IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 21, 2021, accepted November 27, 2021, date of publication November 30, 2021,
date of current version December 17, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3131799

A Hybrid Reinforcement Learning-Based Model for the Vehicle Routing Problem in Transportation Logistics

THANANUT PHIBOONBANAKIT[1,2], TEERAYUT HORANONT[2], VAN-NAM HUYNH[1], (Member, IEEE), AND THEPCHAI SUPNITHI[3]

[1]School of Knowledge Science, Japan Advanced Institute of Science and Technology, Nomi, Ishikawa 923-1292, Japan
[2]School of Information, Computer, and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University, Pathum Thani 12000, Thailand
[3]National Electronics and Computer Technology Center (NECTEC), National Science and Technology Development Agency, Pathum Thani 12000, Thailand

Corresponding authors: Teerayut Horanont (teerayut@siit.tu.ac.th) and Van-Nam Huynh (huynh@jaist.ac.jp)

This work was supported in part by an Excellent Thai Student Scholarship from the Sirindhorn International Institute of Technology, Thammasat University, under Grant ETS-G-S1Y17/061; in part by the SIIT-JAIST Dual-Degree Scholarship Program; in part by the U.S. Office of Naval Research Global under Grant N62909-19-1-2031; and in part by the Center of Excellence in Urban Mobility Research and Innovation, Thammasat University, Pathum Thani, Thailand.

ABSTRACT Currently, the number of deliveries handled by transportation logistics is rapidly increasing because of the significant growth of the e-commerce industry, resulting in the need for improved functional vehicle routing measures for logistic companies. The effective management of vehicle routing helps companies reduce operational costs and increases its competitiveness. The vehicle routing problem (VRP) seeks to identify optimal routes for a fleet of vehicles to deliver goods to customers while simultaneously considering changing requirements and uncertainties in the transportation environment. Due to its combinatorial nature and complexity, conventional optimization approaches may not be practical to solve VRP. In this paper, a new optimization model based on reinforcement learning (RL) and a complementary tree-based regression method is proposed. In our proposed model, when the RL agent performs vehicle routing optimization, its state and action are fed into the tree-based regression model to assess whether the current route is feasible according to the given environment, and the response received is used by the RL agent to adjust actions for optimizing the vehicle routing task. The procedure repeats iteratively until the maximum iteration is reached, then the optimal vehicle route is returned and can be utilized to assist in decision making. Multiple logistics agency case studies are conducted to demonstrate the application and practicality of the proposed model. The experimental results indicate that the proposed technique significantly improves profit gains up to 37.63% for logistics agencies compared with the conventional approaches.

INDEX TERMS Freight, intelligent transportation, logistics, policy, reinforcement learning, vehicle routing problem.

I. INTRODUCTION

Vehicle routing problem (VRP) models are developed to achieve vehicle routing that yields minimum traversal costs while simultaneously considering vehicle capacity, energy consumption, and time windows. VRP is widely applied to maximize the efficiency of delivery tasks for real-world logistics agencies, and the results are promising even though they are highly complex in practice. Such tasks are subjected to environmental uncertainties, such as delivery incidents (e.g., postponement or cancellations), road and traffic conditions, changing customer requirements, and fleet resources. Using conventional VRP optimization techniques may have drawbacks.

Conventional approaches (e.g., mixed-integer linear programming (MILP) and general reinforcement learning) cannot handle the dynamic nature of transportation systems that change throughout the day. These approaches generally assume an ideal transport environment and may not be robust when applied to real-world problems. Hence, penalties may

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

be imposed because they neglect the possibility of delivery failure, traffic delays, and other constraints. When all environmental factors are included to formulate a VRP using MILP, the added penalties often result in task infeasibility, owing to its highly constrained nature.

Although numerous efforts have been made to resolve these issues with modern models (e.g., dynamic and multimodal network models [1], [2]), studies have only considered road-network and traffic conditions, which is beneficial to general VRP but is often ineffective for transportation logistics problems. Hence, the need to address environmental uncertainties remains. In the presented paper, we aim to develop a novel methodology for solving VRP in transportation logistics. The proposed model hybridizes reinforcement learning (RL) and tree-based regression methods to handle the environmental uncertainties discussed while optimizing vehicle-routing tasks.

RL refers to the training of machine-learning (ML) models to make sequential decisions. Generally, RL problems involve learning what actions are to be made and how to transform states into actions. The technique is to maximize a numerical reward signal for solving particular problems. According to [3], RL consists of model-free and model-based methods. A model-free RL enables an RL agent to explore the environment via trial and error. However, doing so does not guarantee that a solution can be employed. The agent may adapt to the new environment to an extent, but it potentially will become stuck in an infinite search. This occurs when the model searches for solutions without obtaining feedback from the external environment. On the other hand, in the model-based RL principle, the ML model reflects the external environment, for which collected data are used as inputs to train the agent at each iteration. Therefore, a model-based RL can be trained and dynamically adapted to environmental changes without redefining the model; it is adjustable and is driven by the data.

This study integrates the advantages of both model-free and model-based RL principles into a ''hybrid model'' technique. Instead of using only model-free RL like other studies, model-based RL is incorporated to enhance the model's adaptability when dealing with uncertainties. The model-based RL has the exploitative ability to create and store known events of the transport environment. In this study, the criteria for evaluating the transport environment consist of feasibility, efficiency, and fairness. These variables are assessed by a reward processing unit and fed into the RL as rewards. Then, the RL agent employs the information as experience in performing vehicle route optimization tasks. This experience helps the agent adjust its actions to compensate for changes.

The use of RL and MILP in VRP research has been well-documented. However, the discussed uncertainties of the transport environment have not been well-addressed. The main contribution of this paper is the proposal of a new methodology for minimizing vehicle traversal costs while incorporating model-free and model-based RL principles to tackle environmental complexities and uncertainties using data collected from the transportation environment. To the best of our knowledge, our study is the first to hybridize model-free and model-based RLs into an optimized model for solving VRP transportation logistics.

The proposed method illustrates the transport environment using the collected data. Then, the reward unit is equipped to evaluate the environment with feasibility, efficiency, and fairness as criteria. The reward is positive under neutral conditions and is negative under abnormal conditions. The reward, state, and action are then modeled using the tree-based regression method to be applied as experiences for RL. When the RL agent initiates vehicle route optimization, the state and action are fed into the tree-based regression model to assess whether the routes are feasible for the given environment. If not, the vehicle routing tasks are adjusted accordingly. After each iteration, the route state action is updated as experiences so that the RL agent can learn to avoid unsupportable actions in later stages. The procedures is repeated iteratively until the maximum iteration is reached; then, the optimal vehicle route is updated and can be utilized to assist in decision making.

The remainder of this paper is organized as follows. Section II provides a literature review related to vehicle route optimization. Section III highlights the problem statement, its significance, and the motivation of this research. Section IV describes the proposed model and the methods used to conduct data analysis and processing. Then, our vehicle route optimization experiment is explained. Section V presents the case studies used to evaluate the proposed model. Section VI illustrates the factors involved in the vehicle route optimization process and the comparative experiment. Section VII provides a discussion of the experimental results. Finally, conclusions, research limitations, and possible research directions are given in Section VIII.

II. RELATED WORK

We categorize the existing works into conventional VRP and ML-based VRP. The advantages and disadvantages of these approaches are discussed in the latter part of each subsection.

A. CONVENTIONAL VRP

Generally, the objective of VRP is to minimize the travel distance and time required to deliver goods from a warehouse to a customer, subject to the vehicle capacity and service time window. VRP is classified as an NP-hard problem by its nature. The complexity of VRP is known to surpass the traditional traveling salesman problem (TSP). [4].

VRP is well-studied for its application and complexity. Considerable efforts have been made to develop VRP model formulations and obtain optimal vehicle routing. For example, a capacitated VRP manages vehicle capacity while handling specific customer demands [5]. This approach includes split-delivery VRP (SDVRP), which allows the demand to be split and a customer to be visited by a vehicle more than once [6]. The multi-depot VRP considers the

pick-up-and-delivery problem, wherein vehicles are required to visit multiple depots to fulfill customer demands. Several other models have been proposed, as shown in the survey by [7]. The survey pointed out the need to address complexity in VRPs. For this reason, the consideration of fuel cost was deemed crucial for logistics policy-making. Hence, several studies have developed VRP models to minimize fuel consumption, as shown in [8], [9] and the survey paper written by [4].

More recent studies have highlighted environmental sustainability so that VRPs may enhance long-term environmental benefits. This ''green VRP'' considers environmental impact factors (e.g., fuel consumption and pollution emissions) as the primary means to the end. [9] studied a VRP that considered fuel consumption and carbon emissions, incorporating fuel cost, carbon footprint, and vehicle usage costs into the conventional VRP problem. The study of green VRP also includes the works of [10]–[13].

Apart from green VRP, customer demand is a critically important factor. Therefore, numerous studies, such as [14]–[16], have proposed related improvements. Algorithms have been developed for monitoring changes in client demand and making decisions to rearrange routes [14], [16]. Other studies used ML to analyze past decisions via a decision-support system to make recommendations for future events [15].

Most related studies aimed to improve vehicle routing efficiency and practicality by considering real-world transportation conditions. However, when considering VRPs in practice, numerous restrictions must be enforced while simultaneously addressing surprise factors. Therefore, when applying conventional optimization techniques, some constraints may be relaxed to obtain feasible solutions. For instance, in [17], a particular set of constraints related to demand and route perspectives were relaxed by assuming that the shortest routes were not always the most economical, owing to traffic congestion. In this case, the traversal cost may increase when enforcing the shortest-only policy. When this constraint is strictly enforced, feasible solutions may not exist.

Travel time is crucial because a delay in goods delivery can cause numerous consequences, such as late fees and reductions in customer satisfaction. Heavier-than-normal traffic easily increases the fuel consumption of a vehicle, impacting the traversal costs and delaying deliveries. Some studies tried to resolve these issues. Musolino *et al.* [1] proposed a VRP model that calculates reliable travel time considering the regional clustering of the data of each road link. They then solved the VRP for optimal freight vehicle routing based on travel time. This study is similar to that of [2], wherein the results highlighted the significance of travel time when optimizing vehicle routing. Shi *et al.* also considered travel time in their model formulation, aiming to ensure that the delivery will be completed within the specified time window [18].

Based on the literature review, it is evident that many optimization approaches have been employed to solve VRPs. These approaches guarantee optimality and, to some extent, are capable of solving traversal cost minimization. However, when considering real-world logistics, modifications and assumptions must be made to the objective function and constraints must be imposed to obtain feasible solutions with precise approaches. For a highly complex problem with numerous constraints and restrictions, some elements cannot be represented by mathematical functions. Therefore, the formulation of a mathematical model becomes impracticable.

Furthermore, most optimization approaches assume an ideal transportation environment and omit uncertainties, which are likely to occur in practice, making those approaches less practical. The incorporation of ML and RL into VRP optimization models may enable more adaptive and flexible models when employed in the dynamic nature of transport environments. A discussion of this technique is provided in section II-B.

B. VRP MODELS DEVELOPED AROUND ML AND RL

To date, considerable research attention has been given to develop advanced approaches to solve more complicated VRPs while accounting for uncertainties in the transport environment. The consideration of these uncertainties enables VRPs to become more practical while improving their implementation ability in real-world settings. As discussed, MILP-based VRPs face difficulties adapting to changing environments because the model must be entirely reconstructed each time a variation occurs. Therefore, the computational costs becomes untenable. In light of this, the use of ML and RL to solve VRPs under uncertain environments can accommodate the adaptability of the model to changes. Hence, complete model reconstruction is not needed. The ML and RL approaches are applied to formulate the transportation graph network model to aid customer prioritization and minimize costs. It also allows the network to adjust as necessary using ML to index network instances.

Sutskever *et al.* were the first to use ML to solve the TSP. They proposed a sequence-to-sequence approach that employed a recurrent neural network (RNN) to predict the next possible nodes from the previously visited ones [19]. However, they did not account for location references between nodes. Thus, if the node was changed at a given time step, the model had to be completely reconstructed. Later, Vinyals *et al.* proposed a pointer network as an extension to Sutskever's model to supplement the location referencing feature. However, the model had to be reconstructed each time the customer demand was updated, resulting in significant computational costs [20]. Kool *et al.* developed an attention mechanism that enhanced the model proposed by Vinyals *et al.* and applied RL to train the network. They eliminated the need to reconstruct the model when variables were updated [21]. An attention layer was added above the node layer to handle the updating process instead of the node. Thus, there was no need to reconstruct the model at all. This model was initially developed only

for solving TSP. However, some modifications for VRP were later proposed.

These previous works did not consider networks consisting of sequences of nodes that are referenced by indices. Neither did they store any prior decisions for node selection. All possible inputs had to be calculated every time the model was executed, causing lengthy computational times. Therefore, Dai *et al.* adopted a structure that combined the vector method with Q-learning [22]. Instead of inputting data as sequences, they modeled the nodes as a graph and used Q-learning to store feasible solutions from past visits. This allowed the RL agent to use experiences to decide which customer to visit next at each iteration. Thus, the computational time was significantly reduced. Nazari *et al.* and Bello *et al.* developed models using RL to support VRP tasks [23], [24].

Unfortunately, the disadvantage of the models presented in this section is that the trial-and-error process cannot guarantee feasibility in real-world settings because the solution-searching process is conducted without any external environment information. In the following, we will briefly discuss the approaches recently proposed for addressing this issue.

C. VRP MODELS THAT COMBINES ML AND RL

The approach of combining ML and RL for VRP solving is called model-based RL. Moerland *et al.* [25] conducted an in-depth literature survey on this topic and accounted for numerous studies dedicating in the development of model-based RL. The authors showed that model-based RL could be classified into two types: model-based RL with a learning model and model-based RL with a known model. The difference between these two types is that the learning model offers a dynamic learning ability that updates the model concordant to changes from the environment. Therefore, it is more suitable for dealing with uncertain environments compared to the known model. The limitation of the known model is that when the agent keeps using information from a previously trained model without detecting the changes in the environment, the model performance may decline.

In addition to the comparison among these two model types, the authors also demonstrated the benefits of using model-based RL, which can enhance data efficiency, targeted exploration, and improved stability. Furthermore, combining model-based and model-free updates can increase the model learning rate with the model-based part [25].

Another interesting work based on model-based RL was conducted by Drori *et al.* [26]. They extend the so-called AlphaD3M [27] using a pipeline grammar and a pre-trained model to find the optimal machine learning pipelines for a OpenML dataset and tasks (e.g., classification and regression). Their experimental results were impressive. The proposed system can be classified as model-based RL with a known model, which discards uncertainty from the environment. Therefore, it might not be suitable for solving real-world problems in domains with high uncertain factors, such as network planning, mobility analysis for transportation and telecommunication.

From the survey paper [25], we discovered opportunities to apply model-based RL to VRP solving. This topic remains an active research area and required further attention, especially for solving VRP in real-world scenarios with uncertain environments.

A study by Mao *et al.* [28] demonstrated that a tree-based regression method could assist RL achieve balance between exploration and exploitation. Furthermore, the model optimized vehicle routing based on collected data by employing a new reward function.

Numerous other methodologies were proposed to integrate ML and RL for solving VRP, as reported in literature surveys conducted by [29], [30]. They pointed out that the computational time can be significantly reduced when applying ML and RL to solve VRPs. The model structure can also be altered when dealing with new problems and requirements. Especially in real-world problems, the ML and RL enable VRP models to efficiently adapt to changes driven by data containing experiences learned from preceding decision patterns.

When RL is employed to solve VRPs, the RL agent (i.e., a component that guides RL to learn how its action is performed) must be considered. Based on the RL principle, this component is referred to as a reward function. Generally, an RL reward function for VRP uses a positive and negative route cost for the reward and penalizes actions accordingly. This type of reward functions was developed in [29], [31], [32]. However, more aspects are required for a thorough evaluation when considering real-world problems not limited only to routing costs. Related examples include traffic conditions and resource utilization. Hence, there is a need to develop a new reward function for RL to solve real-world problems. Cruciol *et al.* [33] proposed one for learning to manage air traffic flow. Their work accounted for capacity, aircraft distribution, and financial factors, and their findings highlighted the practicality of the proposed reward function in a real-case study.

Based on these reviews, it appears that combining ML and RL can effectively solve and optimize VRP similarly to conventional approaches. However, the VRP models which combine ML and RL are more flexible for modification than the conventional VRP models. With a model-free RL, the trial-and-error process cannot guarantee feasibility in real-world settings because the solution-searching process is conducted without any external environment information. However, model-free RLs were applied by most related works. This is why we assert that a model-based RL, which accounts for the external environment, should be incorporated.

Using the model-based RL principle, the external environment is created to train the agent using the collected data from each episode. Therefore, a model-based RL is trained to dynamically adapt to any new environmental changes without redefining the model, owing to data-driven capabilities. Similar findings were well-noted in [3], [25], [29], [30], [34].

D. SUMMARY OF REVIEW

Based on the literature reviews shown in sections II-A – II-C, the VRP models developed based on conventional VRP and combination of ML and RL approaches can solve VRP to some extent. However, the limitations of these approaches are as follows. First, the transportation environment is generally assumed to be always feasible for employing the recommended solution. However, the recommended solutions may not be applicable to the real situation, imposing adverse effects on the operational cost owing to unsuccessful and delayed deliveries. Second, most models developed by ML and RL are only suitable for dealing with customer-demand uncertainties. These models may not be as applicable to other factors, including traffic congestion, changing requirements, and resource availability.

Therefore, in this study, we utilize the methodology proposed by [1], [23], [33] to develop a vehicle route optimization model that is dynamically adaptable to transportation environmental uncertainties without the need to restructure the model each time. Additionally, the novel model-based RL is equipped to create a transportation environment for RL training. We apply a tree-based regression model adapted from [28] to resolve the infinite-loop searching issue, which is a major drawback of the model-free RL. Notably both model-free and model-based RLs have shortfalls, and previous works were developed based on one of these approaches only. Our proposed methodology bridges this research gap by integrating model-free and model-based RLs, allowing the new model to solve dynamic transportation logistics VRPs more effectively.

III. PROBLEM DEFINITION

This study addresses SDVRP within an uncertain transport environment. The considered uncertainties consist of delivery incidents (e.g., postponement or cancellations of deliveries) and issues caused by road-network traffic conditions, changing customer requirements, and the logistics agency fleet. A time-dependent variable is also included to ensure that goods are delivered within a specific time window.

The SDVRP is generally defined as a graph, $G = (\mathcal{V}, E)$, with a vertex set, $\mathcal{V} = \{0, 1, \ldots, n\}$, where 0 denotes the depot, all other vertices denote customers that each vehicle is required to visit, and E denotes a set of edges. Each vehicle must start and end a given route at the depot. The indices, parameters, and variables used for problem formulation are described in Table 1.

The objective of this SDVRP is to minimize the total traversal cost incurred from making deliveries to each customer. In the SDVRP setting, customer demands may exceed the capacity of a vehicle; therefore, there is a need to split these demands into sub-demands so that multiple vehicles may be used to deliver goods to customers until all demands are satisfied while keeping the traversal costs minimal. The optimization model aims to search for optimal customer-demand handling and vehicle assignments with minimum costs. Formally, the SDVRP objective function can be expressed as follows:

$$\min \sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{v=1}^{m} c_{ij}x_{ij}^{v} \tag{1}$$

$$\text{subject to: } \sum_{i=0}^{n}\sum_{v=1}^{m} x_{ij}^{v} \geq 1; \quad j = 0, \ldots, n \tag{2}$$

$$\sum_{i=0}^{n} x_{ip}^{v} - \sum_{j=0}^{n} x_{pj}^{v} = 0; \quad v = 1, \ldots, m;\ p \in \mathcal{V} \tag{3}$$

$$\sum_{i\in V}\sum_{j\in V} x_{ij}^{v} \leq |V| - 1; \quad v = 1, \ldots, m;\ V \subseteq \mathcal{V}^{+} \tag{4}$$

$$\sum_{i=0}^{n} x_{ip}^{v} = \sum_{j=1}^{n} x_{pj}^{v}; \quad v = 1, \ldots, m;\ p \in \mathcal{V} \tag{5}$$

$$y_{iv} = h_{i_1}^{v} dm_i^1 +, \ldots, +h_{i_k}^{v} dm_i^k; \quad i = 1, \ldots, n; \quad v = 1, \ldots, m \tag{6}$$

$$\sum_{v=1}^{m} y_{iv} = dm_i; \quad i = 1, \ldots, n \tag{7}$$

$$\sum_{i=1}^{n} y_{iv} \leq Q_v; \quad v = 1, \ldots, m \tag{8}$$

$$dt_v = \sum_{i=0}^{n} se_i + t_{ij}, \forall j \in V; \quad v = 1, \ldots, m \tag{9}$$

$$dt_v \leq T; \quad v = 1, \ldots, m \tag{10}$$

$$ct_i + t_{ij} \leq \hat{l}_{ij}, \quad ct_i + t_{ij} \leq dl_{ij}^{v};\ i = 0, \ldots, |\mathcal{V}|;\ j = 1, \ldots, |\mathcal{V}^{+}|; \quad v = 1, \ldots, m \tag{11}$$

where the constraints are divided into three categories: routing, customer demand, and time constraints. These are explained next.

Constraints (2) to (4) are the routing constraints. Constraint (2) ensures that each customer location, i, will be visited at least once by vehicle v until the customer demand is satisfied. Constraint (3) compares routing similarities. If the comparison result equals zero, then the routes are considered identical. This constraint aims to prevent recommending the same routes with inverted directions for vehicle v, whereas Constraint (4) eliminates sub-tours of other recommended routes. The current routing sequence should not resemble previously recommended routes apart from the depot location.

Constraints (5) to (8) are related to customer demand. Constraint (5) indicates that the demand can be delivered to customer i only if vehicle v passes through route i. If the delivery by vehicle v is less than or equal to the demand of customer i, the constraint ensures that vehicles farther away from the customer will not be recommended. Constraint (6) allows demand for customer i to be split, but each order is not detachable. Constraint (7) guarantees that all demands for

TABLE 1. Set of parameters and variables.

Notation	Description
i, j	The origin i and destination $j \in \mathcal{V}$
n	The number of nodes in $\mathcal{V}$
m	The number of vehicles that need to be used
p	The index of node in $\mathcal{V}$, i.e., $p = 0, \ldots, n$
v	Vehicle $v \in \{1, \ldots, m\}$
K	The number of good units that need to be delivered
k	Good unit $k \in \{1, \ldots, K\}$
Q_v	The vehicle v's capacity
$\mathcal{V}^+$	Vertices in the set of vertices $\mathcal{V}$ except the depot, i.e., $\mathcal{V}^+ = \mathcal{V} \setminus \{0\}$
V	Subset of vertices in $\mathcal{V}^+$, i.e., $V \subseteq \mathcal{V}^+$
T	Limit of working hours per trip by limiting consecutive hours
c_{ij}	The traversal cost of route from i to j
t_{ij}	The travel time of route from i to j
y_{iv}	The demand of i delivered by vehicle v
ct_i	The current time at vertex $i \in \mathcal{V}$
dm_i	The demand of vertex $i \in \mathcal{V}^+$
dt_v	The accumulative working hours of vehicle v
dl_{ij}^v	The deadline for visit vertex j from i of vehicle v
se_i	Service time of delivery vehicle at i
$[l_{ij}, \hat{l_{ij}}]$	Arrival time at vertex j from vertex i, where l_{ij} is the best arrival time and $\hat{l_{ij}}$ is the maximum arrival time that the customer can accept
x_{ij}^v	Route selection $= \begin{cases} 1, & \text{if } v \text{ travel from } i \text{ to } j \\ 0, & \text{otherwise} \end{cases}$
$h_{i_k}^v$	Goods selection $= \begin{cases} 1, & \text{if } v \text{ delivers goods unit } k \text{ to customer } i \\ 0, & \text{otherwise} \end{cases}$

each customer, *i*, are satisfied by the delivery from vehicle *v*. The total items from all vehicles, *v*, delivered to customer *i* should equal the customer's demand, *i*. Constraint (8) indicates that the delivery amount of each vehicle, *v*, cannot exceed the vehicle capacity (Q_v). This constraint ensures that vehicles having lower capacity than the delivery size are not chosen. The delivery is to be split and shared among vehicles. In this study, the capacities of all vehicles are identical.

Constraints (9) to (11) are the time constraints. Constraint (9) refers to dt_v as the accumulated travel time when vehicle *v* arrives at customer *j*. Constraint (10) ensures that incidents do not occur. It also includes the delay in goods handling, and drivers do not exceed the limit of working hours per trip by limiting consecutive hours of vehicle *v* to 8 h. Constraint (11) is the time-window limit for each customer, *j*, requiring the vehicle to arrive before a given deadline (dl_{ij}^v). Therefore, the accumulative travel time should be less than or equal to the time-window limit. Otherwise, the trip is not recommended.

It can be seen that the formulation of the general SDVRP problem already requires numerous constraints. However, the model can become more complex because more factors must be considered simultaneously under real-world settings. Additionally, those factors may not be wholly represented by mathematical equations. Therefore, instead of directly incorporating these elements as a general MILP, we include them in the RL problem formulation for ease of interpretation. The RL formulation is demonstrated by the following steps.

The SDVRP was first formulated as a MILP using Equations (1) to (11). It was then transformed into an RL problem using the RL formulation derived in [35]. The RL problem formulation consists of a tuple, $< S, I, A, T, G, R, C >$, and their definitions are as follows:

- **States** (*S*): *S* is a finite set of states (*s*) of the environment. The elements in *S* are defined as $s = [latitude, longitude, demand, vehicle_{avail.}, driver_{avail.}, vehicle_{maintenance}]$. For simplicity, states are the current environment with which the RL interacts and provides a set of partial solutions to the problem (e.g., a partially constructed route for the VRP problem).
- **Initial state** (*I*): This is the current vehicle location with current fleet availability status specified by the company. At this stage, the element consists of six dimensions: $latitude$, $longitude$, $demand$, $vehicle_{avail.}$, $driver_{avail.}$, and $vehicle_{maintenance}$.
- **Actions** (*A*): *A* is one of possible actions performed by the RL (e.g., choose, swap, and skip). In this context, *A* is the RL agent's actions by choosing customer location to be visited, rearranging visiting order, and making a delivery from the current location in the current state, *s*. The delivery should satisfy customer requirements.

TABLE 2. Variable used to calculate the profit and traversal cost.

Notation	Value	Description
$incomes^{dow}$	17,000 THB	Payment made by the customer to the logistics agency.
dow	$1, \ldots, 7$	Days of the week (Sunday to Friday)
vr	$1, \ldots, N$ vehicle	Vehicle used in the current route
N	45 vehicles	The number of vehicle used in a day
d	From the routing in Kilometers	Distance of travel
wt	From the routing in Hours	Waiting time required at the customer location until the delivery is completed
tr	From the routing in Hours	Travel time aggregated from GPS data
td	From the routing in Minutes	Traffic congestion time aggregated from GPS data
mt	1.35 THB/Kilometers	Maintenance cost
ins	105 THB	Insurance cost
de	2.8 THB/Kilometers	Depreciation of vehicle cost
sc	100 THB	Service cost of handling the containers
f	7.29 THB/Kilometers	Fuel cost of each trip
f_{price}	19.89 THB/Liter	Fuel cost per liter
l	300 THB/Day or 12.5 THB/Hours.	Driver wage
al	200 THB/Day or 8.33 THB/Hours	Driver assistant wage
ovt	1,000 THB/Day or 41 THB/Hours	Overtime cost

- **Transition model** (T): T is the probability of state transitioning from $S \times A \times S \rightarrow [0, 1]$. This statement is defined in the "The Proposed Optimization Model by RL" section in Equation (29). Therefore, in each state, $s \in S$, the best possible action to choose is calculated from the optimal policy, $\pi^*(s) \in A$.
- **Goal test** (G): G determines whether deliveries satisfy customer requirements and whether the traversal cost of each trip, defined in Equation (1), is minimized. It is also used to justify whether actions follow the requirements from Equations (2) to (11). Later, these constraints become the utility evaluation functions for the RL problem considering real-world elements.
- **Reward** (R): For each state transition, the reward function is defined as $S \times A \times S \rightarrow G(Z)$, where the definition of $G(Z)$ is defined in Equation (14) of the "Reward Processing for the RL" section.
- **Path cost** (C): C is a traversal cost incurred from $S \times A \times S$ defined in Equation (12).

Based on the "Path Cost" defined in the RL problem formulation, the traversal cost and profit per day are calculated using Equations (12) and (13), respectively.

$$\begin{aligned} traversal_{cost}^{dow} = \sum_{vr=1}^{N} & ((d^{vr} \times mt) + ins^{vr} \\ & + (d^{vr} \times de) + sc^{vr} + (d^{vr} \times f) \\ & + \left(\left(\frac{td^{vr} \times 20}{1,000}\right) \times f_{price}\right) \\ & + (tr^{vr} \times l) + (tr^{vr} \times al) \\ & + (ovt^{vr} \times wt)) \end{aligned} \tag{12}$$

The value of each variable in Equations (12) and (13) are listed in Table 2. They were derived from an actual operational report to calculate operational cost and profit. Equation (12) computes the total traversal cost of the routing each day, dow. This cost is the summation of the traversal cost obtained from all vehicles used for delivery during the day. The traversal cost is then used to calculate the profit in Equation (13):

$$profit^{dow} = incomes^{dow} - traversal_{cost}^{dow}. \tag{13}$$

The main currency used in this paper is Thai Baht (THB). The remaining parameters are generic and widely adopted by various logistics agencies when computing their costs. Thus, no adjustment is required. However, the currency must be modified accordingly when applied to optimize other countries' vehicle routes.

The SDVRP problem is solved assuming an uncertain transport environment. The SDVRP problem with time constraints is formulated using Equations (1)—(11). Owing to the difficulties representing environmental uncertainties with mathematical formulations, the general SDVRP tree-based regression method is applied to illustrate the uncertainties in the RL for solving SDVRP. The RL is equipped to solve SDVRP to minimize the traversal cost under the objective function and constraints defined in Equations (1)—(11). Furthermore, the RL utilizes the environmental states information obtained from the tree-based regression model to discover and adjust its strategies according to the current transport environment. Hence, the optimization result recommended by RL is dynamically adaptable to variations from the environment. When the minimum traversal cost is obtained, the recommended route and the total profit are returned. The experimental result is then analyzed in terms of profit improvement.

The problem definition and all essential parameters are presented in this section, and the mechanism to solve the SDVRP is explained in the next section.

IV. METHODOLOGY

This section outlines the procedures of developing the proposed RL-based SDVRP. The methodology consists of

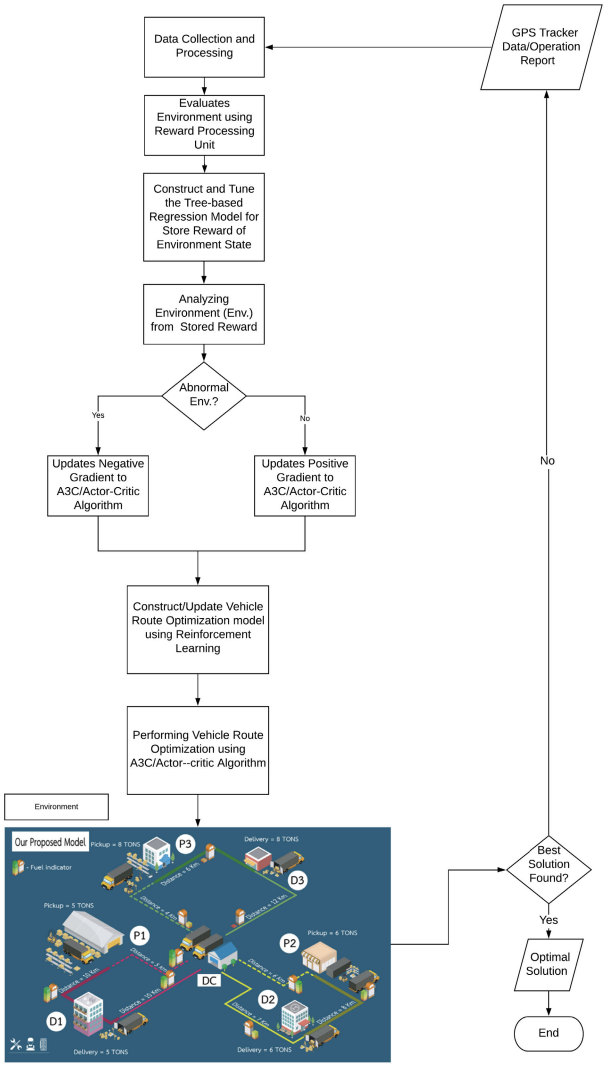


FIGURE 1. Demonstration of the proposed methodology using reinforcement learning and tree-based regression models to solve the vehicle routing problem.

five steps: 1) problem transformation; 2) data collection; 3) reward function formulation and environment evaluation; 4) optimization using RL; and 5) case studies. The methodology framework is illustrated in Figure 1.

A. DATA COLLECTION AND FEATURE ENGINEERING

The data used in this study were collected from two sources: 4G Long-term Evolution Global Positioning System trackers installed on vehicles of the logistics agency and 6-GB of their operation reports. The data collection was conducted from January 2017 to July 2019 to create the transport environment and to train the RL as the vehicle route optimization tasks are performed.

The data specifications are shown in Tables 3–4. Table 3 presents an example from the spatial-temporal data used to analyze driver behaviors daily routines of the vehicle route assignment. Table 4 presents the monthly operation statistics.

TABLE 3. Demonstration of the spatial-temporal data.

Name	Value	Data Type
DeviceID	213L2017000968	String
Latitude	13.68691	String
Longitude	100.46526	String
Speed	45.4	Float
Distance	69.2	Float
Time-spend	79	Integer
Time-stamp	2018-01-01 18:50:23	Time-Stamp

TABLE 4. Demonstration of the raw transport operation report.

Name	Value	Unit
Date	2/10/2018	-
Number of Available Vehicle	45	Vehicles
Number of Occupied Vehicle	7	Vehicles
Number of Vehicle with No Assigned Driver	8	Vehicles
Number of Vehicle with Back Order Work	0	Vehicles
Number of Vehicle in Maintenance	0	Vehicles
Number of Vehicle with Driver Taking Leave	0	Vehicles
Number of Total Requested from Client	45	Orders
Number of Request Received	45	Orders
Number of Order Canceled or Postponed (Import)	0	Orders
Number of Order Canceled or Postponed (Export)	0	Orders
Quarter of the Year	Q4	-

In addition, the relationships of these data variables are illustrated in Figure 3.

After data collection, the logistics management features (e.g., delivery success rate, utility, and productivity of resources) were constructed using feature engineering methods. Further details can be found in the logistics management strategies book written by [36], [37]. The data were then divided into training and testing datasets and deposited in storage. The training set consists of data collected from January 2017–July 2019, and the testing set is from May–July 2018. Note that the testing data is not included in the training data; it was collected during a site survey at our partner logistics agency, and the company helped validate the accuracy of the data from both drivers and staff.

After data separation, Power BI was used to determine the data relationships to measure each feature's fundamental statistics and to create links to other data attributes using time-stamps. Power BI is software used to discover relationships from multi-source data, which helps understand their coherency. It can also be used to perform data visualization tasks.

In addition to data specification, Figure 2 illustrates the statistics of goods deliveries from 2017 through 2019. From Figure 2, it appears that the delivery success rate was less than 50% when comparing the planned and actual delivery, indicating that the current logistics planning is not as practical.

From Figure 3, the relationships of the variables are somewhat correlated, and most are independent of each other. The analysis shows that the transportation environment has numerous variants and is quite complicated. This observation aligns with the previous discussion regarding the lack of efficiency of conventional VRP approaches in such

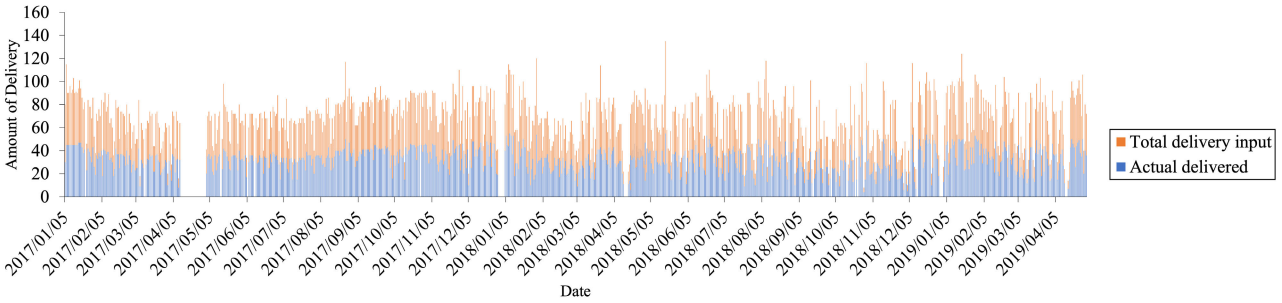


FIGURE 2. Demonstration of the goods deliveries statistics from the year 2017 until 2019. The orange bars represent the total intended deliveries, and the blue bars are the successful deliveries. Note that the blank areas denote Thailand holidays.

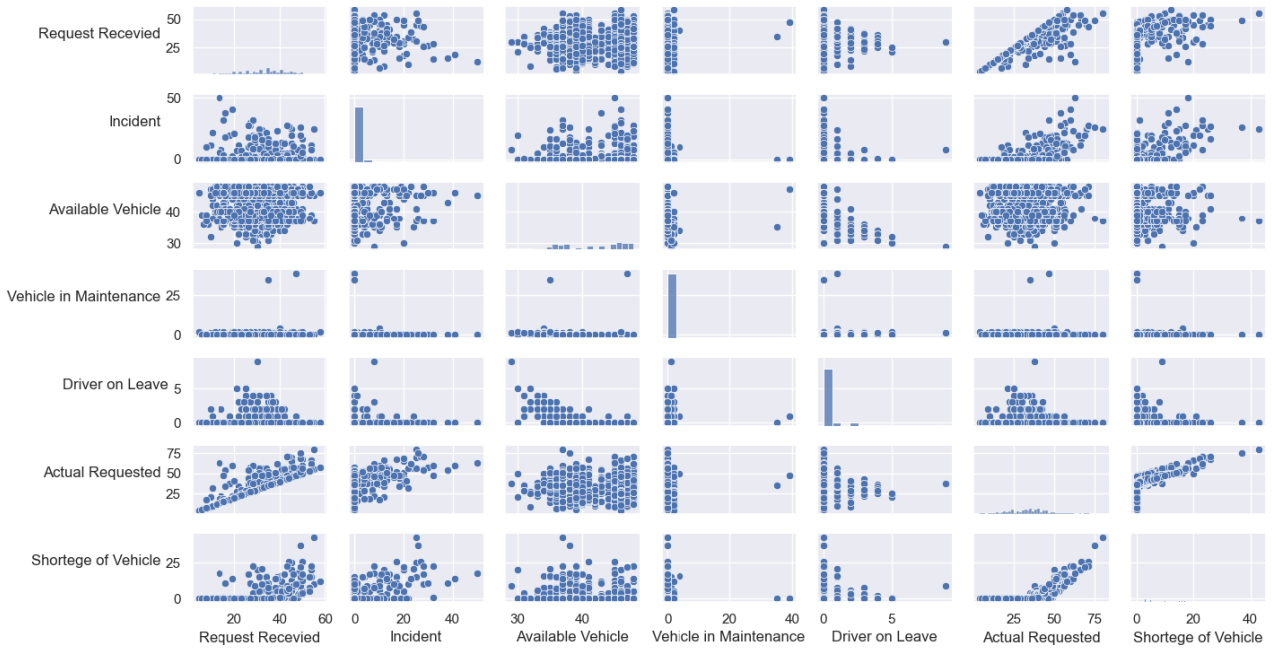


FIGURE 3. Demonstration of variables relationships from multi-source data.

complex environments. Thus, we proposed a reward function approach to evaluate the condition of the environment before RL optimization.

B. REWARD PROCESSING UNIT FOR RL

The transport environment created using the collected data reinforces RL so that it adequately performs the assigned tasks. The state of the environment is evaluated with the reward calculated from the previous interactions made by RL. If the optimal solution is returned and the environment state is normal, then the reward is positive. Otherwise, the reward is deducted in each episode of the RL environment with a negative reward. These values are communicated to the RL agent later with the actor and critic gradient. The full definition of this gradient is given in Section IV-D.

The data in Tables 3–4 are used as inputs for computing the reward for evaluating the transport environment. The reward functions determine the operational feasibility, efficiency, fairness, and the delivery and road-link factors of the task assignments. We adapted the methodology from [33] and the multiplicative utility function theorem proposed in [35]. The reward function, $G(Z)$, is defined in Equation (14):

$$G(Z) = \beta U(Z) + \delta T(Z) + \gamma I(Z), \tag{14}$$

where Z is the transport environment under evaluation, $U(Z)$ is the amount of vehicle usage, $T(Z)$ represents the number of delayed/non-delayed deliveries, and $I(Z)$ represents the fairness caused by the actions taken. These functions are known as utility evaluation functions. The reward function is the summation of the utility values from multiple data attributes input from the collected data. They represent different aspects of the environment. Thus, higher utility values indicate that the transport environment is in a normal state. Additionally, the weight of importances β, δ, and γ are assigned to the income in the utility evaluation function. In this study, β, δ,

and γ are set to 0.33 each because all utilities are equally essential.

The definition of normal and abnormal in this context is clarified as follows. Assuming that $G(Z)$ is less than 27.71, the environment is abnormal, and attention is required. Otherwise, the environment is normal. The value is the lower-bound value derived by performing a pre-computation using the 3-sigma method [38] on the data shown in Figure 2.

The utility functions inside the reward function are expressed using Equations (15)–(21). Note that Equations (14)—(21) are adapted from [33] because the air-traffic flow management and transportation planning shared similarities in policy development strategies. Therefore, only minor modifications were required for the proposed SDVRP.

- The operation evaluation function is designed to compute the efficiency ($p(x)$) for managing goods deliveries to the customer. It is evaluated in terms of a vehicle's capacity utilization. A larger $p(x)$ denotes more available vehicle space, signifying that more goods can be loaded onto the vehicle to utilize it fully. The excesses must be allocated to other available vehicles. The function is expressed in Equation (15):

$$p(x) = \Theta(c - W(x))^{(c-W(x))}, \tag{15}$$

  where c is the vehicle's maximum capacity (v), and $W(x)$ is the number of tasks assigned to the vehicle during period x. Θ is a function that returns zero if the input is negative, meaning that there is an excess usage of fleet resources. The goods-handling capability of a vehicle increases exponentially to the vehicle capacity. A larger $p(x)$ value indicates that the vehicle (v) can handle more deliveries.

  The total usage ($U(Z)$) is the summation of the fleets' vehicle capacity usage. A total usage is more than zero means that the RL agent does not violate any vehicle constraints in Equations (5)–(8). If a constraint is violated, the penalty is imposed on the RL agent as enforced by Equation (16):

$$U(Z) = \sum_{x \in Z} p(x). \tag{16}$$

- The delay evaluation function assesses the delivery performance ($d_{dt}(v)$) of each vehicle. This function eliminates solutions that violate time constraints, such as behind-schedule deliveries or prolonged waiting times at depot or customer locations. The computation is based on the departure, arrival, and traveling times based on road-network conditions, as indicated in Equation (17):

$$d_{dt}(v) = \Theta(1 + (t - \propto (t_{ad}, t_{aed}))), \tag{17}$$

  where t, t_{ad}, and t_{aed} denote the current time, actual departure time, and estimated departure time, respectively. $\propto$ is a function for estimating the departure and arrival times from the traffic estimation model, and v is the vehicle. Θ is a function that returns zero if the input is negative, indicating delayed deliveries.

  The total delivery performance is the summation of the delayed and non-delayed deliveries during period x. A positive total delivery performance indicates that the RL agent has not violated any time constraints shown in Equations (9)—(11). If a constraint is violated, the penalty is imposed on the RL agent as enforced by Equation (18).

$$T(Z) = \sum_{x \in Z} \sum_{v \in x} d_{dt}(v). \tag{18}$$

- The operation feasibility function determines the feasibility and fairness ($I(Z)$) of the solution suggested by the RL. The feasibility in this context reflects the possibility that the solution can be used without violating any constraint shown in Equations (2)—(4) and without any unsuccessful delivery. The operation feasibility function is expressed as Equation (19), consisting of two terms. The first expresses the equality of tasks assigned to each driver as obtained from Equation (20). A large equality value signifies that the tasks assigned to drivers are not equally distributed. This equation reduces excessive task assignments to specified drivers. The second term expressed in Equation (21) evaluates the result's feasibility. This equation eliminates solutions that are not feasible within the current transport environment.

$$I(Z) = \sum_{x \in Z} (d_{ad}(x) + O_{at}(x)), \tag{19}$$

  where $d_{ad}(x)$ is the equality state of the task distribution, and $O_{at}(x)$ is the operation loss caused by the delivery delay or cancellation in all vehicles within period x.
- The task distribution function evaluates the fairness ($d_{ad}(x)$) of distributing tasks to drivers, as indicated in Equation (20):

$$d_{ad}(x) = \sum_{v \in x} \left(\frac{d_{at}(v)}{size(delivery)}\right) \times 100, \tag{20}$$

  where d_{at} is the number of vehicles, v, used to meet customer demand, and its magnitude represents the overall deliveries flowing into the system.
- The feasibility evaluation function is used to determine the operational feasibility of the solution suggested by RL. The ($O_{at}(x)$) function is assessed based on the delay time and the number of incidents during period x, as indicated in Equation (21):

$$O_{at}(x) = \sum_{v \in x} \Theta(O(v) - \hat{d_{at}}(v))^{(\Theta(O(v) - \hat{d_{at}}(v)))}, \tag{21}$$

  where $O(v)$ is the expected delivery carried by vehicle v, and $\hat{d_{at}}(v)$ denotes the total delays or cancellations of deliveries in vehicle v. Θ is a function that returns zero if the delivery of vehicle v is delayed or unsuccessful. A large ($o_{at}(x)$) value indicates that the suggested solution is feasible for employment. Note that the x in the equations expresses the period for delivering goods to customers.

The reward for each environmental state is then input and concatenated with the other essential data into the tree-based regression model of the transport environment (e.g., $[S, A, G(Z)], \ldots, [S_n, A_n, G(Z)_n]$). The methodology of the tree-based regression is explained in the following section.

C. TREE-BASED REGRESSION METHOD FOR MODELING THE TRANSPORT ENVIRONMENT

The transport environment state (S) and its reward obtained from the RL agent action discussed in the previous section is modeled to train the RL for additional vehicle route optimization tasks, and the environment state and associated reward are used as inputs to the tree-based regression to train and test with grid-search parameter tuning to obtain the optimal model.

Generally, a regression tree assigns a prediction value to each leaf node, where the prediction value is the reward obtained from a given set of transport environment states. The regression tree model is defined in Equation (22):

$$kernel_d(i^l, i) = \frac{I(i^l, i)}{\sum_{a,b \in d} I(a, i)}, \tag{22}$$

where $I(i^l, i)$ is an indicator function that determines whether i^l and i belong to the same class. Furthermore, to avoid over-fitting, the ensemble method (e.g., random forest and bagging) is applied. The modification of Equation (22) is made accordingly as in Equation (23):

$$kernel_d(i^l, i) = \frac{1}{P} \sum_{m=1}^{P} \frac{I^m(i^l, i)}{\sum_{a,b \in d_m} I^m(a, i)}, \tag{23}$$

where d_m denotes a subset of the training data used to construct m regression trees (mth). $i^m(i^l, i)$ is an indicator function that indicates whether i^l and i belong to the same class of the mth tree. If the method constructs an ensemble of P different regression trees, the average of the P predicted values is used as the final prediction. More details of the tree-based regression model construction can be found in [28]. The models used in this study include the extra tree, random forest, bagging, and decision tree. The model having the highest accuracy in predicting the environment state is adopted to create the environment in the final step.

When using the tree-based regression model to predict the environment state represented by reward, a flag is set when the unit is used to predict the reward; otherwise, new reward functions are used to evaluate the provided utilities represented by the environment elements. Thus, the reward algorithm to model the transport environment is modified as indicated in Algorithm 1.

When the optimal tree-based regression model is accepted and constructed, the reward analyzed from the current transport environment is transferred to the RL agent using the actor–critic gradient. The definition of this gradient is discussed in the next section. This process informs the RL agent about its actions and changes made to the transport environment when performing vehicle route optimization. A positive reward means the current environment is normal, and the RL agent's actions are feasible. Therefore, the actor–critic gradient is also positive. Otherwise, if the reward is negative (e.g., in the range of $[-\infty, 27.71]$), it means that the current environment is in a critical state, and adjustments are required. Thus, the actor–critic gradient is also negative.

Algorithm 1 Modified Reward Function Used to Model the Transport Environment With the Tree-Based Regression Model

Input : input tour $sample_{solution}$,
$sample_{solution_{titled}} \leftarrow Stack(sample_{solution})$
feature set $F_1, F_2, \ldots F_n$,
$flag \leftarrow true$,
$vehicle_{used} \leftarrow count(sample_{solution_{titled}})$
Output: reward

for $n = 1, 2, \ldots$ **do**
  if $flag$ is true **then**
    reward $\leftarrow model_{tree}\,(\,action, current_{state}, next_{state}\,)$
    return reward
  else
    $d^n \leftarrow cal_{dist}\,(\,sample_{solution}, sample_{solution_{titled}}\,)$
    $travel^n \leftarrow d^n / model_{traff}\,(\,F_1, F_2, \ldots F_n\,)$
    $behavior_{stat} \leftarrow$
    $model_{behav}\,(\,Vehicle_{used}, F_1, F_2, \ldots F_n\,)$
    $eval_{feasibility} \leftarrow \sum_{i=0}^{n}(\,\beta \times eval_{util}\,(work)\,) +$
      $\hookrightarrow\ (\,\delta \times eval_{util}\,(delay)\,) +$
      $\hookrightarrow\ (\,\gamma \times eval_{util}\,(operation, fairness)\,)$
    if $behavior_{stat}$ is true **then**
      reward $= -(d^n + waiting_{time} + eval_{feasibility})$
      return reward
    else
      reward $= d^n + eval_{feasibility}$
      return reward
    end if
  end if
end for

When the transport environment model is completed, and its associating reward function is defined, an optimization model using RL can then be constructed. The details of the procedures are given in the next section.

D. PROPOSED OPTIMIZATION MODEL BY RL

This section outlines the procedures of constructing the proposed vehicle route optimization model, which can adapt to any environmental changes using RL. The set of inputs containing essential data, such as customer coordinates and demands, is denoted as $X = \{x^i\}, i = \{1, \ldots, M\}$. The elements in X are updated as the RL agent selects the customer for making a delivery. A demand (d) is delivered to the customer at time t and location in state s. This step is the decoding stage, wherein the network node that encodes customer data are decoded to construct a vehicle route sequence.

To encode the customer data into the network node, each input of X is represented as x^i and is denoted by a sequence of nodes expressed as $x^i \doteq (s^i, d^i)$, where $i = 0, 1, \ldots, n$. This concept is adopted from [23] and represents how the network node encodes customer data.

The results from the data encoding process are represented as Network node 1 = [1, 100.934, 13.535, 15], Network node 2 = [2, 100.4534, 13.635, 30], Network node 3 = [3, 100.334, 13.565, 2], . . . , and Network node N = [N, longitude, latitude, demand]. The data are decoded to create the sequence of nodes as Customer 1, Customer 3, Customer 2, until Customer N. This step continues until all nodes are decoded, meaning that all customer demands are satisfied. The details of the decoding steps are discussed in a later section.

When the customer data are encoded, a mechanism for solving SDVRP is constructed. This section highlights how the proposed model can solve SDVRP, as discussed in Section III. The masking scheme introduced by [23] is applied to label the nodes to force the model to support the SDVRP as follows:

1) Customer nodes without demand are not visited.
2) Customer nodes with demand higher than the vehicle capacity are masked.
3) All customer nodes are masked if the remaining vehicle capacity is zero.

These conditions apply to the classical VRP constraints. However, condition (2) should be relaxed to allow a vehicle to visit a customer more than once in SDVRP. All customer demands are satisfied under the masking scheme conditions when the traveling plan is set. A solution is chosen if a particular condition is satisfied. In [23], the authors set the log probabilities of infeasible solutions to $-\infty$. Furthermore, the transport environment model presented in Section IV-C is connected to this masking scheme. Therefore, the RL can be reinforced to solve SDVRP and adapt according to the transport environment. This mechanism is a crucial difference between our proposed model and the framework proposed by Nazari *et al.*

The optimization task is initialized from the input, X_0, with the pointer denoted as y_0 as a reference. At each decoding step, $t(t = 0, 1, \ldots, n)$ y_{t+1} are the pointers for the remaining inputs, X_t. According to the policy, π, the most adjacent node is selected as the next decoder step.

Policy π is determined by the policy gradient method, which comprises the actor–critic networks. The actor network is used to predict the probability of subsequent actions to be performed at a given decision step. The critic network is used to calculate the value of the actions performed in the current state. It is generated from sequences of Y to minimize the loss of the objective function.

The optimal policy denoted as π^* generates the optimal value with a probability of one. Therefore, the value of π and π^* must be as close as possible to determine the optimal solution. We use the probability chain rule to determine the probability of generating sequence Y from the given next sequence, y_{t+1}, to decode node X_t per Equation (24):

$$\Pr(Y|X_0) = \prod_{t=0}^{T} \Pr(y_{t+1}, X_t). \tag{24}$$

The state is repeatedly updated with the state transition function denoted as f, as indicated in Equations (25)–(26):

$$X_{t+1} = f(y_{t+1}, X_t), \tag{25}$$

$$\Pr(Y_{t+1}|Y_t, X_t) = softmax(g(h_t, X_t)), \tag{26}$$

where g is the function that calculates the distance between the input vectors, and h_t is the state of the RNN. The outcomes are the probability that the preceding decoded step will transition to the next decoder node and environment state obtained by the *softmax* function.

When the maximum iteration is reached, the output is the routing sequence that assigns each vehicle in the fleet to fulfill customer demands. The minimum traversal cost associated with the distance, time, and environmental elements are returned. This output is then recommended to the logistics agency for creating their delivery plan.

The mechanism used to obtain the optimal solution is demonstrated in this section. It applicable for general VRP, which contains a graph consisting of nodes and edges. However, further modifications are required when applying the ML and RL to construct those kinds of graphs. The following section outlines the method of modeling RNN as a graphical network.

1) PROPOSED NETWORK FOR DEALING WITH SEQUENCE OF INPUTS

Generally, neural-network (NN) neurons do not have a function that reveals and connects adjacent neurons. Therefore, using NN to model a vehicle route requires a mechanism to connect each network's node as a graph.

When applying NN to construct this network, the input described previously is embedded instead of using the RNN hidden states to handle the data. By doing so, the network node represents the customer location and the demand information. Instead of the RNN hidden state, the embedded inputs are used to avoid difficulties inputting the demand directly to the network node. Therefore, the RNN model is divided into two components. The first is a set of embedding processes that connect the input attributes and project them onto a D-dimensional vector space. The second component is the decoder layer that points to an input, which is the next visited node at every decoding step.

The RNN is used to model the decoder network to determine the next visiting node from the sequence of remaining encoded nodes. When the next vehicle visiting node is identified, the update is made to the delivery because the number of deliveries changes over time as goods are delivered. When the customer location and demand updates are required, the model must be re-executed because the information cannot be updated dynamically on the network node itself. Therefore,

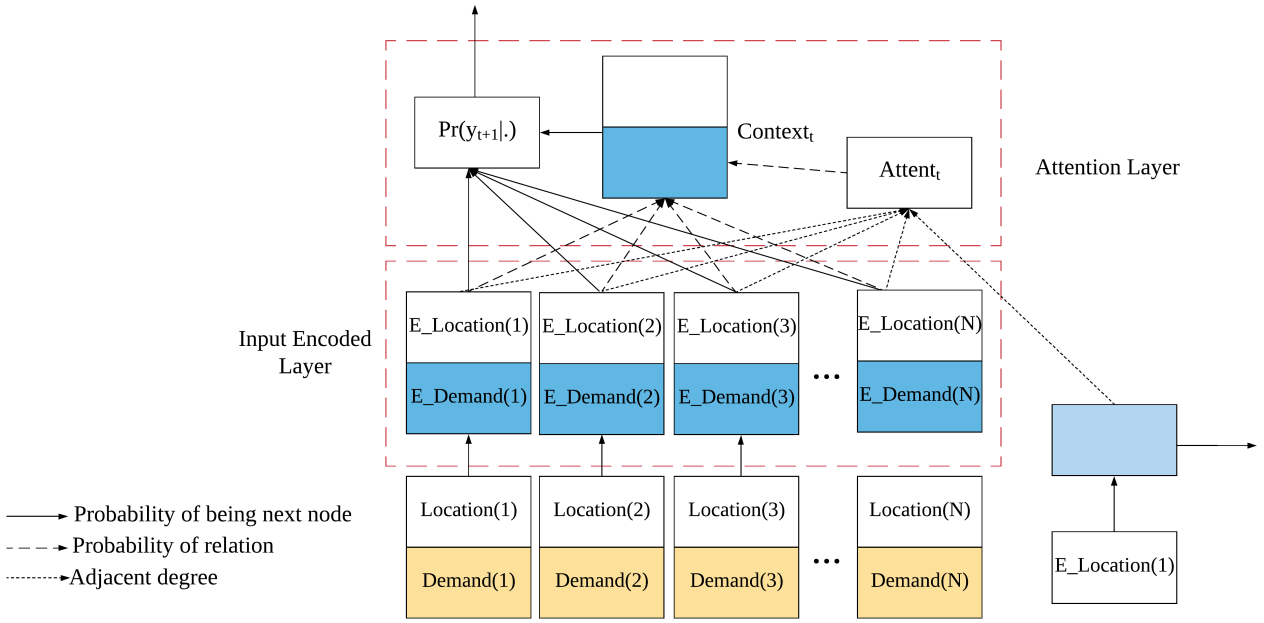


FIGURE 4. Demonstration of the model architecture for the vehicle routing problem.

we adopt the attention mechanism of [21], [23] to assist the network node as it references and dynamically updates the delivery information.

This attention mechanism is used to store the demand information of each network node and connect independent network nodes as a graph. Therefore, every time the vehicle delivers to a customer location (e.g., [latitude, longitude]), the remaining demand information is updated on the ''attention layer'' instead of the network. Therefore, model rebuilding is no longer required, and the demand information can be updated dynamically. A framework of the described processes is illustrated in Figure 4.

The attention mechanism references the input at the decoder step, i. It is used to calculate the probability that the adjacent node is the subsequent node of the current network node. The encoded input is denoted as $\bar{x}_t^i = (\bar{s}^i, \bar{d}_t^i)$. t and $h_t \in \Re^D$ represent the state memory of the RNN at decoding step, t. For example, x_0 indicates the beginning of the tour, and the attention mechanism is used to find nodes to be visited by vehicles. The suitable subsequent node of the current decoded node is determined by the context value. If required, the process also updates the encoded values (e.g., update the coordinates, current demand, or both).

The attention layer is denoted by $attent_t$, as defined in Equation (27):

$$attent_t = attent_t(\bar{x}_t^i, h_t) = softmax(u_t), \tag{27}$$

where $u_t^i = v_a^T \tanh(W_a[\bar{x}_t^i; h_t])$ is the compatibility between two adjacent nodes, and M is the set of nodes in the Y sequence. v_a and W_a are the training variables. When the attention value is obtained, the node context ($context_t$) and the possible nodes for the next visit are determined by Equation (28). The high value of $context_t$ indicates that the nodes are adjacent.

$$context_t = \sum_{i=1}^{M} attent_t^i \bar{x}_t^i. \tag{28}$$

Using the encoded input, the values are normalized using the *softmax* function. $Pr(y_{t+1})$ denotes the next sequence in Y given the current sequence, Y_t and decoded node X_t, as expressed in Equation (29):

$$\Pr(y_{t+1}|Y_t, X_t) = softmax(\tilde{u}_t^i), \tag{29}$$

where $\tilde{u}_t^i = v_c^T \tanh(W_c[\bar{x}_t^i; context_t])$ is the compatibility of two adjacent nodes. v_c and W_c are the training variables.

A single long short-term memory (LSTM) layer for a decoder size 128 was used to support decoded tasks. From tuning parameters and evaluating solution accuracy, 128 is the most suitable vector size to store all data without losing any information for the optimization solution. However, the network's parameters must be adjusted when applying to other application domains with different types of input data.

The processes described above show how each network node is connected as a graphical network. The remaining process is the customers' demand updated after each delivery. The demand (dm_i^t) is updated at step time t as the current demand, dm_i^t, minus new deliveries of demand $new_{del_t}^i$, and is linked back to the network node, i, reference at the attention layer.

After the actor–critic networks are constructed, the network can be trained using the two algorithms for training

RL agent: actor–critic and asynchronous actor–critic (A3C) algorithms. Detailed definitions of these two algorithms can be found in [3]. In this study, these algorithms were trained and tested with 120,000 steps and 100,000 steps, respectively. The channel for reinforcing and adjusting RL behavior is defined by the actor–critic gradient. The actor gradient is defined by Equation (30), and the critic gradient is defined by Equation (31).

$$d\theta \leftarrow \frac{1}{N}\sum_{n=1}^{N}(R^n - V(X_0^n;\phi))\bigtriangledown_{\theta}\log \Pr(Y^n|X_0^n), \tag{30}$$

where $d\theta$ denotes the actor gradient, and R^n denotes the reward from the environment state, n. The reward, R^n, is obtained from the ''Tree-based Regression Method for Modeling the Transport Environment'' section.

$$d\phi \leftarrow \frac{1}{N}\sum_{n=1}^{N}\bigtriangledown_{\phi}(R^n - V(X_0^n;\phi))^2, \tag{31}$$

where $d\phi$ denotes the critic gradient, and R denotes the reward from the environment state, n. If the environment state created by RL is normal, the gradient is set to positive. Otherwise, it is set to negative. Figure 5 illustrates how the reward is applied to update and reinforce the RL agent to perform appropriate actions.

Additionally, the optimal hyperparameters are set as follows. The Adam optimizer has a learning rate of $10^{-}4$, and a dropout with a probability of 0.1 in the LSTM decoder. The model is trained on a Google Colab GPU.

In summary, the model in Figure 4 originates from the customer coordinates with encoded demand presented in the ''Input Encoded Layer.'' The encoded node is then used to compute the adjacency degree (attention value) in the ''attention layer'' as in Equation (27). Next, the attention values are computed to determine whether each node is related and adjacent (Equation (28)). Then, the probability of choosing and decoding the current node in the routing sequence is computed using Equation (29). Last, the demand at each node is updated according to the new deliveries made by vehicles. This process repeats until the maximum iteration is reached.

Following the formulations of all necessary parameters, these equations are converted to the programmable computer algorithm shown in Algorithm 2. Afterward, the vehicle route optimization task can be performed. Then, the significance of experimental results can be evaluated using the given evaluation metrics.

Algorithm 2 Reinforcement Learning With Model-Based Integration Algorithm

Input : random weight θ, θ^n, ϕ, ϕ^n
Output: Route sequence (Y^n) for vehicle V_n
initialize the actor network with random weight θ and critic network
    ↪ with random weight ϕ
initialize N thread-specific actor and critic networks with weights θ^n and ϕ^n associated with thread n.
for each thread n **do**
  for *iteration* $= 1, 2, \ldots$ **do**
    reset gradient: $d\theta \leftarrow 0$, $d\phi \leftarrow 0$
    sample N instance according to Φ_M
    for $n = 1, \ldots, N$ **do**
      initialize step counter $t \leftarrow 0$ and select vehicle v_n
      repeat
        select y_{t+1}^n refer to $\Pr(y_{t+1}^n \mid Y_t^n, X_t^n)$
        $X_t^n \leftarrow X_{t+1}^n$
        $t \leftarrow t + 1$
      until terminated condition is matched
      reward $R^n \leftarrow model_{behav}(Y^n, X_0^n)$
      % the model-based in section IV-C.
    end for
    $d\theta \leftarrow \frac{1}{N}\sum_{n=1}^{N}(R^n - V(X_0^n;\phi))$
      ↪ $\bigtriangledown_{\theta}\log \Pr(Y^n|X_0^n)$
    $d\phi \leftarrow \frac{1}{N}\sum_{n=1}^{N}\bigtriangledown_{\phi}(R^n - V(X_0^n;\phi))^2$
    update θ, ϕ
  end for
  return Y^n
end for

E. PERFORMANCE EVALUATION METRICS

This section outlines the evaluation metrics used for the tree-based regression models and the RL. The tree-based regression models were evaluated using the root mean-square error (RMSE), mean-square error (MSE), and mean absolute error (MAE). Whereas the RL was evaluated by the cumulative reward and the agent action value in each training step. These metrics were derived from the previous studies by [39]–[41]. The larger value of these measures indicates an outstanding RL model. The cumulative action value (CAV) is expressed as Equation (32):

$$CAV = \sum_{i=0}^{N} d\phi, \tag{32}$$

where i denotes the training step, N denotes the total training steps, and $d\phi$ denotes the RL agent's action value defined in Equation (31).

In addition to the CAV, manually-made and state-of-the-art model solutions were used as a baselines to determine the improvements when performing the same optimization tasks against the RL agent. Such improvements were referred to as the ''optimal gap'' in the previous studies, as defined in Equation (33):

$$Optimal_{Gap}(\%) = \frac{\sum_{i=1}^{N} Baseline_i/N - current_{solution}}{\sum_{i=1}^{N} Baseline_i/N} \times 100 \tag{33}$$

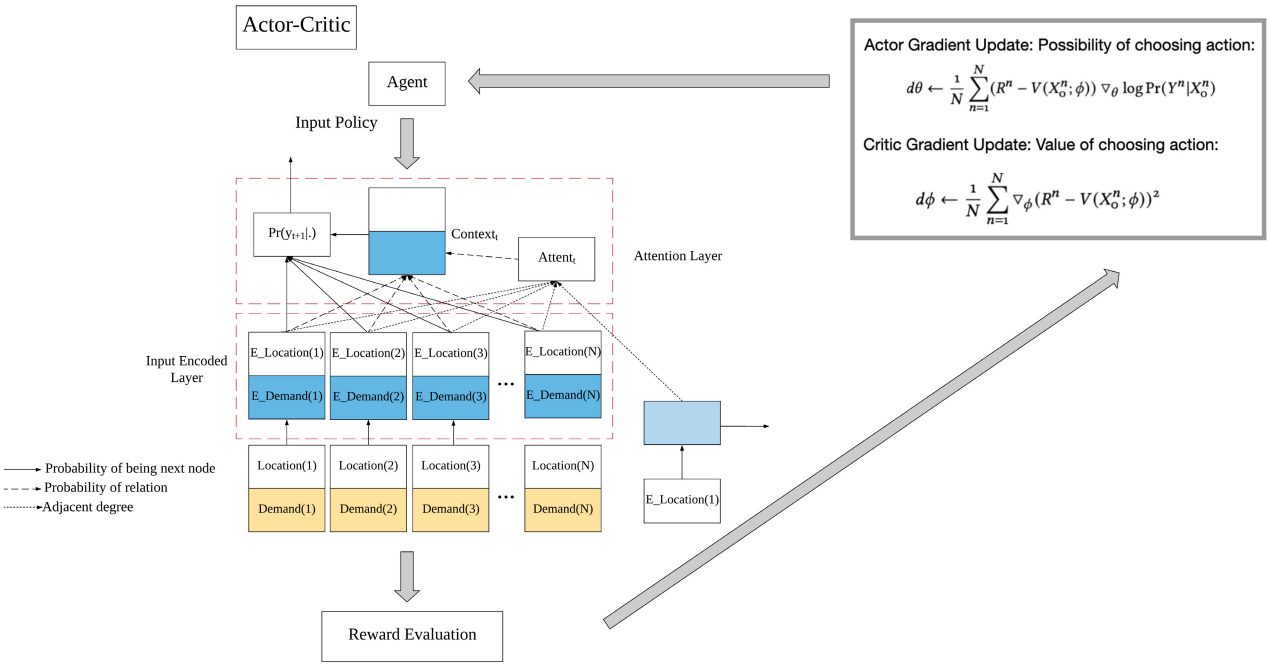


FIGURE 5. Demonstration of the actor–critic algorithm architecture.

TABLE 5. The Z-value for the confidence interval.

Confidence Interval	z
80%	1.282
85%	1.440
90%	1.645
95%	1.960
99%	2.576
99.5%	2.807
99.9%	3.291

where i denotes a baseline index, $Baseline_i$ denotes a result from the state-of-the-art models, and $current_{solution}$ denotes the result of the proposed model.

The RL model was evaluated using the agent action value defined in Equation (32). The optimization results were compared with state-of-the-art models using Equation (33). Herein, the confidence interval (CI) is also employed to determine the boundary of the optimization result. The CI and Z-values (z) are defined in Equation (34), which are referred from [42], [43].

$$CI = \bar{X} \pm z\frac{sd}{\sqrt{n}}, \tag{34}$$

where $\bar{X}$ denotes the mean of the optimization result, z denotes the 95% of CI value chosen from Table 5, sd denotes the standard deviation, and n is the number of test samples.

Finally, the vehicle route optimization tasks are performed using the proposed methodologies. The optimization results are compared and validated in a two-stage manner. The first stage is when the SDVRP instances are used as inputs to demonstrate the generality of the proposed model in solving general academic problems. Therefore, the VRP50 instance taken from Nazari *et al.*'s paper was used for model validation. The significant difference from their experiment is that the coordinate system (e.g., European Petroleum Survey Group (EPSG) 4326 for coordinate projection) was modified to reflect real-world scenarios better. The second stage is when the actual operation data are used as inputs and evaluated through case studies. Their results were also validated with case studies described in the following section.

V. CASE STUDIES

In this study, several case studies are applied to demonstrate the effectiveness of the proposed methodology in different settings, both under static and uncertain transport environments. This section outlines two types of case studies (with and without uncertainties) used for model validation. The comparison between the two types of case studies determines how well the proposed model performs under different transport situations faced by the vehicle fleet during its daily operation.

In the case studies, each customer demand may contain multiple containers, which one vehicle cannot fulfill. Therefore, the demand must be split and handled by the same, if available, or other vehicles until customers' demands are fulfilled (e.g., quantity and delivery time). Thus, the SDVRP has to be employed in the case studies of the task assignments of container trucks making deliveries across Thailand. An example of the task assignment is presented in Figure 6.

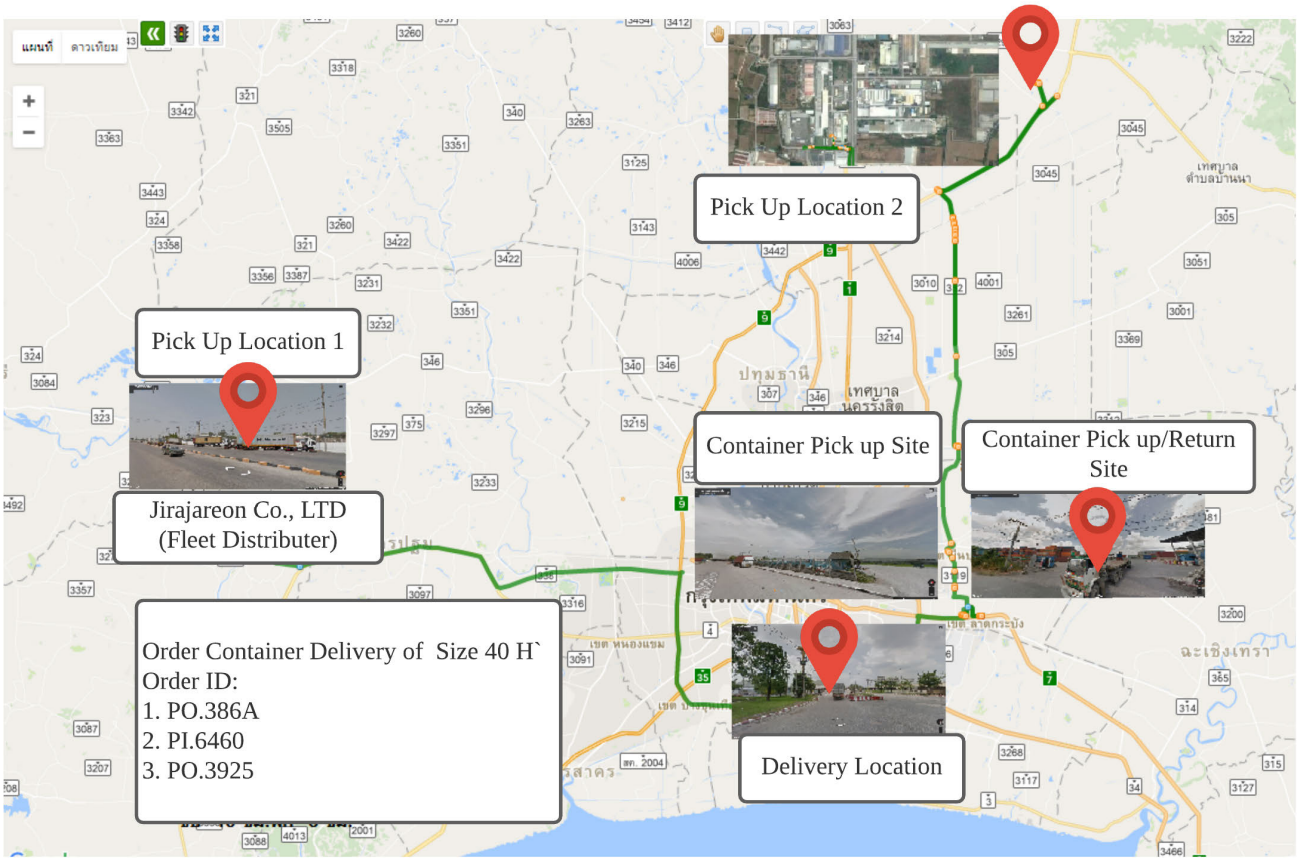


FIGURE 6. Demonstration of a routine route that is assigned to each vehicle in the fleet.

For model validation, two types of case studies were employed. The first represents the normal state transport environment. The second case study represents the critical state transport environment containing disturbances that may affect the delivery tasks. More details of these two case studies are explained in the following subsections.

A. CASE STUDY WITHOUT UNCERTAINTY

The trained RL was implemented to optimize vehicle routes under the case study where no sudden change occurs during the daily operation (e.g., no order postponement, cancellation, vehicle shortages, or human resources issues). The detailed testing dataset used for the validation is as follows. Generally, the logistics agency's fleet that handles container deliveries has routines for medium-to-long hauls with an amount of 21 container deliveries per day, each with the capacity of 2,700 cubic feet (cu.ft.). The delivery amount includes the actual deliveries of the company, as shown in Figure 2. There are 37 available vehicles, and the size of the test dataset was larger than 1 GB.

The RL model was tested with longitude, latitude, and customer demand data. The data was also formatted similarly to the logistics agency when it performed the transportation plan. The RL model was executed to optimize the vehicle routing following the methodology presented in Section IV.

After the RL model reaches the maximum iteration of 100,000, a feasible vehicle route solution was returned. The solution was assessed using the evaluation metrics presented in Section IV-E. The result obtained from the RL model was also compared to the actual company routing plan and state-of-the-art models in terms of profit improvement and computational time.

The optimization result from the proposed model was first compared with the RL models developed by [21], [23] to demonstrate the advantage of hybridizing the model-free and model-based RL principles against the only model-free RL model. Second, the proposed model is compared with the optimization models using MILP approaches on the models from [1], [15], [44] to demonstrate how significant the environmental uncertainties are. Lastly, the proposed model was compared with the manually made routing plan conducted by the logistics company to demonstrate how the model can improve the solution using transport-environment information obtained multiple sources. In the following subsection, the more complex case study under uncertainties is presented.

B. CASE STUDY WITH UNCERTAINTIES

In this case study, the environment is uncertain with regards to incidents and disturbances. This type of circumstance may arise when demand exceeds the fleet capacity. Therefore, the

TABLE 6. Investigation results of the tree-based regression model for predicting the transport environment states.

Model	RMSE	MSE	MAE
Extra Tree	335.049	112,257.900	43.044
Random Forest	6.509	42.367	0.604
Bagging Tree	6.160	37.951	0.605
Decision Tree	7.830	61.310	0.788

container deliveries were adjusted to 60 for testing data, and the environment was represented as critical where numerous incidents befell.

Similar to the first case study, the test data here were employed to optimize the vehicle routing tasks following the methodology presented in Section IV under the same evaluation procedures. The experimental results from these cases are explained in the next section.

VI. RESULTS

This section describes the experimental results from the construction and execution of the RL vehicle route optimization. It also includes the details of an experiment that searched for the optimal tree-based regression for modeling the transport environment. In the following subsections, the result from the tree-based regression model is first discussed, followed by the results from SDVRP instances, and case studies without and with uncertainties, respectively.

A. RESULTS OF TREE-BASED REGRESSION MODEL

The experimental results of the procedures in Section IV-C are displayed in Table 6. The results highlight the effectiveness of ML for predicting the environmental state represented as the reward for the RL agent's interactions.

Referring to Table 6, the lowest of these evaluation metrics values denotes that the transportation environment for logistics was accurately defined and modeled. When these methods' experimental results are presented, the optimization result from this experiment's regression model is presented.

B. EXPERIMENT RESULTS UNDER SDVRP INSTANCES

The proposed optimization model was validated using the SDVRP instances shown in [23], and the experimental results are shown in Table 7. There, the proposed model is denoted in bold. Additionally, "+" denotes an improvement from the baseline, and "−" signifies no improvement. *Baseline* denotes the target traversal cost incurred from the transportation plan with the given requirements.

C. EXPERIMENTAL RESULTS FROM THE CASE STUDY WITHOUT UNCERTAINTY

The experimental results from the RL vehicle route optimization in a static environment are presented in Table 8. The "$Result_{mean} \pm CI$" column represents the optimal traversal cost from a given set of customers in the testing dataset represented in THB. The *CI* is computed to assess the reliability of the model. A lower *CI* implies a stable model. The SD column represents the standard deviation of the optimization result, where a lower SD indicates a stable model. The third and fourth columns showed the computational time required for model training and testing. Finally, the fifth and sixth columns exhibit the efficiency of the optimization model. The "$Optimal_{Gap}$" column shows the differences between the optimal solution and the baseline as a percentage. In the analysis column, "+" denotes positive increase, and "−" denotes negative decrease. Lastly, the last row of Table 8 outlines the results of the proposed model.

Additionally, the *CAV* values of the RL are illustrated in Figure 7. The higher *CAV* value indicates that the RL performs vehicle route optimization tasks more appropriately.

D. EXPERIMENTAL RESULTS FROM THE CASE STUDY WITH TRANSPORT ENVIRONMENT UNCERTAINTIES

This case study illustrates situations wherein the RL handles uncertain changes occurring during the vehicle route optimization task. The experimental results are displayed in Table 9. The same procedure was used to analyze the results of Table 8 as that for Table 9. The result obtained from the proposed model is indicated in bold. Additionally, the *CAV* of RL for this case study is presented in Figure 8.

VII. DISCUSSION

This section discusses the experimental results under different case-study settings and the tree-based regression model for predicting the transport environment state displayed in the previous section.

A. RESULTS OF THE TREE-BASED REGRESSION MODEL

The results displayed in Table 6 indicate that the bagging tree has the highest performance when predicting the transport environment state from the given RL agent action (6.160 of RMSE), followed by the random forest (6.509 of RMSE) and decision tree (7.830 of RMSE). These methods are ensemble algorithms that combine predictions from multiple models. Therefore, they operate more effectively if the predictions from each model are uncorrelated. The least efficient predictor is the extra tree having a 335.049 RMSE.

The significant difference between random forest and bagging trees is that they consider different features when using the split operator for diving nodes. The random forest only considers a subset of features that provide the best performance for splitting among the overall features. In contrast, the bagging tree considers all features for splitting nodes regardless of their performance. Figure 3 shows that the features shown in this study have both dependency and non-dependency relationships. Discarding some features may reduce the model's accuracy. Therefore, the result from the bagging tree has higher accuracy than the random forest.

In the experiment, a five-fold grid search was used to discover optimal model parameters to avoid overfitting. Accordingly, each model was constructed based on the optimal setting. The number of estimators was 50, the minimum

TABLE 7. Result of vehicle route optimization with VRP50 instance and capacity of 40.

Model	$Result_{mean} \pm CI$	SD	$Compute_{Time}$	$Optimal_{Gap}$	Analysis
Nazari et al.	2,839.60±1.24	20.972	0.550 s	0.005%	+
Kool et al.	2,839.53±1.25	21.11	0.525 s	0.007%	+
Musolino et al.	2,839.83±0.00	0.00	0.045 s	0.002%	+
OR-Tools	2,840.03±0.00	0.00	0.040 s	0.009%	+
Baseline	2,839.75±0.00	-	-	-	–
Proposed model(A3C-tree)	2,843.095±0.28	4.793	0.730 s	0.118%	+

TABLE 8. Summary of results under case studies without uncertainty using RL with the tree-based regression method.

Model	$Result_{mean} \pm CI$	SD	$Training_{Time}$	$Prediction_{Time}$	$Optimal_{Gap}$	Analysis
Nazari et al.	15,616.80±1.75	28.19	6.31 h	289 s	24.10%	–
Kool et al.	15,616.80±1.75	28.23	5.25 h	288 s	24.10%	–
Giovanni et al.	24,100.96±0.00	0.00	7 h	-	14.63%	+
Musolino et al.	27,246.74±0.00	0.00	6.35 h	-	24.48%	+
OR-Tools	20,156.80±0.00	0.00	7 h	-	2.04%	+
Company	27,387.15±0.00	0.00	-	-	24.87%	+
Average Baselines	20,575.70±0.00	0.00	-	-	-	-
Proposed model(A3C-tree)	16,494.89±0.06	0.98	6.45 h	391 s	19.83%	–

TABLE 9. Investigation results of a case study when uncertain changes did occur during the vehicle route optimization using reinforcement learning with the tree-based regression method.

Model	$Result_{mean} \pm CI$	SD	$Training_{Time}$	$Prediction_{Time}$	$Optimal_{Gap}$	Analysis
Nazari et al.	15,616.80±1.75	28.19	6.31 h	289 s	24.10%	–
Kool et al.	15,616.80±1.75	28.23	5.25 h	288 s	24.10%	–
Giovanni et al.	24,100.96±0.00	0.00	7 h	-	14.63%	+
Musolino et al.	27,246.74±0.00	0.00	6.35 h	-	24.48%	+
OR-Tools	20,156.80±0.00	0.00	7 h	-	2.04%	+
Company	27,387.15±0.00	0.00	-	-	24.87%	+
Average Baselines	20,575.70±0.00	0.00	-	-	-	-
Proposed model(A3C-tree)	32,990.26±0.08	1.22	6.45 h	391 s	37.63%	+

leaf sample was 20, and the minimum sample for the split was 20 for the extra tree. For the random forest, the number of estimators was 50, and the max depth was 100. Additionally, the number of estimators was 50, the random state was two for the bagging tree, and for the decision tree, the number of estimators was 50. The max depth was 100.

The tree-based method was selected over other ML models, because they require hyperparameter tuning, causing a lack of flexibility and long computational times. Moreover, the regression was selected over the classification problems to determine the environmental state value instead of using its class. Therefore, the bagging tree was used as the predictor in the hybrid model to model the transport environment.

In the next section, the experiment results of using the RL for performing the vehicle route optimization are presented.

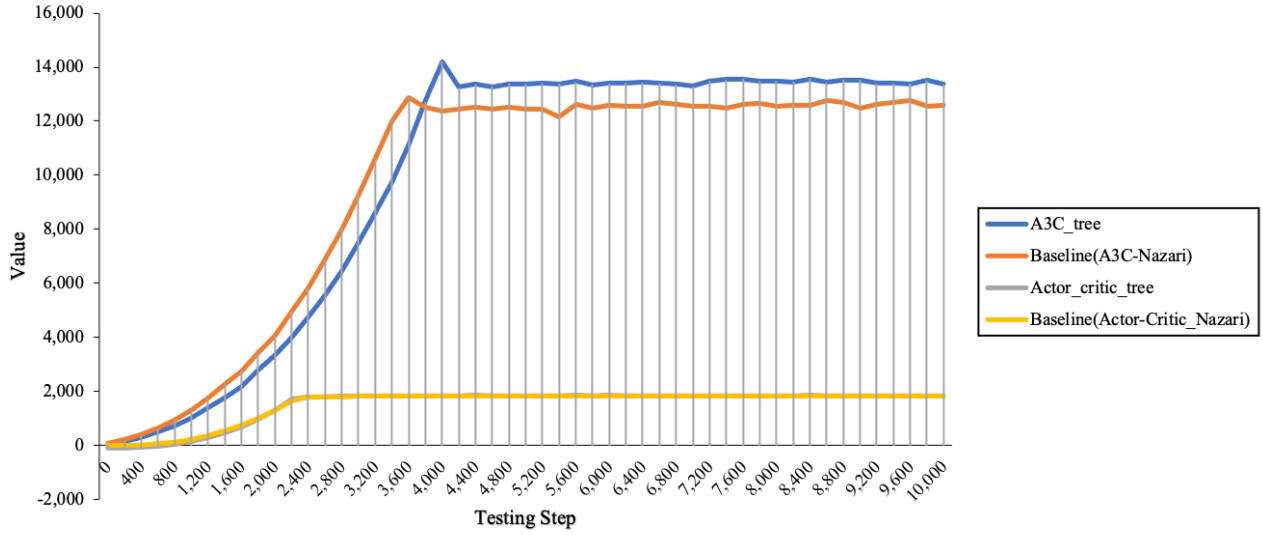
B. EXPERIMENTAL RESULTS UNDER SDVRP INSTANCES

This section highlights the generality of the proposed model in solving the SDVRP problem with 50 instances. VRP50 was chosen for this study, owing to its similarity to the actual transport environment. Table 7 shows the trained model obtained from the proposed methodology, the approaches from the state-of-the-art models, and previous-work near-optimal results with an up-to 0.118% optimality gap. It also appears that the computational time was not significantly different among models.

These findings demonstrate the generality of the proposed methodology for solving the general SDVRP with minor modifications. The required modifications include the inputs fed into the reward processing unit for RL, the tree-based regression method, and the model's hyperparameter tuning, including learning rate, epochs, and batch size.

Only the demand and customer coordinates derived from the dataset were used to create the transportation environment in this experiment. The state of the transport environment stored in the tree-based regression method was applied to analyze whether the environment was normal. However, this environment generation did not affect the efficiency of the optimization.

(a) Training Stage



(b) Testing Stage

FIGURE 7. Training (a) and testing (b) stages regarding the situation which no uncertain changes from environment occur using reinforcement learning with the tree-based regression model.

In the next section, the application of the trained model to the variants of actual case studies is demonstrated.

C. EXPERIMENTAL RESULTS FROM THE CASE STUDY WITHOUT UNCERTAINTY

Figure 7 indicates that the convergence of *CAV* of the RL started from step 2,400 for the actor model and step 4,200 for the A3C algorithm. In each training and testing step, these *CAV* were calculated using Equation (32). Based on Figure 7, it appears that the agent trained by the A3C algorithm yielded the highest *CAV* for the performed tasks, because the A3C algorithm had more opportunities to explore the transport environment than the conventional actor–critic algorithm. Therefore, the algorithm was more adaptable according to the transport environment than the actor-critic algorithm.

CAV signifies the efficiency of the RL agent in optimizing vehicle routing. This value is influenced by the reward or penalty received by the RL agent. A higher *CAV* indicates that the RL agent returns appropriate actions without violating the environmental condition. Thus, the solution is close to optimal.

When comparing the suggested profit obtained from the optimization model shown in Table 8, the proposed model's result shows profit improvement by approximately 5.32% compared with that of [21], [23]. This finding suggests that using prior experience (e.g., the previous outcome from the agent's action interacts with the environment) instead of the trial-and-error strategy to tune and reward the RL agent is vital. The reward principle used in this paper is more effective than the general agent rewards, including positive or negative route values with a trial-and-error strategy by the model-free

(a) Training Stage

(b) Testing Stage

FIGURE 8. Training (a) and testing (b) stage regarding to the situation which uncertain changes from environment occur using reinforcement learning with the tree-based regression model.

RL principle. However, the model-free RL excludes the external environment; hence, the model is not fully adaptive to changes. Thus, the proposed model, which combines the benefits of model-based RL and model-free RL, is applicable to real-world settings.

Supposing the transportation environment differs from the previous known environment, it might detract from the optimization model. The environment from the previous day might not fully represent the transportation environment in the current day. This finding is demonstrated through the result in Table 8. The result shows that the suggested solution of the proposed model is reduced when compared to the average baselines. It denotes that the environment used in this case study has not been fully modeled and stored in the tree-based regression model. This shortcoming is caused by the limitations of creating and storing the transport environment state using tree-based regression. When events in the environment do not constantly occur in a similar pattern, the stored experiences can only represent events that do not fully portray the current environment. Therefore, it may cause difficulty for the RL agent to adapt its strategy to this event change. Under uncertain events, such as variations of demand, vehicle utility, and productivity, using prior environment states as experience for training the RL agent is efficient to some extent. However, there is room for improvement when dealing with such a complex transport environments.

These limitations can be compensated by considering real-time information so that the obtained solutions are adaptable to changes during the current time frame. It is thus necessary to enable the RL agent to take the real-time

information from the environment while applying the experiences obtained from prior environment states to solve the VRP. This additional improvement can be pursued in future works. Therefore, the discussion of results obtained under this case study with uncertainties from the transport environment is discussed in the following section.

D. EXPERIMENTAL RESULTS FROM THE CASE STUDY WITH TRANSPORT ENVIRONMENT UNCERTAINTIES

Similar to the first case study, the results obtained from Section V-B indicate a profit improvement of approximately 52.66% compared with [23] and [21], approximately 26.67% compared with [15], approximately 38.90% compared with [44], and 17.141% compared with [1].

The reason is that these baseline models were formulated to minimize the traversal costs while assuming that it is feasible to visit all customers. When the uncertainties from the transport environment are included, these models cannot compensate for these changes. Thus, the profit is decreased, owing to the costs of delay and waiting, which account for unsuccessful deliveries. Consequently, the result of the vehicle assignment performed by the company has more positive gaps that deviate from the model proposed by [1], [15], [21], [23], [44].

This experiment highlights our proposed model's effectiveness because in the model, RL quickly adapts to the transport environment uncertainties with distinctive patterns of abnormalities similar to the prior environment. These abnormalities consist of shipment incidents (e.g., postponement and cancellation of shipments), over-usage of vehicles, and capacity. The optimal policies for handling these events merge and rearrange shipments using RL, which executes optimizations from learned experience.

This experience is more feasible when handling uncertain events than normal environment states, which are more diverse. With uncertainties, more in-depth data are required to create environment policies that satisfy all features. Therefore, using the previous environment states defined by the reward processing unit as experiences for RL enables better adaptability and effectiveness for the model.

Furthermore, our proposed model considers the staff's general questions when performing vehicle routing tasks under uncertainties, such as "Is it feasible for the customer to take the delivery?" "Should we proceed or postpone the order?" and "What is the success rate of delivery to customers at a specific time of the day?" This way of thinking is input to the proposed model through the utility function of the reward processing unit. Thus, it forces the model to obtain enhanced vehicle routing solutions.

In summary, the SDVRP should consider utility and productivity from the deliveries and the distance in real-world applications because the minimum distance route is not always optimal. For example, route sequences at 8:00 a.m. differ from those at noon, owing to heavier traffic, resulting in the possibility of delayed delivery.

This study provides a relevant development direction that advances agents' experience to resemble human experience in vehicle route optimization. Moreover, the proposed methodology exhibits superior performance over approaches developed in previous studies that excluded information from the environment. However, the limitation of the model includes decreased efficiency when uncertain events do not exhibit a similar pattern from previous events. As shown in the experimental result, under such situations, the model's efficiency decreases by 19.83%, indicating room for improvement as future work to consider uncertainties with different patterns.

E. PRACTICAL APPLICATIONS AND LIMITATIONS

Currently, information from multi-sensors is essential for decision-making supporting tools in many disciplines (e.g., smart-city planning, transportation, and governance). In this study, the data-driven approach was applied to support logistics vehicle-route optimization. This study aimed to compute an optimal routing sequence for delivering goods that are robust to any disturbances from the transport environment using data processing to extract relevant information.

The extracted information was used to determine the first-visited locations based on traffic conditions, successful delivery rates, current fleet utility, and productivity. Then, a decision was made according to the information retrieved from the data. The proposed model combined data-driven and RL approaches to train the RL agent to solve the VRP as a decision-support tool. This model is not meant to replace humans. However, some tasks are replaced by ML and RL for better efficiency. This action allows humans to make decisions better on high-level decision-making tasks, such as negotiations between parties or evaluating the suitability of solutions suggested by AI.

Additional extensions can be made for the proposed methodology to make it more practical to other areas:

1) Controlling road traffic signals to optimize time delays at intersections to solve traffic congestion.
2) Controlling urban public transportation to solve public transportation shortages based on schedules.
3) Determining taxi routes and stands to optimize taxi parking locations to increase profit by improving the efficiency of trips.
4) Determining electric vehicle charging-station locations.

It is therefore unnecessary to modify the model structure in these contexts when applied to new applications because the model is data-driven. However, the coordinate data for creating an environment and inputs to the model must be in the corresponding format.

The main limitation of the proposed methodology is that it supports only structured data. Therefore, future data should be well-structured with organized attributes in the same manner as those presented in Section IV-A. The mandatory features required for the optimization model include the coordinates of longitude and latitude, targeted system utility, and

productivity ratio. If unstructured data are involved or new features are created, data preprocessing method and model tuning are required before the data can be used as inputs.

VIII. CONCLUSION

This paper proposed a novel methodology for a new-vehicle route optimization model using an RL interconnected with a tree-based regression model to create the RL's transport environment. The proposed model was adaptable and capable of handling multiple transport-environment settings. According to our understanding, the RL agent used previous environment states as experiences to select appropriate actions for determining vehicle routes in the current time by choosing from optimal policies. The experiment showed that using prior environment states as experience improved its ability to adapt to changes in the transport environment.

Additionally, the proposed model was generic such that the adjustment and expansion could be made with minimal modifications, owing to the data-driven ability of the model-based RL principle and the attention mechanism developed by Kool *et al.* and Nazari *et al.*

The actual data extracted from reports on vehicle scheduling and disturbances in the route optimization process by company staff of multiple case studies were applied to demonstrate the practicality of the proposed SDVRP. Furthermore, a new hybrid model proposed in this paper was trained with the A3C algorithm with experiences obtained from the tree-based regression model to maximize agent action utility. Based on the result, the model outperformed other state-of-art methods and previous approaches in terms of their effectiveness.

As a result, a daily vehicle route suggested for the logistics company yielded a profit of 16,495 THB (approximately 520 USD) with an average improvement of 5.32% for non-incident 32,990 THB (approximately 1,039 USD) with an average improvement of 37.63% for the incident case. The rationale behind this improvement comes from storing the RL's experiences learned previously in the tree-based regression model to influence the RL agent's current behavior. The stored experiences trained the RL agent to perform appropriate vehicle routing actions. Thus, the global optimal was returned.

However, the proposed methodology had some limitations. The first is that the stored experiences could only handle some aspects of the events when they did not match a similar pattern. Therefore, there may be difficulties for the RL agent to adapt its strategy to this event change, resulting in less effectiveness. The second limitation is that, when considering uncertainties of events, such as demand, vehicle utility, and productivity, using prior environment states as experience for training the RL agent is only practical to a certain extent. There is room for improvement when dealing with complex transport environments.

As a future research direction, behavioral and root-cause analysis can be implemented to analyze the dynamic nature of the various logistical environments. Then, prior experiences can be applied to improve vehicle route optimization solutions. Hence, the solution quality and ability to handle uncertainties in real-world settings can be significantly improved.

ACKNOWLEDGMENT

The authors would like to thank our colleagues from Jirajareon Company Ltd., who provided insight and expertise that greatly assisted our study.

DISCLOSURE STATEMENT

No potential conflict of interest was reported by the authors.

REFERENCES

[1] G. Musolino, A. Polimeni, and A. Vitetta, "Freight vehicle routing with reliable link travel times: A method based on network fundamental diagram," *Transp. Lett.*, vol. 10, no. 3, pp. 159–171, 2018.
[2] A. Polimeni and A. Vitetta, "Vehicle routing in urban areas: An optimal approach with cost function calibration," *Transportmetrica B, Transp. Dyn.*, vol. 2, no. 1, pp. 1–19, Jan. 2014.
[3] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. London, U.K.: MIT Press, 2017.
[4] K. Braekers, K. Ramaekers, and I. Van Nieuwenhuyse, "The vehicle routing problem: State of the art classification and review," *Comput. Ind. Eng.*, vol. 99, pp. 300–313, Sep. 2016.
[5] N. Helal, F. Pichon, D. Porumbel, D. Mercier, and É. Lefèvre, "The capacitated vehicle routing problem with evidential demands," *Int. J. Approx. Reasoning*, vol. 95, pp. 124–151, Apr. 2018.
[6] A. Hottung and K. Tierney, "Neural large neighborhood search for the capacitated vehicle routing problem," in *Proc. Frontiers Artif. Intell. Appl.*, vol. 325, 2020, pp. 443–450.
[7] C. Lin, K. L. Choy, G. T. S. Ho, S. H. Chung, and H. Y. Lam, "Survey of green vehicle routing problem: Past and future trends," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 1118–1138, 2014.
[8] Y. Xiao, Q. Zhao, I. Kaku, and Y. Xu, "Development of a fuel consumption optimization model for the capacitated vehicle routing problem," *Comput. Oper. Res.*, vol. 39, no. 7, pp. 1419–1431, Jul. 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0305054811002450
[9] J. Zhang, Y. Zhao, W. Xue, and J. Li, "Vehicle routing problem with fuel consumption and carbon emission," *Int. J. Prod. Econ.*, vol. 170, pp. 234–242, Dec. 2015.
[10] Ç. Koç and I. Karaoglan, "The green vehicle routing problem: A heuristic based exact solution approach," *Appl. Soft Comput.*, vol. 39, pp. 154–164, Feb. 2016.
[11] D. A. Jovanović, D. S. Pamučar, and S. Pejčić-Tarle, "Green vehicle routing in urban zones—A neuro-fuzzy approach," *Expert Syst. Appl.*, vol. 41, pp. 3189–3203, Jun. 2014.
[12] S. Erdoğan and E. Miller-Hooks, "A green vehicle routing problem," *Transp. Res. E, Logistics Transp. Rev.*, vol. 48, no. 1, pp. 100–114, 2012.
[13] Y. Xiao and A. Konak, "A genetic algorithm with exact dynamic programming for the green vehicle routing & scheduling problem," *J. Cleaner Prod.*, vol. 167, pp. 1450–1463, Nov. 2017.
[14] V. Pillac, C. Guéret, and A. L. Medaglia, "An event-driven optimization framework for dynamic vehicle routing," *Decis. Support Syst.*, vol. 54, no. 1, pp. 414–423, Dec. 2012.
[15] L. D. Giovanni, N. Gastaldon, M. Losego, and F. Sottovia, "Algorithms for a vehicle routing tool supporting express freight delivery in small trucking companies," *Transp. Res. Proc.*, vol. 30, pp. 197–206, Jan. 2018.
[16] F. Ferrucci, S. Bock, and M. Gendreau, "A pro-active real-time control approach for dynamic vehicle routing problems dealing with the delivery of urgent goods," *Eur. J. Oper. Res.*, vol. 225, no. 1, pp. 130–141, Feb. 2013.
[17] C. Ning and F. You, "Optimization under uncertainty in the era of big data and deep learning: When machine learning meets mathematical programming," *Comput. Chem. Eng.*, vol. 125, pp. 434–448, Jun. 2019.
[18] Y. Shi, Y. Zhou, T. Boudouh, and O. Grunder, "A lexicographic-based two-stage algorithm for vehicle routing problem with simultaneous pickup–delivery and time window," *Eng. Appl. Artif. Intell.*, vol. 95, Oct. 2020, Art. no. 103901.
[19] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, vol. 2. Montréal, QC, Canada: MIT Press, 2014, pp. 3104–3112.

[20] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Advances in Neural Information Processing Systems*, vol. 1. Montreal, QC, Canada: Curran Associates, 2015, pp. 2692–2700.
[21] W. Kool, H. Van Hoof, and M. Welling, "Attention, learn to solve routing problems!" in *Proc. 7th Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–25.
[22] H. Dai, E. B. Khalil, Y. Zhang, B. Dilkina, and L. Song, "Learning combinatorial optimization algorithms over graphs," in *Advances in Neural Information Processing Systems*, vol. 1. Long Beach, CA, USA: Neural Information Processing Systems Foundation, 2017, pp. 6349–6359.
[23] M. Nazari, A. Oroojlooy, M. Takáč, and L. V. Snyder, "Reinforcement learning for solving the vehicle routing problem," in *Advances in Neural Information Processing Systems*, vol. 1. Montréal, QC, Canada: Curran Associates, 2018, pp. 9839–9849.
[24] I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio, "Neural combinatorial optimization with reinforcement learning," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*. Toulon, France, 2017, pp. 1–5.
[25] T. M. Moerland, J. Broekens, and C. M. Jonker, "Model-based reinforcement learning: A survey," in *Proc. Int. Conf. Electron. Bus. (ICEB)*, 2020, pp. 421–429.
[26] I. Drori, Y. Krishnamurthy, R. Lourenco, R. Rampin, K. Cho, C. Silva, and J. Freire, "Automatic machine learning by pipeline synthesis using model-based reinforcement learning and a grammar," in *Proc. 6th ICML Workshop Automated Mach. Learn.*, 2019, pp. 1–8.
[27] I. Drori, Y. Krishnamurthy, R. Rampin, R. De, P. Lourenco, J. P. Ono, K. Cho, C. Silva, and J. Freire, "AlphaD3M: Machine learning pipeline synthesis," in *Proc. AutoML Workshop ICML*, 2018, pp. 1–8.
[28] C. Mao and Z. Shen, "A reinforcement learning framework for the adaptive routing problem in stochastic time-dependent network," *Transp. Res. C, Emerg. Technol.*, vol. 93, pp. 179–197, Aug. 2018.
[29] N. Mazyavkina, S. Sviridov, S. Ivanov, and E. Burnaev, "Reinforcement learning for combinatorial optimization: A survey," 2020, *arXiv:2003.03600*.
[30] Y. Bengio, A. Lodi, and A. Prouvost, "Machine learning for combinatorial optimization: A methodological tour d'horizon," *Eur. J. Oper. Res.*, vol. 290, no. 2, pp. 405–421, Apr. 2021.
[31] A. Haj-Ali, N. K. Ahmed, T. Willke, J. Gonzalez, K. Asanovic, and I. Stoica, "A view on deep reinforcement learning in system optimization," 2019, *arXiv:1908.01275*.
[32] M. W. Ulmer, J. C. Goodson, D. C. Mattfeld, and B. W. Thomas, "On modeling stochastic dynamic vehicle routing problems," *EURO J. Transp. Logistics*, vol. 9, no. 2, Jun. 2020, Art. no. 100008.
[33] L. L. B. V. Cruciol, A. C. de Arruda, L. Weigang, L. Li, and A. M. F. Crespo, "Reward functions for learning to control in air traffic flow management," *Transp. Res. C, Emerg. Technol.*, vol. 35, pp. 141–155, Oct. 2013.
[34] M. Lombardi and M. Milano, "Boosting combinatorial problem modeling with machine learning," in *Proc. 27th Int. Joint Conf. Artif. Intell. (IJCAI)*, Jul. 2018, pp. 5472–5478.
[35] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2010.
[36] E. Frazelle, *Supply Chain: The Logistics of Supply Chain Management*. New York, NY, USA: McGraw-Hill, 2002. [Online]. Available: http://www.lavoisier.fr/livre/notice.asp?id=O3LW2LA2RR3OWA
[37] M. Christopher, *Logistics and Supply Chain Management*, vol. 41, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2005.
[38] T. Phiboonbanakit, V.-N. Huynh, T. Horanont, and T. Supnithi, "Detecting abnormal behavior in the transportation planning using long short term memories and a contextualized dynamic threshold," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput., ACM Int. Symp. Wearable Comput.*, London, U.K., Sep. 2019, pp. 996–1007.
[39] J. Insa-Cabrera, D. L. Dowe, and J. Hernández-Orallo, "Evaluating a reinforcement learning algorithm with a general intelligence test," in *Proc. Conf. Spanish Assoc. Artif. Intell.*, in Lecture Notes in Computer Science, vol. 7023, 2011, pp. 1–11.
[40] D. L. Poole and A. K. Mackworth, *Artificial Intelligence: Foundations of Computational Agents*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2017.
[41] M. E. Taylor, B. Kulis, and F. Sha, "Metric learning for reinforcement learning agents," in *Proc. 10th Int. Conf. Auton. Agents Multiagent Syst. (AAMAS)*, vol. 2, 2011, pp. 729–736.
[42] S. Kotz, N. L. Johnson, and C. B. Read, *Encyclopedia of Statistical Sciences*, 1st ed. Hoboken, NJ, USA: Wiley, 2006.
[43] B. S. Everitt and A. Skrondal, *The Cambridge Dictionary of Statistics*, 4th ed. Cambridge, U.K.: Cambridge Univ. Press, 2010.
[44] Google Developers. (2019). *Google OR-Tools*. [Online]. Available: https://developers.google.com/optimization

THANANUT PHIBOONBANAKIT received the B.S. and M.S. degrees in science and the Ph.D. degree in engineering and technology from the Sirindhorn International Institute of Technology, Thammasat University, Thailand, in 2015, 2017, and 2020, respectively, and the second Ph.D. degree in knowledge science from the Japan Advanced Institute of Science and Technology (JAIST), Japan, in 2021. He is currently working as a Specialist–Data Science, Information Technology Department at Kerry Express (Thailand) Public Company Ltd., Thailand.

TEERAYUT HORANONT received the B.Arch. degree in architecture from Chulalongkorn University, Thailand, in 1999, the M.Sc. degree in remote sensing and geographic information system from the Asian Institute of Technology (AIT), Thailand, in 2002, and the Ph.D. degree in spatial information engineering from The University of Tokyo, Japan, in 2010. He is currently an Assistant Professor at the School of Information, Computer, and Communication Technology, Sirindhorn International Institute of Technology (SIIT), Thammasat University, Thailand.

VAN-NAM HUYNH (Member, IEEE) received the Ph.D. degree in mathematics from the Vietnam Academy of Science and Technology in 1999. He is currently a Professor of the School of Knowledge Science, Japan Advanced Institute of Science and Technology (JAIST). He was also a Visiting Professor at the National Electronics and Computer Technology Center, Thailand (2019), an Adjunct Professor at Chiang Mai University, Thailand (2015–2017), and a part-time Lecturer at Tsukuba University, Japan (2011–2015). His current research interests include AI and machine learning, modeling and reasoning with uncertain knowledge, argumentation, multi-agent systems, decision analysis and management science, and Kansei information processing and applications. He has (co-) edited over 15 Springer volumes and seven special issues for *International Journal of Approximate Reasoning*, *Annals of Operations Research*, and *Data and Knowledge Engineering*. He is also serving as an Area Editor of *International Journal of Approximate Reasoning*, the Editor-in-Chief of *International Journal of Knowledge and Systems Science*, and an Editorial Board Member of the *Array* journal.

THEPCHAI SUPNITHI received the B.S. degree in mathematics from Chulalongkorn University, Thailand, in 1992, and the M.S. and Ph.D. degrees in engineering from Osaka University, Japan, in 1997 and 2001, respectively. He is currently the Director of the Artificial Intelligence Research Group, NECTEC, Thailand.