to direct the search process or produce preliminary solutions [12]. Conversely, classical heuristics guarantee the viability of the solution and further improve the paths [12]. Neural combinatorial optimization methods often use pointer networks, attention-based architectures, or graph neural networks (GNNs) to learn rules that suggest solution components [6]. To improve routing performance, these learned strategies can then be used with conventional optimization procedures [7].

Many studies in the Vehicle Routing Problem show that combining reinforcement learning with static models provides a workable solution, as well as integrating deep learning with metaheuristic approaches, which improves feasibility in routing search in this field [9]. Notable outcomes from AI-based hybrid networks are possible by using GPS sensors, IoT devices, and traffic monitoring systems in smart transportation systems [15]. These results indicate that it is possible to develop an AI-based hybrid optimization framework in VRP for use in real-world applications [13].

There are still gaps to get optimum solution in the VRP using hybrid AI frameworks in new urban logistics settings [15]. Most of the research that has been done so far looks at benchmark cases or big cities with a lot of data. There are fewer studies that talk directly about the problems and missing data that come with building logistics systems [15]. There is still a need for frameworks that can handle changing traffic and demand at the same time, use real-time signals and historical learning, and be useful for operators with limited computing power [12].

## 2.4 Positioning of the present work

The paper contributes to the literature by proposing a hybrid DNN-RL-Split model that is specially developed to solve dynamic VRP in the context of urban logistics in Morocco [2]. The suggested approach uses deep learning to learn routing patterns and state representations on past and simulated data, uses reinforcement learning to modify routing decisions in real-time based on a reward function that is consistent with the logistics goals, and a Split-based operation within a combinatorial optimization layer to impose capacity and time window constraints [2]. By testing the method on scenarios based on Moroccan cities, the research also sheds light on how hybrid AI solutions might work in new logistics markets where conditions change quickly, but data quality and infrastructure might not be as good [15].

## 3 Problem formulation

We consider a dynamic Vehicle Routing Problem (VRP) within the framework of urban logistics. A fleet of vehicles should be able to serve a group of customers within a given time with low costs and adhering to capacity and time window requirements. The system is influenced by dynamic factors such as variable travel times and the unpredictable customer requirements.

### 3.1 Sets and indices

The main sets used in the model are defined as follows:

- $V = \{0, 1, \dots, n\}$ : set of nodes, where node 0 represents the depot and nodes  $1 \dots n$  represent customers.
- $K = \{1, \dots, |K|\}$ : set of vehicles.
- $E \subseteq V \times V$ : set of directed edges.
- $T = \{1, \dots, |T|\}$ : set of discrete time periods

### 3.2 Parameters

- $c_{ij}(t)$ : travel cost from node  $i$  to node  $j$  at time  $t$
- $d_i(t)$ : demand at customer  $i$  during period  $t$
- $Q_k$ : capacity of vehicle  $k$
- $[a_i, b_i]$ : time window at customer  $i$
- $s_i$ : service time at customer  $i$
- $\tau_{ij}(t)$ : travel time from  $i$  to  $j$  when leaving at time  $t$
- $M$ : large constant for time-related constraints

### 3.3 Decision variables

- $x_{ijk}(t) \in \{0, 1\}$ : 1 if vehicle  $k$  travels from  $i$  to  $j$  at time  $t$
- $u_{ik}(t)$ : remaining load in vehicle  $k$  after serving  $i$  at time  $t$
- $y_{ik}$ : service start time of vehicle  $k$  at customer  $i$

### 3.4 Objective function

We minimize total time-dependent operational cost:

$$\min \sum_{t \in T} \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij}(t) x_{ijk}(t) \tag{1}$$

### 3.5 Constraints

1. Customer service

$$\sum_{k \in K} \sum_{t \in T} \sum_{j \in V} x_{ijk}(t) = 1, \forall i \in V \setminus \{0\}. \tag{2}$$

2. Flow conservation

$$\sum_{t \in T} \sum_{j \in V} x_{ijk}(t) = \sum_{t \in T} \sum_{j \in V} x_{jik}(t), \forall k \in K, \forall i \in V \setminus \{0\}. \quad (3)$$

3. Depot departure and return

$$\sum_{t \in T} \sum_{j \in V} x_{0jk}(t) = 1, \sum_{t \in T} \sum_{i \in V} x_{i0k}(t) = 1, \forall k \in K. \quad (4)$$

4. Capacity

$$u_{jk}(t) \leq Q_k, \forall j, k, t, \\ u_{jk}(t) \geq u_{ik}(t) - d_j(t) - M \left( 1 - \sum_{t' \in T} x_{ijk}(t') \right), \forall i, j, k. \quad (5)$$

5. Time windows and travel times

$$a_i \leq y_{ik} \leq b_i, \forall i \neq 0, \forall k, \\ y_{jk} \geq y_{ik} + s_i + \tau_{ij}(t) - M \left( 1 - \sum_{t' \in T} x_{ijk}(t') \right), \forall i, j, k. \quad (6)$$

6. Domains

$$x_{ijk}(t) \in \{0,1\}, u_{ik}(t) \geq 0, y_{ik} \geq 0. (7)$$

This formulation captures key dynamic aspects of the VRP and serves as the optimization layer for the hybrid AI framework.

## 4 Hybrid AI methodology

We create a hybrid AI framework that uses deep neural networks (DNN), reinforcement learning (RL), and a Split-based combinatorial optimization procedure to solve the dynamic VRP more quickly and flexibly.

### 4.1 Dataset description

This study used vrp dataset stored in an excel file and represents a Vehicle Routing Problem with Time Windows (VRPTW). Each row to a customer node, including the depot. The dataset has 50 customers for scenario 1 and 80 customers for scenario 2. Customer ID, Spatial coordinates ( $x_i, y_i$ ), Demand, Ready time, Due time, and Service time are all features that have numerical values. The depot is represented as a special node with zero demand and acts as the starting and ending point for all vehicles. Herein, Euclidian based method used for distance between two samples.

## 4.2 Preprocessing

Data preprocessing is a crucial step for model performance. The following procedures are followed to handle data:

- Z-score normalization is used to normalize the set of features and to provide the similarity of scales of all input variables.
- The data is divided into training, testing and validation parts in a 70:15:15 ratio respectively.
- The input data are ready according to the structural and functional needs of both deep neural networks (DNN) and reinforcement learning (RL) models.

## 4.3 Overall architecture

There are three parts to the framework:

- Deep neural network (DNN): The deep neural network is trained on underlying routing patterns, which are then fed to the component of the reinforcement learning (RL).
- The reinforcement learning module develops an optimal policy and identifies routing solutions through reward–penalty mechanisms within the dynamic environment.
- Split-based component is the part that makes sure that the solutions produced by the DNN and the RL modules do not violate real-world operational constraints, producing practical and reliable routing results.

Figure 1 shows how the proposed hybrid AI framework for dynamic vehicle routing is put together. The system uses both historical data, like past routes, demands, and travel times, and real-time data from the operational environment, like traffic conditions and new customer requests. First, a deep neural network (DNN) is trained offline to learn how to route and give initial guidance in the form of scores or priority values for the next node. An RL agent looks at the current state of the system and chooses routing actions when the system is running online. After that, these actions are sent to a Split-based optimization layer that makes routes that fit within capacity and time window limits. The fleet follows the routes that are made, and the RL policy is changed, and the data is made richer for future DNN retraining based on feedback from the environment about real travel times, delays, and service outcomes.

## 4.4 State, action and reward modelling

We model routing as an MDP:

- State,  $S_t$ : includes vehicle locations, remaining capacities, served and unserved customers, current time, and current travel time/traffic indicators
- Action,  $A_t$ : choice of the next customer (or depot) for a vehicle, or a route termination decision
- Reward,  $R_t = -(\Delta\text{distance}_t + \lambda_1 \Delta\text{time}_t + \lambda_2 \Delta\text{fuel}_t) - \phi \cdot \text{penalty}_t$

where penalties capture time window violations, capacity overloads, or unserved customers.

## 4.5 Deep neural network for route pattern learning

The proposed framework's deep neural network (DNN) is a supervised learning model that takes an encoded representation of the routing state as input. This representation includes information like customer locations, outstanding demands, current traffic conditions, and relevant time windows. The outputs are either a probability distribution for the next customer to visit or a set of priority scores for all possible customers. These can be used to make a visit sequence. When using vectorised features, the network architecture can be purely feedforward. When the road network and customer set are shown as a graph, the architecture can be graph-based. By minimising a cross entropy or regression loss with respect to reference decisions or route quality scores, the DNN learns from real or simulated routing data from the past. After training, the DNN can quickly suggest good next node options or initial visit sequences that will help with the next steps in the optimisation process.

## 4.6 Split algorithm for feasibility

Sequences generated by DNN/RL may violate constraints. The Split algorithm ensures feasibility:

- It takes an ordered list of customers and scans it to partition the sequence into segments does not exceed vehicle capacity and time window constraints remain satisfiable.
- It outputs a set of feasible routes, assigned to vehicles, that respect capacity, time windows, and depot constraints.

This step bridges learning based sequence generation and VRP feasibility.

## 4.7 Reinforcement learning for dynamic adaptation

The RL component adapts to real time changes:

- A policy  $\pi_{\theta}(a | s)$  is learned using deep reinforcement learning (e.g., Deep Q Learning, actor-critic, or policy gradients)
- At each decision step, the agent observes  $S_t$ , selects  $A_t$ , receives  $R_t$ , and transitions to  $S_{t+1}$

The policy parameters  $\theta$  are updated to maximize expected cumulative reward. Actions from the RL policy are filtered by the Split layer; constraint violations trigger penalties and guide policy improvement.

## 4.8 Algorithmic flow

The proposed architecture has two operating environments. The former is an offline setting, which combines simulated and historical information, such as travel times across multiple months in Casablanca, customer demand, and previous routing history. During the first step of the hybrid model, such data are fed to the DNN that learns customer behavioral patterns.

Then, the RL network forms a policy according to the current operational state, such as the location of the vehicles, active delivery requests, and the real-time traffic conditions, through reward- and penalty-based mechanisms. Split algorithm is then used to impose time window and capacity constraints so as to ensure feasibility of the generated solutions.

Lastly, the system constantly tracks the routing and optimizes decisions using real-time information on travel times and delays. It balances cost and service quality of operation by updating rewards and penalties, to obtain optimal routing performance.

#### 4.9 Fair comparison protocol

To ensure fairness and consistency across all methods, the following protocol is adopted:

- All methods are evaluated through travel distance, late delivery, time consumption on the same dataset to ensure fairness comparison
- We used equal number of customers, vehicle, constraint to the models for ignoring biasing of a models
- For training and evaluation of hybrid AI network, we divided 70% and 30% data respectively
- Model best parameter set during training set data when model deliver highest performance
- To gain model stability, we run multiple times through python based jupyter notebook environment
- Finally, set hyperparameter from search space where,  $DNN\_hidden = 128$ ,  $rl\_hidden = 128$ ,  $rl\_lr = 0.01$ ,  $rl\_epochs = 200$ ,  $capacity = 100$

### 5 Experimental setup and results

This study examines parcel delivery and transit in Moroccan cities using a mixed AI model. We take into account parameters like journey distance, delivery time, late deliveries, and a fuel consumption index in order to assess the model's success. We also use simulated instances. The scenarios are designed to mimic last-mile delivery processes in highly urbanised cities, both with the usage of the past data of the operator and with the synthetic cases that recreate the city structure of such cities as Casablanca or Marrakech. These involve customer locations in various districts, varying and changing demands, and a realistic delivery time on both business and residential customers. The travel times that depend on time are based on the distance and time of day congestion patterns seen in this data. The fleet is modelled with realistic capacity and shift limits. We vary the number of customers, the fleet size and the degree of demand volatility to generate several cases to observe the effectiveness of the hybrid approach in various scenarios.

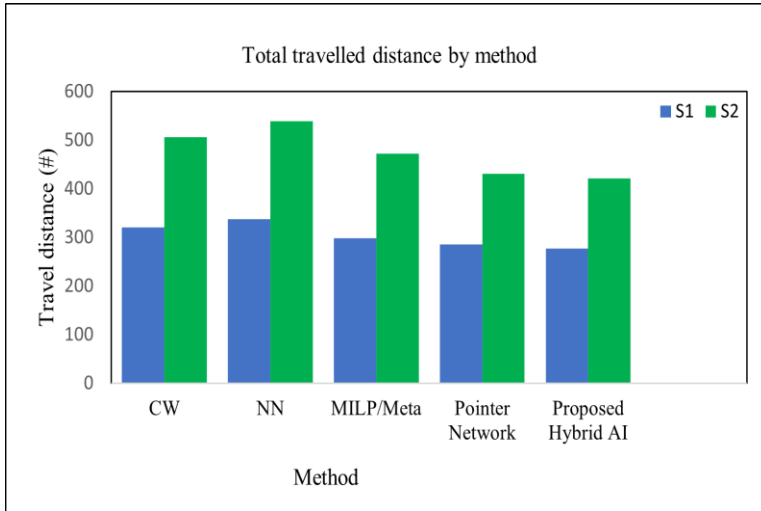
The DNN part of the implementation uses a deep learning library, the RL agent part uses a reinforcement learning library, and the Split procedure is done by hand. We tune hyperparameters such as learning rates, network depth, number of training episodes and exploration parameters on various validation scenarios. The experiments are all run on a workstation-level machine with limited resources in the form of a GPU, making sure that the computing environment is similar to what is usually available in a typical logistics company. The main performance indicators looked at are the total distance travelled, the average time it takes to calculate routes, a relative fuel consumption index based on distance and speed profiles, and service quality metrics like the number of late deliveries.

Figures 1, along with Table1, show how the proposed hybrid AI framework will affect key performance indicators in scenarios S1 and S2. Figure 1 shows the total distance travelled for each method. It shows that the hybrid AI approach is always better than the

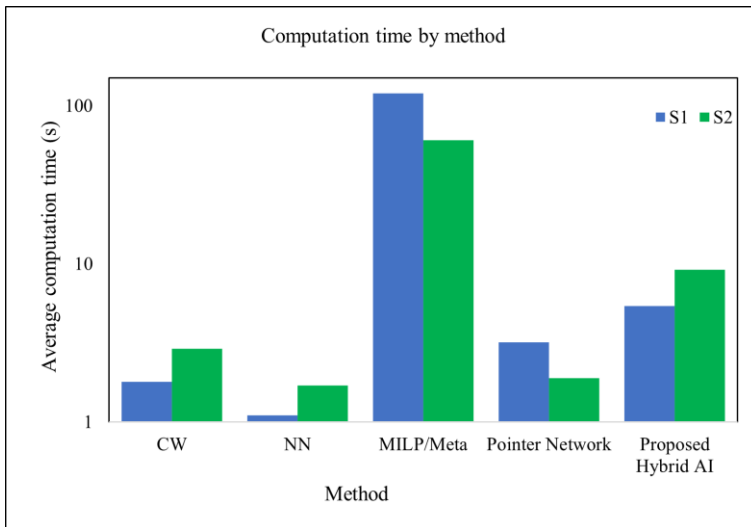
Clarke-Wright, nearest neighbour, and MILP/metaheuristic baselines. Figure 2 shows the average computation time on a logarithmic scale. It shows that the hybrid framework has moderate runtimes that are good for making decisions on the fly, while exact and metaheuristic baselines take a lot more time. Figure 3 shows how many late deliveries there were and shows that the proposed method is more reliable than classical heuristics and competitive with the MILP or metaheuristic reference. The numbers show that the hybrid AI method always does better than classical heuristics when it comes to distance travelled and fuel use, and it also cuts down on the number of late deliveries. For instance, in scenario S1, the hybrid method cuts the distance by a lot compared to Clarke–Wright and nearest neighbour.

**Table 1.** Comparative performance of the hybrid AI approach and baseline methods over dynamic VRP scenarios.

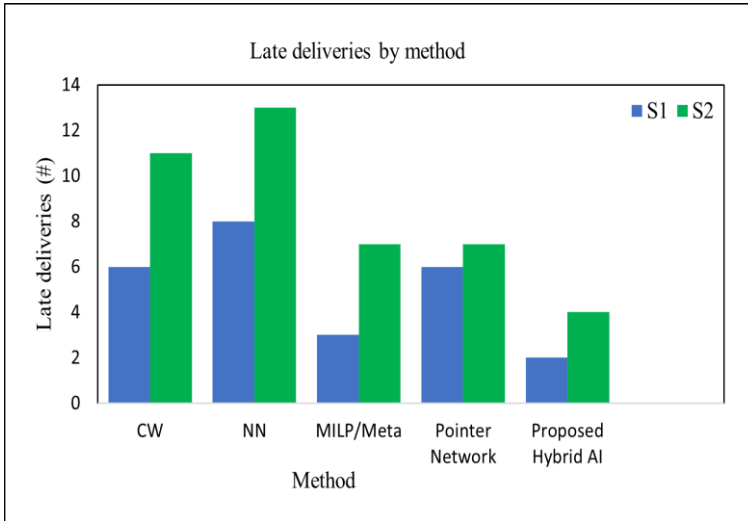
Scenario	Customers	Vehicles	Method	Total distance (km)	Avg. comp. time (s)	Fuel index (rel.)	Late deliveries (#)
S1	50	5	Clarke–Wright	320.5	1.8	1.00	6
			Nearest Neighbor	337.2	1.1	1.06	8
			MILP (small)	298.4	120.0	0.93	3
			Pointer Network	285.16	3.19	1.49	6
			Proposed Hybrid AI	276.9	5.4	0.88	2
S2	80	8	Clarke–Wright	505.7	2.9	1.00	11
			Nearest Neighbor	529.1	1.7	1.05	13
			Metaheuristic	472.3	60.5	0.94	7
			Pointer Network	430.6	1.89	1.50	7
			Proposed Hybrid AI	421.6	9.2	0.86	4



**Fig. 1.** Total travelled distance (km) obtained by Clarke–Wright, nearest-neighbor, MILP/metaheuristic and the hybrid AI method on scenarios S1 and S2. The hybrid AI approach consistently achieves the lowest distance across both instances.



**Fig. 2.** Average computation time (s) required by each method on scenarios S1 and S2. The hybrid AI framework maintains moderate computation times suitable for dynamic decision making, while exact and metaheuristic baselines remain significantly more time-consuming.



**Fig. 3.** Number of late deliveries under each method on scenarios S1 and S2. The hybrid AI solution reduces late deliveries compared to Clarke–Wright and nearest-neighbor heuristics and approaches or improves upon the performance of the MILP or metaheuristic baselines.

It also stays close to the solution found by the exact MILP baseline, but it takes a lot less time to compute once training is done. In the bigger scenario S2, the framework is still competitive with the metaheuristic baseline. This model offers an acceptable value of distance and fuel index over other heuristic models as well as balancing performance in the late delivery and computational time.

The findings show that hybrid systems have the capability of adapting to dynamic traffic conditions, and in complex and dynamic environments, the static approaches tend to perform poorly. The results also indicate that the split mechanism is effective to impose capacity and time window limits, which guarantees the viability of both DNN and RL outputs. Majority of the computational work is done in the offline phase with the online phase only being used in routing decisions based on the current traffic conditions and state features. Nevertheless, to attain strong performance in real-time and complex settings, it is not just the availability of data but also the proper tuning of the RL and DNN models. Hence, the process of finding the best routing solutions to various metropolitan environments and logistics operations is a difficult task.

## 6 Conclusion and future work

This paper suggests a new solution, which uses artificial intelligence methods to aid in route planning of delivery companies in Moroccan cities. The algorithm integrates neural networks, reinforcement learning, and a tailored optimization method to deal with dynamic routing. The proposed model makes sure that feasible decisions are made by considering time and vehicle capacity constraints, which makes the route evaluation effective.

The model demonstrates high performance with different sizes of problems when applied to real-world data of a Moroccan delivery company. It is more efficient in terms of fuel consumption, time of delivery, and distance covered when compared to the existing

methods. Nevertheless, the model is yet to be tested on a variety of datasets in different cities, and its performance can be poor in highly dynamic conditions. Consequently, it might not be applicable in some parts of Morocco. Besides, the model should be set up with a lot of caution, and this may prove to be a challenge when it comes to putting it into practice.

We are going to improve the strength of the proposed approach by using more advanced techniques in future work. We want to create a model that would be able to work across cities without retraining. We will also test the model on real-world data augmented with sensor-based and IoT-based data to capture the dynamics of traffic and the conditions of the operations better.

## References

- [1] Nazari, M., Oroojlooy, A., Snyder, L.V., Takáč, M.: Reinforcement learning for solving the vehicle routing problem. *NeurIPS DRL Workshop* (2018).
- [2] Ma, X., He, H., Wu, X., Zhou, H.: Learning to solve routing problems via deep reinforcement learning. *Transp. Res. C* 130, 103289 (2021).
- [3] Chen, X., Zhang, Y., Li, J., Wang, H.: Deep reinforcement learning approach to solve dynamic vehicle routing problem with stochastic customers and time windows. *AAAI*(2023).
- [4] Pillac, V., Gendreau, M., Guéret, C., Medaglia, A.L.: A review of dynamic vehicle routing problems. *Eur. J. Oper. Res.* 225(1), 1–11 (2013).
- [5] Ritzinger, U., Puchinger, J., Hartl, R.F.: A survey on dynamic and stochastic vehicle routing problems. *Int. J. Prod. Res.* 54(1), 215–231 (2016).
- [6] Kool, W., van Hoof, H., Welling, M.: Attention, learn to solve routing problems! *ICLR*(2019).
- [7] Bello, I., Pham, H., Le, Q.V., Norouzi, M., Bengio, S.: Neural combinatorial optimization with reinforcement learning. *ICLR Workshop* (2017).
- [8] Li, J., Sun, Y., Zhao, X., Wang, Y.: Solve routing problems with a residual edge-graph attention neural network. *Expert Syst. Appl.* 207, 117950 (2022).
- [9] Wang, L., Zhou, S., Chen, J., Zhang, R.: Enhancing vehicle routing problem solution through deep reinforcement learning and graph neural networks. *Int. J. Eng. Intell. Syst.* 30(3), 145–158 (2022).
- [10] Bengio, Y., Lodi, A., Prouvost, A.: Machine learning for combinatorial optimization: a methodological tour d’horizon. *Eur. J. Oper. Res.* 290(2), 405–421 (2021).
- [11] Veličković, P., Bengio, Y.: Recent advances in deep learning for routing problems. *ICLR Blog Track* (2022).

- 
- [12] Ahmed, A., Gupta, R., Singh, P.: A review of hybrid machine learning and metaheuristics for vehicle routing problems. *Metaheuristic Optimization Review* 2(2), 48–58 (2024).
- [13] Kendall, G., Khalil, E., Di Maio, M.: Analytics and machine learning in vehicle routing research.
- [14] Shen, Z.M., Hu, H., Li, Y., Huang, K.: Data-driven approaches for last-mile routing in urban logistics: A review. *Transp. Res. E* 162, 102709 (2022).
- [15] Sun, Y., Yuan, X., Li, Q., Wang, F.Y.: AI-enabled dynamic route planning in transportation systems: A survey. *IEEE Trans. Intell. Transp. Syst.* 25(3), 3456–3475 (2024).