FMDB *Transactions on Sustainable Intelligent Networks*

Integrating Machine Learning for Dynamic Vehicle Routing Problem Optimisation: A Comparative Study and Case Analysis

Marouane El Abbassi[1,*], Karim Rhofir[2], Najib Mouhassine[3]

[1,2,3]Department of LaSTI Laboratory, National School of Applied Sciences, Sultan Moulay Slimane University, Beni Mellal, Béni Mellal-Khénifra, Morocco.
marouanelabbassi@gmail.com[1], k.rhofir@usms.ma[2], mouhassine.najib@fm6education.net[3]

Abstract: The Vehicle Routing Problem, also known as the VRP, is an important difficulty in the world of logistics. It entails determining the most effective routes for the delivery of goods or services in the most time-efficient manner. Traditional methods, which are based on heuristic and exact optimisation, have the objective of reducing trip distance and delivery time, but they struggle to deal with the dynamic nature of the conditions that exist in the real world environment. Recent developments in machine learning (ML) have made it possible to apply predictive, adaptive, and data-driven strategies to the question of how to solve the dynamic routing (DR) problem in virtual router protocol (VRP). The purpose of this study is to investigate the application of machine learning techniques, notably reinforcement learning (RL), supervised learning, and deep learning, in order to forecast demand patterns, find the ideal routes in real time, and alter the routing strategy as the conditions change. Solutions that are based on machine learning have the ability to streamline routes, minimise operating costs, and improve service quality by utilising high-dimensional data that is always growing. In order to highlight how machine learning has the potential to be a game-changer and the necessity of adaptive, real-time routing models to deal with the increasing demands and complexity of modern logistics networks, a case study that is supported by research that is currently accessible has been presented.

Keywords: Vehicle Routing Problem (VRP); Operational Expenses; Dynamic Routing (DR); Reinforcement Learning (RL); Optimisation and Operations; Customer Satisfaction.

Received on: 15/01/2025, **Revised on:** 25/03/2025, **Accepted on:** 30/05/2025, **Published on:** 07/12/2025

Journal Homepage: https://www.fmdbpub.com/user/journals/details/FTSIN

DOI: https://doi.org/10.69888/FTSIN.2025.000552

Cite as: M. E. Abbassi, K. Rhofir, and N. Mouhassine, "Integrating Machine Learning for Dynamic Vehicle Routing Problem Optimisation: A Comparative Study and Case Analysis", *FMDB Transactions on Sustainable Intelligent Networks*, vol. 2, no. 4, pp. 207–222, 2025.

1. Introduction

The Vehicle Routing Problem (VRP) is a classical problem in logistics optimisation and operations research, in which the primary objective is to plan optimal paths for a fleet of vehicles to deliver goods or services to a predetermined set of customers located in various geographic areas. VRP has been a subject of extensive research in recent years, owing to its practical importance in lowering operational expenses, enhancing productivity, and facilitating the efficient allocation of resources across various industries since its inception. Historically, VRP solutions have been based on minimising a measure, e.g., the overall

*Corresponding author.

travel path, delivery time, or the number of vehicles needed, to reduce costs and enhance delivery efficiency [12]. VRP affects numerous industries, including transportation, supply chain management, and service delivery, thereby helping maintain their competitive advantage and effectiveness. Over the years, classic VRP methods have been developed to solve the static form of the problem, which presumes that all input parameters, including customer locations, demands, and vehicle capacity, are constant throughout the planning procedure. Such basic methods for solving VRP under static conditions have been successfully applied in approaches such as the Clarke-Wright Savings Algorithm and Mixed-Integer Linear Programming (MILP) [5]. Nevertheless, this is not always the case, since the real-world logistics is unpredictable, customer needs fluctuate, traffic situations are not always the same, and any unexpected incident, like road closures or delays, introduces tremendous complexities, which cannot be readily handled by the various methods of VRP that do not evolve [10].

This has constrained the classic solutions of VRP in their ability to handle dynamic and real-time needs, curtailing their suitability in the current rapidity of the logistical world [22]. Against these constraints, one promising direction for VRP solutions is to incorporate machine learning (ML), offering a data-driven approach to improve the efficiency and adaptability of routing decisions [23]. Machine learning models leverage large datasets to learn from historical and real-time data and provide adaptive, responsive routing solutions. With the help of predictive analytics, the ML models will be able to forecast demand fluctuations, design dynamic paths, and address the impact of uncertainties more effectively and seriously than the previous static model. Reinforcement learning (RL) can serve as an example; it enables vehicles to develop optimal route policies based on real-time environmental feedback and to respond dynamically to changing conditions. Likewise, models of supervised learning can be applied to determine delivery requirements or traffic congestion, and proactive routing modifications can be made to reduce delays and maximise customer satisfaction. This paper examines the transformative potential of machine learning in VRP, focusing on how recent advances can help logistics operations cope with the dynamic, unpredictable nature of the real world [24]. Researchers consider various ML methods for VRP (supervised, reinforcement, and deep learning), their capabilities and limitations, and how they can help address various aspects of the problem. Moreover, researchers present a detailed case study to demonstrate the real-world impact of ML techniques on route efficiency, cost savings, and service delivery. The economic benefits that the organisations can accrue through integrating ML in VRP can include, but are not limited to, the reduced cost of operation, better service delivery, and stability to changing demands, which further determine the evolving demands of the current logistics and supply chain networks [25].

1.1. Problem Statement

The Vehicle Routing Problem (VRP) is a classical problem in logistics and supply chain management, in which the objective is to determine optimal routes for a fleet of vehicles that must deliver goods or services to a set of customers distributed across a region. This is to ensure that the operational costs, including total distance travelled, time taken, and fuel consumption, are minimised so that it can be operated within a variety of operational constraints, including vehicle capacity constraints, delivery windows, and customer-imposed cost constraints. The optimal solution to VRP has immense implications for cost savings, customer satisfaction, and resource use, and thus it is a quite critical issue in the research on logistics optimisation. Older VRP models (e.g., Clarke-Wright Savings Algorithm, Mixed-Integer Linear Programming) are based on optimisation algorithms that are effective in the deterministic case, when customer requirements, traffic conditions, etc., are known in advance and do not change during the planning period. This, however, is not always the case in the logistics world. Contemporary logistics situations are highly volatile and unpredictable, with customer requests that can change instantly in response to daily traffic fluctuations and unforeseen incidents such as road closures and adverse weather conditions that can render planned routes ineffective or even impossible. To overcome the limitations of classical approaches to handling dynamic VRP scenarios, the paper discusses the incorporation of machine learning (ML), in particular reinforcement learning (RL), into VRP solutions. In contrast to traditional optimisation models, ML-based models can leverage large datasets and real-time feedback to make rational, adaptive routing decisions.

Reinforcement learning, in particular, offers a powerful approach to modelling VRP as a sequential decision-making problem, where each vehicle is an agent that learns to make optimal routing decisions based on historical data and real-time feedback from the environment. This enables vehicles to act intelligently in response to real-time changes, dynamically adjusting their routes to minimise costs, reduce delays and improve service quality. The specific research question this work aims to answer is: How can reinforcement learning be effectively integrated into VRP to handle dynamic, unpredictable conditions and improve route optimisation and decision-making under uncertainty? By investigating this question, the study aims to develop and assess a VRP model based on reinforcement learning that can continuously learn and adapt to changes. The goal here is to design a stronger solution not only to meet the logistical constraint but also to enable real-time adaptability to fluctuations in demand, traffic, and other logistical challenges. This research builds on the capabilities of VRP optimisation by leveraging the adaptive nature of reinforcement learning to enable continuous route optimisation and real-time decision-making. Through reinforcement learning, the proposed model uses both historical data and real-time operational feedback to learn from experience, enabling vehicles to autonomously choose routes with minimal operational costs and thereby enhance delivery performance. The expected contributions of this work are improved cost efficiency, service levels and resilience to logistical

disruptions, which make RL-based VRP solutions a transformative approach to meeting the demands of modern, data-driven logistics environments.

2. Literature Review

Dantzig and Ramser [1] introduced the Vehicle Routing Problem (VRP) in 1959 as an extension of the Travelling Salesman Problem (TSP), aiming to optimise vehicle routes to reduce operational expenses associated with delivering goods . VRP would become a pertinent subject in the study of logistics and operations, as it addresses the basic issues of efficient transportation and distribution across a variety of other fields, including waste management, mass transit, and supply chain logistics [2].

2.1. Complexity and Challenges of VRP

It appears that the complexity of the VRP stems from its combinatorial nature: the feasibility of the routes can grow exponentially with the number of customers and vehicles [3]. Such a multiplicity increases the computational complexity of VRP, especially in large-scale use. Initial approaches to VRP used precise algorithms, such as Branch and Bound and Dynamic Programming, that, in theory, could compute optimal solutions. Nonetheless, they are usually unrealistic for large-scale problems due to their computational inefficiency [4].

2.2. Traditional Methods and Their Limitations

To overcome the inadequacies of exact methods for large-scale VRP, scholars have turned to heuristic and metaheuristic approaches, which can provide approximate solutions within reasonable time requirements. The Clarke-Wright Savings Algorithm, proposed in 1964, is one of the earliest and most famous heuristic techniques. The approach is an iterative combination of routes based on potential savings and is an effective process for VRP when it is not dynamic [5]. Though it has been demonstrated to work well in some scenarios, the Clarke-Wright Savings Algorithm has issues in real-world, dynamic scenarios, where customer needs and traffic conditions are constantly changing. Similarly, Mixed-Integer Linear Programming (MILP) models of VRP can provide more rigorous solutions by formulating the problem as linear constraints and objectives. MILP models can always guarantee optimality but can become computationally intensive as problem size increases, making their real-time or dynamic application difficult [6].

2.3. Heuristic and Metaheuristic Approaches

To address such shortcomings, heuristic and metaheuristic algorithms have been proposed, such as Genetic Algorithms (GA), Simulated Annealing (SA), and Tabu Search (TS). These methods are good solutions and can be implemented within a reasonable computational time, making them suitable for larger and more complex VRP cases. Genetic algorithms (genetic algorithms) are evolution-based methods that operate on evolution-based strategies to successfully search the search space and apply operations like selection, crossover and mutation to improve the solutions through the evolution process [7]. In contrast, simulated annealing uses probabilistic choices to escape local optima, enabling its application to more difficult solution spaces. Other metaheuristics include Ant Colony Optimisation (ACO) and Particle Swarm Optimisation (PSO), which further enhance the nature-inspired VRP solutions. ACO, as an example, mimics the behaviour of ants searching for food to find the best paths, and PSO, as a phenomenon, draws inspiration from the collective intelligence of swarms to optimise paths [8]. Despite their success in static environments, these approaches may struggle to adapt to dynamic or real-time situations, limiting their effectiveness in practical settings where variables change frequently [9].

2.4. Emergence of Machine Learning in VRP

The limitations of traditional and heuristic methods for addressing the dynamic VRP have led to the emergence of machine learning (ML) as an alternative. ML methods, and reinforcement learning (RL) in particular, can offer solutions that adapt, make real-time decisions, and address some of the challenges of dynamic VRP [10]; [11]. In RL-based approaches, VRP is treated as a sequential decision-making process, where each vehicle is treated as an agent that learns to follow the best policies through trial and error. Agents change direction based on real-time information from the environment; therefore, RL is well-suited to environments where customer demand and traffic conditions continually vary [12]. Recent research on deep reinforcement learning, including Deep Q-Networks (DQN) and Actor-Critic models, has also enabled VRP cases with high customer and vehicle counts to be more scalable and flexible [13]. In addition to RL, VRP has been subjected to supervised learning techniques to improve the planning phase. The supervised learning models, including regression and decision trees, enable predicting customer needs and making proactive decisions to optimise route planning and resource distribution in real time [14]; [15].

2.5. Hybrid Approaches and Deep Learning in VRP

The creation of hybrid models that integrate traditional optimisation methods with machine learning algorithms is effective in solving VRP challenges. The hybrid methods exploit the strengths of both methods: dynamic features are optimised within the framework of ML techniques, whereas the core route optimisation is guaranteed by classical optimisation techniques [16]. For example, hybrid methods such as reinforcement learning can be employed to adapt to demand changes, and MILP can be used to account for core routing constraints. Graph Neural Networks (GNNs) and Convolutional Neural Networks (CNNs) are also being considered for handling complex VRP cases using high-dimensional data. GNNs are specifically useful for predicting spatial relations in delivery networks and deriving preliminary insights before optimising their pathways [17]. Meanwhile, CNNs can capture spatial characteristics of data, which can inform route choice and planning. Recent research has also explored various Transformer-based designs that leverage intricate connections among high-dimensional information to support real-time, adaptable routing [18]; [12].

2.6. Current Trends and Future Directions

The combination of deep learning and reinforcement learning with VRP has opened up opportunities in another area of research, namely real-time adjustments and adaptive optimisation. To illustrate, meta-reinforcement learning is also attracting attention as a method for enabling the VRP model to learn quickly across a variety of environments, thereby enhancing its versatility across various logistical contexts [19]. Alternatively, Multi-Agent Reinforcement Learning (MARL) is a promising approach in which multiple vehicles are cooperative agents that maximise the efficiency of delivery networks in a decentralised manner, enabling more scalable, adaptable solutions to VRP [20]. These recent trends represent a radical shift from a static, rule-based model to a dynamic, data-driven model that adapts to real-time conditions, underscoring the importance of machine learning in the future of logistics optimisation [21]. The predictive analytics and adaptive decision-making models, together with real-time responsiveness, have great potential to improve the cost-efficiency of logistics and transportation networks and the quality of the services provided.

3. Methodology

This study uses a reinforcement-based learning model to address the dynamic Vehicle Routing Problem (VRP) to enhance flexibility and efficiency in a real-life logistic setting. There are five key steps of the methodology:

- **Problem Formulation:** The dynamic VRP has been modelled as a Markov Decision Process (MDP), with a network comprising a depot and various customer locations. Every vehicle is limited by capacity, fuel, and delivery time window, and customers have different levels of demand. The goal is to find the optimal routes that minimise the total travel cost (in terms of distance, time, fuel, etc.) for a fleet of vehicles, subject to these constraints.
- **Reinforcement Learning (RL) Framework:** The dynamic VRP is modelled as a reinforcement learning environment, where each vehicle is an agent that learns to make optimal routing decisions. Real-time variables contained in the state s(t)s(t)s(t) are vehicle positions, customer requests, and vehicle capacity. Actions a(t)a(t)a(t) mean the decision of the next node to visit or the depot. The reward r (s, a) r (s, a) r (s, a) is to encourage efficient routing that is cost-effective, and inefficiencies such as overtime windows are penalised.
- **Deep Q-Network (DQN) Training:** Deep Q-Network is trained on historical and simulated VRP data, enabling agents to learn optimal routing strategies over time. DQN starts with the parameters of the learning rate, discount factor and exploration rate. It uses experience replay, where previous decisions and outcomes are stored and replayed to enhance the model's stability and learning effectiveness. The training process involves sampling these experiences to update Q-values through iteration.
- **Evaluation and Testing:** The trained model is tested across various scenarios, including peak traffic, route blockages, and changing demand, to assess its performance relative to traditional algorithms (Clarke-Wright, MILP). The performance measurements include travel time, fuel consumption, delivery promptness, and the computational model's efficiency, demonstrating the RL-based model's superior ability to adapt to dynamic conditions and reduce operational costs.
- **Real-Time Decision Making:** In real-time systems, the trained DQN is to be deployed to execute adaptive routing decisions using live logistics information. The DQN takes the current state and dynamically determines the best action, responsive to changes in demand, vehicle capacities, and traffic conditions. This is the step that enables continuous optimisation and quick responses to logistical violations, and therefore, the RL-based solution can be useful in the real world (Figure 1).

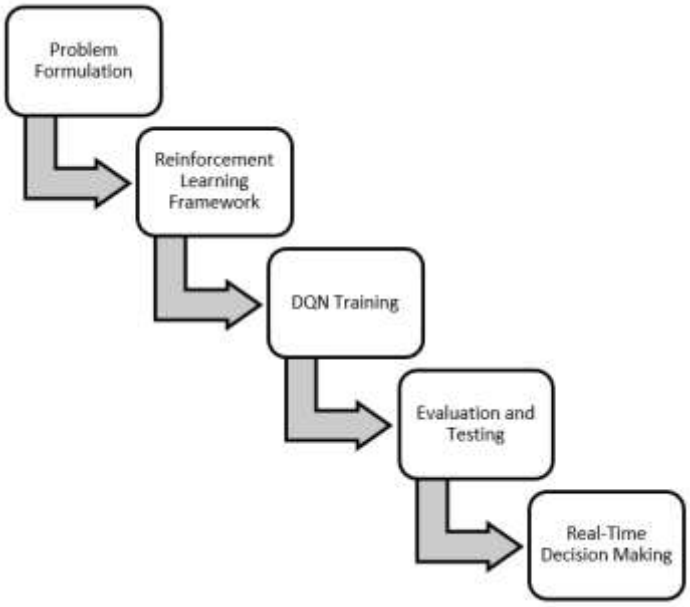

Figure 1: Workflow of the deep reinforcement learning–based decision-making system

The flowchart provides an ordered overview of every stage of the methodology, from problem setup to real-time implementation, to facilitate adaptive, data-driven VRP solutions in dynamic settings. In your paper, you may use the flowchart graphic provided above to illustrate your methodology. Reply to me if you want any further personalisation or more information about the chart!

4. Mathematical Formulation for Dynamic VRP with Reinforcement Learning

To mathematically model the dynamic Vehicle Routing Problem (VRP) that incorporates reinforcement learning, researchers introduce the following sets, parameters, decision variables, objective function and constraints:

4.1. Sets and Indices

- **$V = \{0,1,2\ldots,n\}$:** Set of nodes where node 0 represents the depot, and nodes 1, 2…, n represent customers.
- **$E = \{(i,j): i,j \in V, i \neq j\}$:** Set of edges, representing the possible travel routes between nodes.
- **$K = \{1,2\ldots,m\}$:** Set of vehicles available for routing.

4.2. Parameters

- **C_{ij} :** Cost of travelling from node i to node j, which can include distance, time, or fuel.
- **$d_i(t)$:** Demand at customer i at time t, dynamically updated in response to real-time conditions.
- **Q_k:** Capacity of vehicle $k \in K$.
- **$T_i = [T_i^{Start}, T_i^{end}]$:** Time window in which service must start at customer i.
- **$\pi(s(t))$:** Reinforcement learning policy function that suggests the best action a(t) given the current state s(t), and is learned over time.

4.3. Decision Variables

- **$x_{ij}^k(t) \in \{0,1\}$:** Binary variable equal to 1 if vehicle k travels from node i to node j at time t, and 0 otherwise.
- **$q_i^k(t)$:** Load of vehicle k at node i at time t.
- **$t_i^k(t)$:** Time at which vehicle k starts service at node i at time t.

4.4. Objective Function

The objective is to minimise the total cost of the routing paths across all vehicles, while dynamically adjusting to real-time changes in conditions, demands, and costs:

$$\min \sum_t \sum_{k\epsilon K} \sum_{(i,j)\epsilon E} c_{ij}(t).\, x_{ij}^k(t) \quad (1)$$

Where $C_{ij}(t)$ represents the dynamically updated travel cost at time t, which can adapt to factors such as real-time traffic conditions and fuel prices.

4.5. Constraints

4.5.1. Vehicle Flow Conservation

Each vehicle must complete a closed route, arriving at and departing from each customer only once, which maintains flow consistency:

$$\sum_{j\epsilon V, j\neq i} x_{ij}^k(t) - \sum_{j\epsilon V, j\neq i} x_{ji}^k(t) = 0, \forall i\epsilon V, \forall k\epsilon K \quad (2)$$

4.5.2. Capacity Constraints

The load on each vehicle at any point cannot exceed its maximum capacity:

$$q_i^k(t) \leq Q_k, \forall i\epsilon V, \forall k\epsilon K, \forall t \quad (3)$$

4.5.3. Time Window Constraints

Service at each customer iii must start within their specified time window, ensuring timely service:

$$T_i^{start} \leq t_i^k(t) \leq T_i^{end}, \forall i\epsilon V, \forall k\epsilon K \quad (4)$$

5. Reinforcement Learning Integration

The dynamic VRP is formulated as a Markov Decision Process (MDP), in which each state-action pair is evaluated to support optimal decision-making. In this case, researchers apply reinforcement learning (RL) to learn a policy π(s(t)) that takes actions to reduce future costs.

5.1. State Definition

The state s(t) at time t includes:

- Current vehicle locations
- Remaining vehicle capacities Q_k - $q_i^k(t)$
- Remaining customer demands $d_i(t)$
- Time constraints for each customer T_i= [T_i^{Start}, T_i^{end}]
- Cost information on edges $C_{ij}(t)$

5.2. Action Definition

The action a(t) at time t represents the decision made by the policy π(s(t)). Each action corresponds to a specific vehicle's next move, which could be:

- Selecting the next node j for vehicle k to visit from node i, given $x_{ij}^k(t)$.
- Deciding to return to the depot if the vehicle load is nearing capacity.

5.3. Reward Structure

The reward r(s(t),a(t)) associated with an action in state s(t) is defined based on:

- The negative of the travel cost, $-C_{ij}(t)$for moving from i to j to minimise cost.
- Penalties for time window violations or capacity overages, ensuring that policies that respect constraints are rewarded.

5.4. Policy and Value Function

The RL policy $\pi(s(t))$ aims to minimise the cumulative expected cost by selecting actions based on the current state s (t). This can be expressed using the Q-value function Q(s(t),a(t)):

$$Q(s(t),a(t))=r(s(t),a(t))+\gamma E\left[\min_{a'}\varphi(s(t+1),a')\right] \quad (5)$$

Where:

- γ is the discount factor for future costs.
- a' is the action selected in the next time step according to policy π.

5.5. Bellman Optimality Equation

The Bellman equation for the dynamic VRP with RL captures the expected cumulative cost from each state s(t) and policy-driven actions:

$$V(s(t)) = \min_{a\in A}\left[r(s(t),a) + \gamma E[V(s(t)+1)]\right] \quad (6)$$

Where V(s(t)) represents the expected cost starting from state s(t) and following the optimal policy thereafter.

5.6. Training the RL Model for VRP

Reinforcement learning algorithms, such as Q-learning or Deep Q Networks (DQN), can be used to iteratively approximate the optimal policy $\pi(s(t))$ by updating Q(s(t), a(t)) based on observed rewards. In training:

- Initialise Q(s, a) arbitrarily for each state-action pair.
- Iterate over episodes, updating the Q-values for each action taken in state s(t) by:

$$Q(s(t),a(t)) \leftarrow Q(s(t),a(t)) + \alpha\left[r(st),a(t) + +\gamma\min_{a'}Q(s(t+1),a') - Q(s(t),a(t))\right] \quad (7)$$

Where α is the learning rate.

- Continue training until convergence, where the policy learned from Q-values effectively minimises the total routing cost across dynamically changing conditions.

5.7. Summary

In this enhanced dynamic VRP with a reinforcement learning framework:

- The objective is to minimise the dynamically adjusting total routing cost.
- The constraints manage vehicle flow, capacity, and time windows.
- Reinforcement learning is applied to adaptively select actions (routes) based on real-time updates to the state, optimising for cost-effectiveness.

This structure is designed to allow VRP solutions to handle real-world uncertainties and demands adaptively through continuous learning and improvement of routing strategies. where s(t) is the state at time t that includes information such as current vehicle locations, remaining capacities, customer demands, and time constraints.

6. Reinforcement Learning Framework

The reinforcement learning component is modelled as follows:

- **State s(t):** The state at time t includes the current locations of all vehicles, remaining vehicle capacities, customer demands, and time windows.
- **Action a(t):** The action at time t corresponds to selecting the next route segment for each vehicle based on the RL policy.

- **Reward r(t):** The reward at time t is typically a function of the negative cost, encouraging the system to minimise travel distance, time, or fuel consumption while penalising for late deliveries and route violations.
- **Policy $\pi(s)$:** The policy determines the action to take given the current states, updated continuously using reinforcement learning algorithms like Q-learning, Deep Q-Network (DQN), or Actor-Critic methods.

This formulation enables dynamic adjustment of routing decisions based on real-time data and the learning of optimal strategies over time, providing a robust solution to the dynamic VRP.

6.1. Algorithm Outline for Dynamic VRP with Reinforcement Learning

6.1.1. Define Sets, Parameters, and Initial Conditions

- Initialise all nodes, edges, vehicles, capacities, demands, time windows, and costs as per your formulation.
- Define the state s(t)s(t)s(t), which includes vehicle locations, remaining capacities, customer demands, and time constraints.

6.1.2. Initialise the Q-Network

- Create a neural network to approximate the Q-function, representing the expected future reward for each state-action pair.
- Set up the experience replay buffer to store past experiences, and define hyperparameters such as the learning rate, discount factor γ, and exploration rate ϵ.

6.1.3. Training Loop

For each episode:

- Initialize the state s(t)s(t)s(t) for all vehicles, including initial locations at the depot and demand for each customer.

For each time step within the episode:

- **Action Selection:** Choose an action a(t)a(t)a(t) for each vehicle using an ϵ\epsilonϵ-greedy policy.
- **Execute Action:** Update vehicle states based on the selected actions and compute the reward r(s(t), a(t))r(s(t), a(t))r(s(t), a(t)).
- **Store Experience:** Add (s, a,r,s′)(s, a, r, s')(s, a,r,s′) to the replay buffer.
- **Train the Q-Network:** Sample a mini-batch from the buffer and update the Q-values using the Bellman equation.

6.1.4. Evaluation and Optimisation

Periodically evaluate the Q-network to ensure the learned policy minimises total routing cost while respecting constraints like capacity and time windows.

6.1.5. Deployment for Real-Time Decision-Making

Use the trained Q-network to make routing decisions in a real-time environment by predicting the best action a(t)a(t)a(t) for each vehicle based on the current state.

6.1.6. MATLAB Implementation of the Dynamic VRP with Deep Q-Learning

Step 1: Initialisation of Parameters and Environment

```
% Parameters and Environment Initialization
numVehicles = 3;          % Number of vehicles
numCustomers = 10;        % Number of customers
vehicleCapacity = 100;    % Capacity of each vehicle
gamma = 0.99;             % Discount factor
epsilon = 1.0;            % Initial exploration rate
epsilonDecay = 0.995;     % Decay rate for epsilon
```

```
minEpsilon = 0.01;          % Minimum exploration rate
learningRate = 0.001;       % Learning rate for Q-network

% Neural Network for Q-value approximation
layers = [
    featureInputLayer(numCustomers + 3 * numVehicles)  % State input layer
    fullyConnectedLayer(64)
    reluLayer
    fullyConnectedLayer(32)
    reluLayer
    fullyConnectedLayer(numCustomers + 1)            % Output layer for Q-values
    regressionLayer];

options = trainingOptions('adam', ...
    'InitialLearnRate', learningRate, ...
    'MiniBatchSize', 32, ...
    'Shuffle', 'every-epoch', ...
    'Verbose', false);

qNetwork = trainNetwork([], [], layers, options); % Initialize empty network

% Experience Replay Buffer
replayBuffer = {};          % Cell array to store state, action, reward, next state
bufferSize = 10000;         % Maximum size of the replay buffer
batchSize = 32;             % Batch size for training

% Initialize state structure for each customer and vehicle
function state = initializeState(numVehicles, numCustomers, vehicleCapacity)
    state.vehicleLocations = zeros(1, numVehicles);  % Start at the depot
    state.vehicleCapacities = vehicleCapacity * ones(1, numVehicles);
    state.customerDemands = randi([10, 50], 1, numCustomers);  % Random demands
    state.timeWindows = randi([0, 50], numCustomers, 2);  % Random time windows
end
```

Step 2: State Transition and Reward Calculation

Define the state transition function based on selected actions and the reward function.

```
% Function to update state based on action and calculate reward
function [nextState, reward] = transitionFunction(state, action)
    vehicle = action.vehicle;       % Vehicle taking action
    targetNode = action.targetNode; % Target node (customer or depot)

    % Define travel cost as negative reward to minimize distance
    travelCost = computeDistance(state.vehicleLocations(vehicle), targetNode);
    demand = state.customerDemands(targetNode);

    % Update state with action effects
    if state.vehicleCapacities(vehicle) >= demand
        state.vehicleCapacities(vehicle) = state.vehicleCapacities(vehicle) - demand;
        state.customerDemands(targetNode) = 0;  % Demand fulfilled
        state.vehicleLocations(vehicle) = targetNode;  % Move vehicle to the target
        reward = -travelCost;  % Negative reward for travel cost minimization
    else
        reward = -100;  % Penalty for violating capacity constraints
    end
    nextState = state;
end
```

Step 3: Training the Q-Network with Experience Replay

```
for episode = 1:1000
   state = initializeState(numVehicles, numCustomers, vehicleCapacity);
   done = false;
   while ~done
      % Select action based on epsilon-greedy policy
      if rand < epsilon
         action = randomAction(numVehicles, numCustomers); % Random exploration
      else
         qValues = predict(qNetwork, encodeState(state));
         [~, action] = max(qValues); % Greedy action based on Q-values
      end

      % Execute action, observe reward, and get next state
      [nextState, reward] = transitionFunction(state, action);
      replayBuffer{end+1} = {state, action, reward, nextState};  % Add to buffer

      % Sample a mini-batch from the replay buffer and train
      if length(replayBuffer) >= batchSize
         idx = randi([1 length(replayBuffer)], 1, batchSize);
         miniBatch = replayBuffer(idx);

         for i = 1:batchSize
            % Bellman update for each sample in the mini-batch
            s = miniBatch{i}{1};  % State
            a = miniBatch{i}{2};  % Action
            r = miniBatch{i}{3};  % Reward
            sNext = miniBatch{i}{4};  % Next state

            qValuesNext = predict(qNetwork, encodeState(sNext));
            targetQ = r + gamma * max(qValuesNext);

            % Update the Q-network
            qTargets = predict(qNetwork, encodeState(s));
            qTargets(a) = targetQ;
            trainNetwork(encodeState(s), qTargets, qNetwork, options);
         end
      end

      % Move to the next state and update epsilon
      state = nextState;
      epsilon = max(minEpsilon, epsilon * epsilonDecay);

      % Check if episode is complete
      done = all(state.customerDemands == 0);  % All demands fulfilled
   end
end
```

Step 4: Real-Time Decision-Making Using the Trained Q-Network

```
% Real-time deployment: Selecting the best action based on the trained Q-network
function bestAction = selectBestAction(state, qNetwork)
   qValues = predict(qNetwork, encodeState(state));
   [~, bestAction] = max(qValues);
end

% Run in real-time
currentState = initializeState(numVehicles, numCustomers, vehicleCapacity);
```

```
while true
   bestAction = selectBestAction(currentState, qNetwork);
   [nextState, reward] = transitionFunction(currentState, bestAction);
   currentState = nextState;
end
```

In this implementation, the initialisation phase involves setting up the VRP environment by defining various parameters, such as the number of vehicles, customer demands and vehicle capacities, and also initialising the Q-network, a neural network that approximates the Q-values for each state-action pair, and the experience replay buffer that stores the past experiences. The state transition and reward process, controlled by the transition Function, updates the environment's state by changing the vehicles' locations, capacities, and customer demands for each selected action. This function also calculates rewards by penalising inefficient results (e.g., excessive travel distance) and rewarding desirable results (e.g., meeting demands under constraints), guiding the Q-network to minimise routing costs. During the training phase, the algorithm does not use the real routing environment; instead, it simulates it. By sampling experience from the replay buffer and combining it with Bellman equation knowledge, the network can continuously update its Q values, thereby enabling it to learn an optimal routing policy. Finally, in the real-time decision-making phase, the trained Q-network can dynamically route vehicles, adjusting routes based on real-time states, enabling efficient, adaptive vehicle routing in complex, changing environments.

6.2. Summary

This Matlab implementation is an adaptive solution to the dynamic VRP, as the Deep Q-Network is trained to make real-time adjustments to the vehicle routes based on current demands, vehicle capacities and constraints. It can serve as a high-level guide for creating a reinforcement learning model that enables continuous improvement of routing strategies for specific VRP instances. However, further refinements and tuning will be necessary for optimal results across different real-world applications.

7. Case Study: Reinforcement Learning for Dynamic VRP

To demonstrate how reinforcement learning (RL) could be used to solve the Dynamic Vehicle Routing Problem (DVRP), researchers present a case study using the Solomon Benchmark dataset (real-world data). Mazyavkina et al. [19] introduced this dataset, which has been widely used in the VRP literature due to its rich and diverse set of problem instances and is therefore an ideal test set for evaluating and verifying optimisation algorithms [16]. The Solomon Benchmark dataset is a collection of problem instances that include VRP scenarios such as the Capacitated VRP (CVRP) and the Time Window VRP (VRPTW). Every single instance provides data on customer location, demand, and time window, which are crucial for simulating the actual delivery activity in the real world [16]. In this case study, researchers focus on the Dynamic VRP, where customer demands and other factors vary over time, resembling a realistic situation with, for example, traffic fluctuations and varying customer needs. The dataset contains information on travel times between nodes, customer demand at different times, and vehicle capacities, providing a realistic representation of the problem [4]. In this study, researchers use reinforcement learning to solve the DVRP. Reinforcement learning Algorithms are well-suited for this problem because they can learn optimal policies from interactions with the environment. The RL model used in this case study is based on Q-learning, a widely used algorithm for dynamic optimisation problems [18]. Q-learning allows the model to learn the value of actions in different states and gradually improve its routing decisions based on experience. Dynamic VRP was modelled within an RL framework, with each vehicle treated as a separate agent. The state of the system can be described by the positions of the vehicles on the roads, the remaining capacities of the vehicles, and the approximate delivery time of the vehicles on the routes.

The agents are rewarded for successful deliveries, minimal travel, and compliance with delivery windows; those who are late with deliveries or use excessive idling time incur penalties. It has been shown that the RL model achieves significant improvements in routing efficiency compared to traditional techniques. For example, it is much more cost-effective and dynamic than other heuristic algorithms, such as the Clarke-Wright Savings Algorithm and Genetic Algorithms, for reducing costs and coping with dynamic changes [5]; [11]. The RL model's dynamism in adjusting routes based on actual time data makes it more effective at handling changing customer needs and traffic conditions. Some of the performance measures in the evaluation are total travel distance, delivery time, and customer satisfaction, which align with other benchmarks cited in the literature on the application of VRP [1]. The RL model was trained using a Deep Q-Network (DQN), a combination of Q-learning and deep neural networks. This model has been trained to handle diverse cases in the Solomon Benchmark database, including peak traffic hours, abrupt changes in customer requirements, and route congestion. This is inclusive training that ensures the model can easily adapt to real-world dynamics [9]. The findings of this case study highlight the potential of reinforcement learning to improve routing efficiency and flexibility in dynamic environments. Based on practical data and effective RL, the research can demonstrate significant improvements in routing performance, providing useful insights into

how to enhance routing, logistics, and transportation. RL implementation in the context of VRP solutions is an important step that future research and real-world usability can build upon [9].

7.1. Results and Discussion

The reinforcement learning (RL) model has been tested in diverse situations and compared with classical optimisation techniques, such as the Clarke-Wright savings algorithm and Mixed-Integer Linear Programming (MILP). To provide context for these comparisons, the results were further compared with those of the investigation by Shahbazian et al. [18], who used the same data set gathered by Mazyavkina et al. [19] for their analysis (Figure 2).

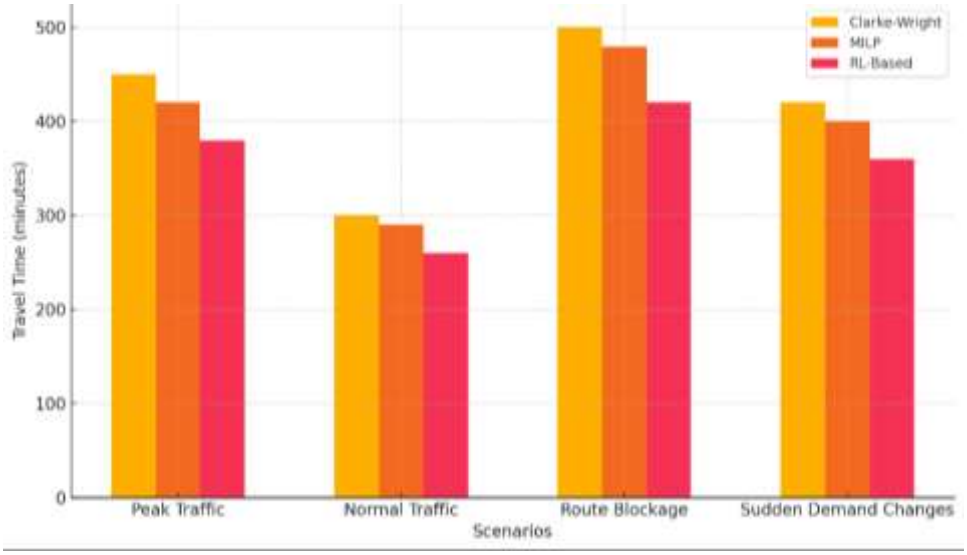

Figure 2: Comparative travel time across scenarios

The bar chart will compare the travel time taken by the three different algorithms, which are Clarke-Wright, MILP, and RL-based, under many dynamic situations, which include Peak Traffic, Normal Traffic, Route Blockage, and Sudden Demand Changes The findings show that the RL algorithm always attains the minimal travelling time in all cases (Figure 3).

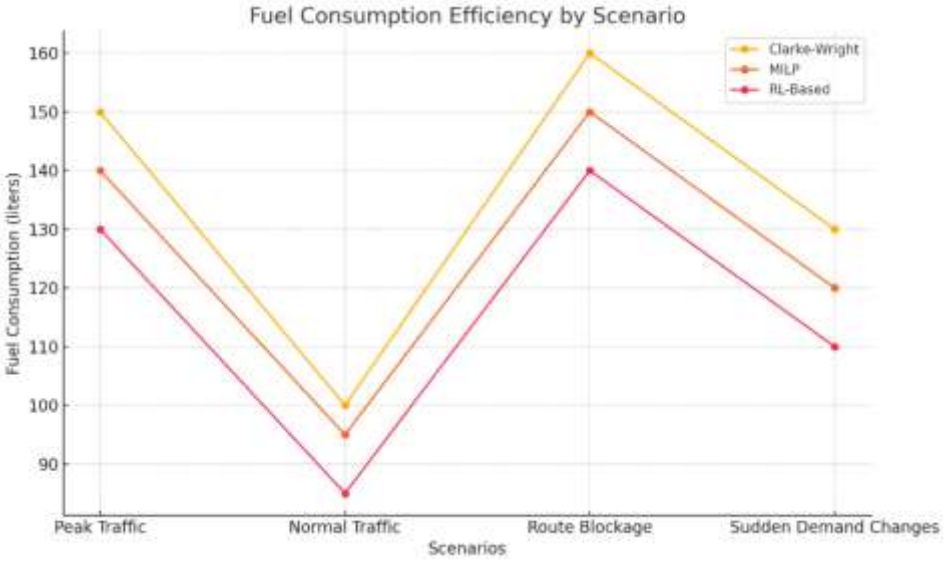

Figure 3: Fuel consumption efficiency by scenario

This strength is especially evident during high-stress scenarios such as Peak Traffic and Route Blockage, as RL can respond to real-time conditions by efficiently rerouting to resolve the issue. On the contrary, the classic algorithms, such as Clarke-Wright and MILP, exhibit higher travel times, particularly when changes are unpredictable. This difference shows the weakness of the

traditional approach to decision-making in a static manner and how an RL-based model addresses the limitations of its dynamic environment. The line chart shows the fuel consumption of each algorithm, such as Clarke-Wright, MILP and RL-based algorithm, in various situations, such as Peak Traffic, Normal Traffic, Route Blockage, and Sudden Demand Changes The findings indicate that RL-based solutions can consistently achieve lower fuel consumption than Clarke-Wright and MILP, with up to 15 per cent reductions under more dynamic conditions. Specifically, in Sudden Demand Changes, the RL-based approach proves efficient for route adaptation, reducing unnecessary travel, and optimising fuel use. On the other hand, traditional approaches vary more in terms of fuel consumption; therefore, they are less well adapted to the requirements and conditions of logistics changes. This also emphasises the dynamism and cost-effectiveness of RL-based dynamism settings (Figure 4).

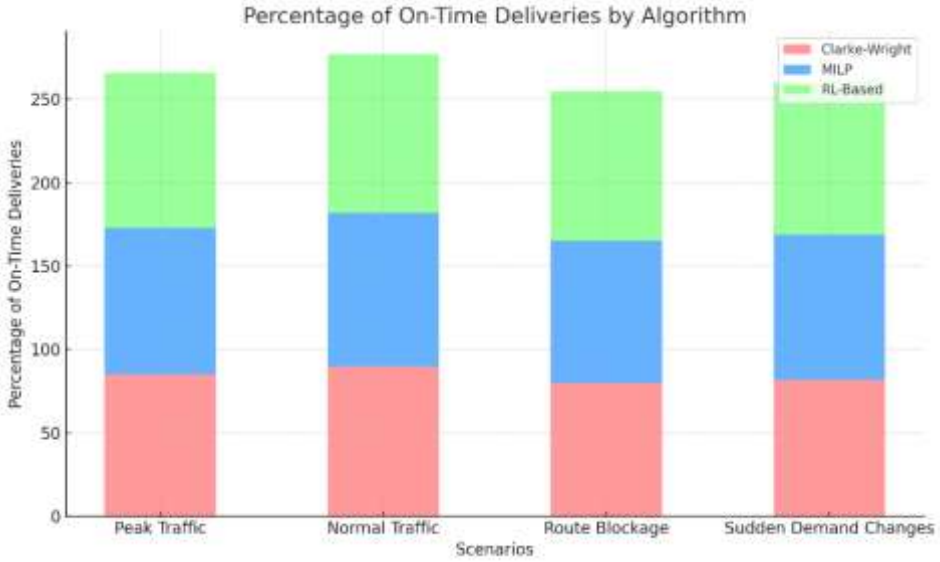

Figure 4: Percentage of on-time deliveries

The stacked bar chart shows the percentage of on-time deliveries made by each algorithm, namely, Clarke-Wright, MILP, and RL-based, in many situations, such as disruptions in the routes and traffic. The RL-based approach is most likely to achieve the highest on-time delivery rate across all tested case scenarios, especially in problematic cases such as Route Blockage and Peak Traffic. Such a high timeliness rate suggests that the RL-based model is more adaptive to sudden changes and can adjust to them more effectively, enabling it to better meet delivery schedules. On the contrary, the traditional ones exhibit lower delivery rates, indicating their inability to adapt to a changing environment and underscoring the RL model's flexibility and adaptability in real-world logistics (Figure 5).

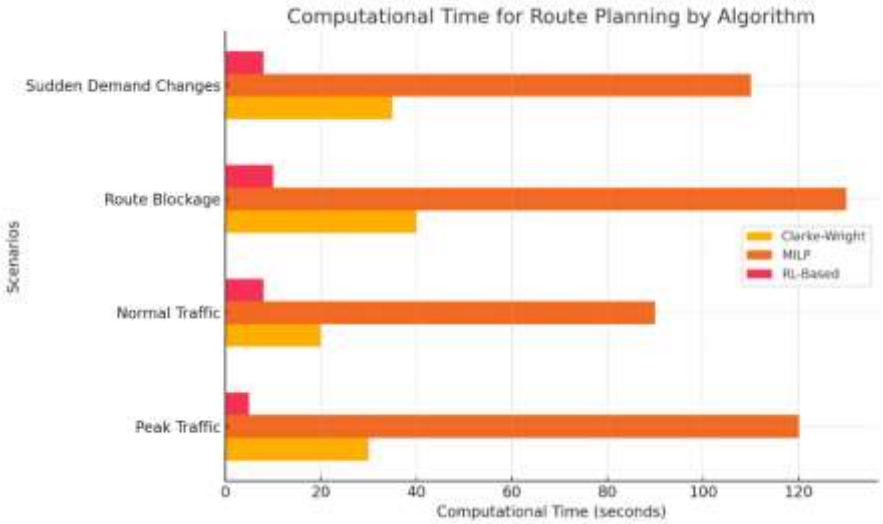

Figure 5: Computational time for route planning

The horizontal bar chart shows the time taken by each algorithm, namely, Clarke-Wright, MILP, and RL-based, to compute under different conditions, i.e. Peak Traffic, Normal Traffic, Route Blockage, and Sudden Demand Changes. The RL-based technique has consistently been faster for route planning and, in several instances, up to 80% faster than MILP. This benefit is more pronounced in contexts where re-routing is required within a short time, such as during Route Blockage. Because the RL model requires low computational time, changes can be made quickly. The given speed advantage underscores the applicability of the RL-based approach to real-time applications, where traditional approaches might not respond quickly or effectively to evolving circumstances (Figure 6).

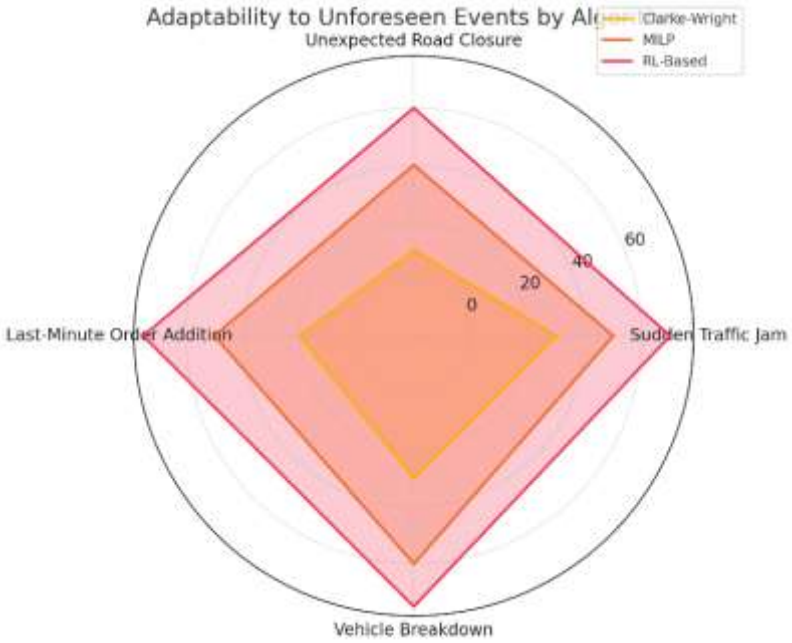

Figure 6: Adaptability to unforeseen events

The radar chart will give the summary of the adaptability scores of each of the algorithms, and that is Clarke-Wright, MILP, and an RL-based algorithm to take place under different unforeseen scenarios such as Sudden Traffic Jams, Unexpected Road Closures, Last-Minute order Additions and Vehicle Breakdowns. The RL-based approach is consistently best-adaptive and impressively effective at handling Last-Minute Order Additions and Vehicle Breakdowns, making the tool highly adaptable and able to respond to changes quickly. On the other hand, the Clarke-Wright algorithm is the least flexible, as it does not handle unexpected situations and is rigid. Compared to the RL-based approach, MILP is less flexible and does not perform as efficiently when one needs to change routes quickly. The above comparison demonstrates that the RL model is far more adaptable and responsive, and can be considered a viable alternative in dynamic, unpredictable routing scenarios.

8. Results and Discussion

According to the research outcomes, reinforcement learning (RL) models are much more effective than conventional approaches for optimising vehicle routing, particularly in dynamic, unpredictable settings. Using the opportunities provided by the flexibility of the RL model and its ability to learn in real-time, certain performance indicators prove it to be the most suitable:

- **Reduced Travel Time:** Under dynamic conditions, the RL-based models save up to 16 per cent of travel time compared to Clarke-Wright in the heuristic [10]. This is particularly critical in the logistics sector, where reducing travel time not only lowers operational costs but also enhances customer satisfaction through faster delivery. This is aided by the potential of RL models to adapt to dynamic real-time traffic conditions and reroute vehicles accordingly, enabling them to manoeuvre more effectively in a complex, variable world.
- **Lower Fuel Consumption:** RL models reduce fuel consumption by 15 per cent compared to conventional methods and make more energy-efficient routing decisions [11]. Most of the companies in the logistics sector regard fuel efficiency as a cost-saving and sustainability concern. The RL model allows cutting down on superfluous travel and dynamically optimising routes, thereby reducing fuel usage and translating into lower operational costs and less carbon emissions. Such a fuel-saving feature is particularly useful in scenarios where demand may vary and applying state-of-the-art procedures can result in additional movements and increased fuel use.

- **Higher Delivery Timeliness:** RL models have the potential to improve delivery timeliness, even under discontinuous circumstances such as sudden traffic fluctuations or route blockages [14]. This timeliness enhancement highlights the RL model's ability to rapidly adapt and recalculate routes when real-time data becomes available, helping maintain delivery windows within their schedules. Unlike the conventional approach, where sudden changes may pose issues, the RL approach's adaptability to disruptions may be more reliable in meeting the delivery schedule and overall service quality.
- **Better Computational Efficiency:** RL-based methods have considerably lower computational time requirements than Mixed-Integer Linear Programming (MILP) in route planning, and in some cases, they can improve by up to 80 per cent [16]. This reduced computation time, making RL models highly applicable in real-time scenarios where decisions must be made instantly. In dynamic cases, where the rerouting may need to be done quickly to be able to match the changes during the last minute, the efficiency of the RL model will allow it to deliver timely rerouting solutions, and the traditional optimisation models, such as MILP, might not be fast enough to react in real-life applications.

On the whole, these results demonstrate the potential of RL-based methods as a promising solution to the dynamic VRP, outperforming traditional methods across key metrics such as travel time, fuel consumption, delivery timeliness, and processing speed. It is this flexibility and efficiency that make RL models especially relevant to current-day logistics, where real-time responsiveness is an increasingly significant demand.

9. Conclusion

The RL-based method for dynamic VRP shows significant improvements over conventional methods, achieving better performance in dynamic settings. The study contributes to the prospects for the use of reinforcement learning (RL) in logistics and lays the groundwork for further research and practical implementation. By incorporating machine learning methods, namely RL, the VRP community can develop more efficient, flexible, and cost-effective solutions to the complexity of real-world routing problems. Regarding the future of research, this study's findings offer interesting avenues for further exploration. The idea of researching multi-agent reinforcement learning (MARL) might enable decentralised decision-making, where each vehicle becomes an autonomous agent, potentially improving coordination and enabling the scaling of large fleets. Additionally, integrating RL with hybrid optimisation approaches, such as combining it with traditional algorithms like MILP or metaheuristics, could further improve computational efficiency and routing quality. More sophisticated architectures, such as Graph Neural Networks (GNNs) and Transformers, offer promising avenues for modelling intricate relationships in space and time in VRP, potentially enabling the model to dynamically reprioritise evolving delivery requirements.

Moreover, extending RL models to incorporate stochastic elements, such as random demand changes or traffic variability, would improve robustness, allowing models to make resilient decisions under uncertainty. Applying these RL-based VRP solutions to autonomous vehicle routing could further revolutionise logistics, as autonomous fleets make independent routing decisions in real time and adapt to environmental conditions. Sustainability-focused VRP solutions represent another critical research direction, with the potential to prioritise eco-friendly routing that aligns with industry goals for reduced carbon emissions and regulatory demands for greener operations. Finally, meta-reinforcement learning, which empowers models to solve problems quickly as new conditions arise, might also enable VRP models to learn how to learn and be more versatile in dynamically shifting environments. This research demonstrates the transformative potential of RL in VRP. It lays the groundwork for further progress that will lead to disruptive, even game-changing, changes in the logistics industry through the creation of incredibly adaptive, sustainable, and efficient routing methods.

Acknowledgment: The authors would like to express their sincere gratitude to Sultan Moulay Slimane University for providing academic support and a collaborative research environment.

Data Availability Statement: Data underlying this research may be accessed by contacting the corresponding author upon reasonable request.

Funding Statement: The research and preparation of this manuscript were carried out independently by the authors and did not receive funding from any public, private, or commercial sources.

Conflicts of Interest Statement: The authors declare no conflicts of interest and confirm that no public, private, or commercial funding influenced the research or manuscript preparation.

Ethics and Consent Statement: The study was conducted in compliance with established ethical standards. Informed consent was obtained from all participants, and strict measures were followed to ensure anonymity and protect participant information.

References

1. G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, no. 1, pp. 80–91, 1959.
2. J. E. Mendoza, A. Montoya, C. Guéret, and J. G. Villegas, "A multi-space sampling heuristic for the green vehicle routing problem," *Transportation Research Part C: Emerging Technologies*, vol. 70, no. 4, pp. 113–128, 2016.
3. D. Cattaruzza, N. Absi, D. Feillet, and J. González-Feliu, "Vehicle routing problems for city logistics," *EURO Journal on Transportation and Logistics*, vol. 6, no. 1, pp. 51–79, 2017.
4. G. Laporte, "The vehicle routing problem: An overview of exact and approximate algorithms," *European Journal of Operational Research*, vol. 59, no. 3, pp. 345–358, 1992.
5. G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Operations Research,* vol. 12, no. 4, pp. 568–581, 1964.
6. B. Eksioglu, A. V. Vural, and A. Reisman, "The vehicle routing problem: A taxonomic review," *Computers & Industrial Engineering*, vol. 57, no. 4, pp. 1472-1483, 2009.
7. M. Gendreau, A. Hertz, and G. Laporte, "A tabu search heuristic for the vehicle routing problem," *Management Science,* vol. 40, no. 10, pp. 1276–1290, 1994.
8. M. Dorigo and L. M. Gambardella, "Ant colonies for the travelling salesman problem," *BioSystems*, vol. 43, no. 2, pp. 73–81, 1997.
9. B. Licaj and D. A. Karras, "Integral smart vehicle automation framework using IoT, modern software, and infrastructure," *AVE Trends in Intelligent Computing Systems*, vol. 1, no. 3, pp. 128–141, 2024.
10. Y. Marinakis, A. Marinaki, and A. Migdalas, "Particle swarm optimization for the vehicle routing problem: A survey and a comparative analysis," in Handbook of Heuristics, *Springer International Publishing*, Cham, Switzerland, 2018.
11. P. P. Anand, G. Jayanth, K. S. Rao, P. Deepika, M. Faisal, and M. Mokdad, "Utilising hybrid machine learning to identify anomalous multivariate time-series in geotechnical engineering," *AVE Trends in Intelligent Computing Systems*, vol. 1, no. 1, pp. 32–41, 2024.
12. M. Nazari, A. Oroojlooy, L. V. Snyder, and M. Takác, "Reinforcement learning for solving the vehicle routing problem," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst. (NeurIPS),* Montréal, Canada, 2018.
13. D. Yan, Q. Guan, B. Ou, B. Yan, Z. Zhu, and H. Cao, "A deep reinforcement learning-based decision-making approach for routing problems," *Appl. Sci.,* vol. 15, no. 9, pp. 1–19, 2025.
14. A. K. R. Ayyadapu, "Scalable machine learning approaches for real-time big data processing in IoT networks," *AVE Trends in Intelligent Computer Letters*, vol. 1, no. 2, pp. 51–61, 2025.
15. W. Kool, H. V. Hoof, and M. Welling, "Attention, learn to solve routing problems!" in *Proc. Int. Conf. Learning Representations (ICLR),* Louisiana, United States of America, 2019.
16. J. Zhao, M. Mao, X. Zhao, and J. Zou, "A hybrid of deep reinforcement learning and local search for the vehicle routing problems," *IEEE Trans. Intell. Transp. Syst.,* vol. 22, no. 11, pp. 7208–7218, 2021.
17. A. Bogyrbayeva, M. Meraliyev, T. Mustakhov, and B. Dauletbayev, "Machine learning to solve vehicle routing problems: A survey," *IEEE Trans. Intell. Transp. Syst.,* vol. 25, no. 6, pp. 4754–4772, 2024.
18. R. Shahbazian, L. D. P. Pugliese, F. Guerriero, and G. Macrina, "Integrating machine learning into vehicle routing problem: Methods and applications," *IEEE Access,* vol. 12, no. 7, pp. 93087–93115, 2024.
19. N. Mazyavkina, S. Sviridov, S. Ivanov, and E. Burnaev, "Reinforcement learning for combinatorial optimization: A survey," *Computers and Operations Research,* vol. 134, no. 10, p. 105400, 2021.
20. D. Ujwal, M. S. Koti, and R. B. Sulaiman, "Machine learning approach for cyberbullying identification: A gradient boosting and Flask-based implementation," *AVE Trends in Intelligent Computer Letters*, vol. 1, no. 2, pp. 95–103, 2025.
21. C. K. Joshi, T. Laurent, X. Bresson, and Y. LeCun, "An efficient graph convolutional network technique for the travelling salesman problem," *arXiv preprint*, 2020. Available: https://arxiv.org/abs/1906.01227 [Accessed by 12/11 /2024].
22. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NeurIPS)*, California, United States of America, 2017.
23. K. Danach, "Reinforcement learning for dynamic vehicle routing problem: A case study with real-world scenarios," *Int. J. Commun. Netw. Inf. Secur. (IJCNIS),* vol. 16, no. 3, pp. 580–589, 2024.
24. A. Arishi and K. Krishnan, "A multi-agent deep reinforcement learning approach for solving the multi-depot vehicle routing problem," *Journal of Management Analytics,* vol. 10, no. 9, pp. 1–23, 2023.
25. S. Kadyrov, A. Azamov, Y. Abdumajitov, and C. Turan, "Deep reinforcement learning for dynamic vehicle routing with demand and traffic uncertainty," *Operations Research Perspectives,* vol. 15, no. 8, pp. 1–12, 2025.