Title	貨物輸送への応用を伴う行動と根本原因分析のためのハイブリッドモデルに関する研究
Author(s)	PHIBOONBANAKIT, Thananut
Citation	
Issue Date	2021-09
Type	Thesis or Dissertation
Text version	ETD
URL	http://hdl.handle.net/10119/17521
Rights	
Description	Supervisor:Huynh Nam Van, 先端科学技術研究科, 博士

A Study on Hybrid Models for Behavior and Root-Cause Analysis with Application to Freight Transportation

PHIBOONBANAKIT, Thananut

Japan Advanced Institute of Science and Technology

Doctoral Dissertation

A Study on Hybrid Models for Behavior and Root-Cause Analysis with Application to Freight Transportation

PHIBOONBANAKIT, Thananut

Supervisor: Professor HUYNH, Nam Van

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
[Knowledge Science]
September 2021

Abstract

Currently, data from multiple sensors and the Internet of Things (IoT) provide essential mobility information for both governments and industries. They use this information to support smart city planning, medical care, and transportation domains. Recently, the transportation domain has played an essential role in the era of digital eCommerce, especially in transport logistics. The rapid growth of logistics demand also reflects the cost and profit of the logistics industry.

Historical statistics show that in 2019, the logistics industry had only 56% fleet utilization in the United States and 54% in Europe. The lack of efficiency in vehicle route optimization caused difficulties in transportation planning and management and created a direct impact on operational costs. To solve this issue, logistics agencies use the data obtained from the IoT to support their operations, for instance, transportation scheduling, planning, and resource allocation. Their goal is to obtain a suitable policy that can minimize agency operational costs and reveal the potential of route optimization.

The policy described in this study is used for managing the vehicle route optimization process. Therefore, numerous methodologies have been introduced to extract rich information from these data. Furthermore, anomaly detection and root-cause analysis are performed to understand the transport operation characteristics. However, these data come from multiple sources. Therefore, conventional methods cannot handle these data directly because of different data formats, and the data are also dependent on spatial-temporal contexts and behavior attributes.

To address these problems, this study provides a novel methodology for performing anomaly detection (e.g., temporal and static anomalies) and root-cause analysis for transportation logistics (e.g., explanation of anomaly in transportation logistics). Later on, the anomaly detection models contributed to analyzing both transportation environment and reinforcement learning (RL) agent's behavior in optimizing daily vehicle routing for the logistics agency. This phenomenon is presented in the case studies. The author assumes that the RL agent has the same role as humans. Suppose that the optimal vehicle route is obtained by the RL agent, it denotes that when a human follows this recommended route pattern provided the optimal decision.

The methodology consists of five models. They are used in two different stages: (1) the detection stage and (2) root-cause analysis stage. In the first stage, anomaly detection using Long Short-Term Memory (LSTM)-based and unsupervised hybrid anomaly detection models is proposed. These two models are designed to detect point, contextual, and collective anomalies. In the second stage, forward and inverse problem analysis models are proposed. They are also compared with the machine learning-based model to derive the root cause of the detected anomaly. These outcomes will increase the reliability and interpretability of the anomaly detection result. The obtained outcomes also increase anomaly detection rates and significantly reduce the bias of labeling the data.

The data from multiple sensors are preprocessed and transformed into structured data, and the features are extracted using feature engineering to perform this experiment. A different set of anomaly detection methods is then used to distinguish between regular operation patterns and disturbances. Its outcome is further used as an input to analyze the root cause of disturbances. Finally, root-cause analysis is performed. Thus, these steps are employed to analyze environmental changes. The analyzed information is then used to adjust the RL agent's behavior when it optimized vehicle routes. The vehicle route optimization solution is therefore adapted to environmental changes by doing so.

To demonstrate the practicality of the proposed methodology, the experimental results are validated with real data and compared against state-of-the-art models. Once the model for detecting anomalies in transportation is developed, the model was also applied to the other application domains to demonstrate the model's generality. The results show an accuracy of up to 0.83 (0.88 of the area under the RoC curve) with less processing time than that required by other existing methods. The model is also general and can be employed in other application domains with minor modifications. Finally, real case studies are presented to demonstrate the practical significance of anomaly detection and root-cause analysis in assisting vehicle route optimization tasks.

The real case studies' results implied that the interconnection between RL, behavior analysis, and reward processing of the proposed model increased the ability of the agent to perform vehicle route optimizations in a similar way as humans for routine daily scheduling. Furthermore, when uncertain changes (e.g., the sudden change of customer demand, road-network traffic condition, and fleet resources) occurred in the environment, the agent also outperformed the humans when making rescheduling decisions. Thus, this proposed methodology improved the vehicle route optimization solution up to 57.91% of profit improvement when compared against the optimal baselines.

Keywords: Anomaly detection, Deep learning, Logistics, Root-cause analysis, Transportation.

Acknowledgements

First, I would like to express my sincere gratitude to my supervisors, Prof. Dr. HUYNH, Nam Van, my second supervisor, Prof. Dr. YUIZONO Takaya, and Asst. Prof. Dr. HORANONT, Teerayut, for their continuous support of my doctoral study and their patient guidance, enthusiastic encouragement, and useful procedural guidelines during my study at Japan Advanced Institute of Science and Technology (JAIST) and Sirindhorn International Institute of Technology (SIIT), Thammasat University.

Additionally, many thanks to my supervisor in the dual degree program, Dr. SUPNITHI, Thepchai from National Electronics and Computer Technology Center for his comments and advice on my research. Without their guidance and persistent help, this dissertation would not have been possible.

In addition to my supervisors, I would like to thank my minor research advisor, Prof. Dr. FUJINAMI Tsutomu, and the committee members, Prof. Dr. DAM, Hieu Chi and Prof. Dr. IKEDA Mitsuru, for their valuable advice and comments during my study. Additional thanks to the JAIST-SIIT collaboration scholarship program of JAIST and SIIT, Thammasat University, the Monbukagakusho Honors scholarship offered by Japan Student Services and Organization (JASSO), and the US Office of Naval Research Global under grant no. N62909-19-1-2031, which gave me the best opportunity of my life to study as a graduate student.

This dissertation would not be complete without the generosity of our third-party company that supported the mobility data and the evaluation of this dissertation. Additionally, my sincere thanks to my senior colleagues and friends at the laboratory who study and work in this field. Their friendship and assistance mean more to me than I can ever express. I could not have completed my research without their invaluable assistance. Finally, my deepest gratitude goes to my beloved family for their support during my doctoral study.

PHIBOONBANAKIT, Thananut

Contents

This dissertation was prepared according to the curriculum for the Collaborative Education Program organized by Japan Advanced Institute of Science and Technology and Sirindhorn International Institute of Technology, Thammasat University.

List of Figures

List of Tables

List Of Symbols/Abbreviations

Symbols/Abbreviations	**Terms**
AHP	Analytic Hierarchy Process
AI	Artificial Intelligence
AB	Absolute Error
AE	Autoencoder
A3C	Asynchronous Actor and Critic Algorithm
ANN	Artificial Neural Network
AUC	Area Under the Roc Curve
BS	Beam Search
CAV	Cumulative Action Value
CI	Confidence Interval
DAE	Deep Autoencoder
DBSCAN	Density-based Spatial Clustering of Applications with Noise
DSS	Decision Support System
EM	Expectation Maximization Algorithm
EWMA	Exponentially Weighted Average
FN	False Negative
FP	False Positive
FPR	False Positive Rate
GA	Genetic Algorithm
GAM	General Additive Model
GD	Greedy Best-First Search
GMM	Gaussian Mixture Model
GPS	Global Positioning System
HMM	hidden Markov model
IS	Isolate Forest
IoT	Internet of Things
IT	Information Technology
Km	Kilometer
k-nn	k-nearest neighbors Algorithm

LSTM	Long Short-Term Memory
LSTM-AE	Long Short-Term Memory Autoencoder
MAE	Mean Absolute Error
ML	Machine Learning
MPI	Mutual Preferential Independence
MSE	Mean Square Error
MVRP	Multi-depot Vehicle Routing Problem
MUI	Mutually Utility Independent
NP	Non-deterministic Polynomial-time
One-SVM	One-class Support Vector Machine
OR	Operation Research
OSS	Open-source Software
PCA	Principle Component Analysis
PNN	Probabilistic Neural Network
PSO	Particle Swarm Optimization
RA	Root-Cause Analysis
RF	Random Forest
RL	Reinforcement Learning
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic Curve
SAE	Stack Auto Encoder
SDVRP	Split-Delivery Vehicle Routing Problem
SVDD	Support Vector Data Description
SVM	Support Vector Machine
TSP	Travel Salesman Problem
TPR	True Positive Rate
TP	True Positive
VRP	Vehicle Routing Problem

Chapter 1

Introduction

1.1 Introduction and Theoretical Framework

Currently, data from multiple sensors and the Internet of Things (IoT) provide essential mobility information for both governments and industries. They use this information to support smart city planning, medical care, and transportation domains. Recently, the transportation domain has played an essential role in the era of digital eCommerce, especially in transport logistics. The rapid growth of logistics demand also reflects the cost and profit of the logistics industry.

Historical statistics show that in 2019, the logistics industry had only 56% fleet utilization in the United States and 54% in Europe [1]. The lack of efficiency in vehicle route optimization causes difficulties in transportation planning and management and creates a direct impact on operational costs [2]. To solve this issue, the data obtained from IoTs is used by logistics agencies to support their operations, for instance, transportation scheduling, planning, and resource allocation. Their goal is to obtain an appropriate policy that can minimize agency operational costs and reveal the potential of route optimization.

The policy denoted in this study is used for managing the vehicle route optimization process. Therefore, various methodologies are introduced to extract rich information from these data. Furthermore, anomaly detection and root-cause analysis are performed to understand the transport operation characteristics. However, these data come from multisensors. Therefore, current methods cannot handle these data directly because of different data formats and reliance on different contexts, as mentioned in [3, 4, 5]. These critical attributes derived from the literature and industry motivate the author to investigate and propose a new methodology for anomaly detection and root-cause analysis. The proposed methodology monitors vehicle route optimization processes in logistics management. It implements machine learning (ML) and artificial intelligence (AI) to address environmental uncertainties and disturbances.

1.2 Problem Statement

By investigating the operation of a logistics agency in Thailand, it was discovered that the agency has various types of data flows to the system. For instance, data from multiple sensors and IoTs installed on vehicles. These data are crucial for assisting in decision-making, such as developing management policies for transportation scheduling, planning, resource allocation, and production. However, these data are massive, fast, and variable. These are data characteristics of big data. Moreover, the data contain outliers and noise. The data attributes are also not clearly separate when projecting high-dimensional data into a low-dimensional space.

Furthermore, some information is mistakenly removed when the data dimensions are reduced. Therefore, it is difficult for staffers in the logistics agency to manually analyze these data or use conventional analysis techniques.

These issues are also widely discussed in academic research. As a result, the low utility and productivity of these data usages also cause the agency to lose opportunities to discover crucial insights for policy development and eliminate disturbances in their fleet management. There is also a high risk of making an incorrect decision when a critical event occurs. Thus, the solutions proposed by conventional methods are not always guaranteed to be feasible to address this study's problem.

1.3 Purpose of the Study

This study proposes a new methodology that can effectively monitor and detect disturbances in the vehicle route optimization process of logistics management. This process takes large-scale data for analysis. The methodology for this study consists of two parts.

The first part entails behavior analysis (anomaly detection), which detects temporal and static anomalies that occur when the vehicle route optimization process is performed. The result is also used to assist in root-cause analysis and further optimization tasks.

The second part entails the root-cause analysis, which determines the root cause of the detected anomaly and validates it using case studies. Therefore, when anomalies are detected, the cause of the anomaly is forwarded to an administrative person in the logistics management division, which can help logistics agencies avoid a critical event that disrupts the vehicle route optimization process.

This study aims to help agencies develop a sustainable operation monitoring and disturbance detection technique for vehicle route optimization by leveraging two expected outcomes. First, it provides information about the ability and efficiency of a fleet to evaluate an agency's operational performance. Second, the obtained data are taken as inputs to support the vehicle route optimization process. As a result, the vehicle route optimization solution is returned with a minimum operational cost. Furthermore, an appropriate policy for managing route optimization is also obtained.

The author believes that this research will have a remarkable impact if the methodology can provide recommendations on the tasks to be set by agencies by presenting rich information related to feasibility compared with the fleet operational capacity in terms of regular and disturbance operation states. The use of ML and AI helps in the detection of disturbances and extracts rich information related to the disturbances, which can be utilized as a guideline for performing vehicle route optimization.

1.4 Significance of the Study

In recent years, the transportation logistics problem has become a crucial task for both academia and industry. They aim to understand the behavior of transportation operations to assist in developing appropriate management policies. The most popular methodologies that are widely used are anomaly detection from large-scale data. Accordingly, a considerable number of solutions and methodologies have been introduced. Each methodology has achieved its goal of problem solving in transportation logistics. However, the real-world problem is much more complicated than ever before. Therefore, implementing these methodologies directly in real-world applications is not feasible.

With respect to the anomaly detection problem, most methodologies have a problem when large-scale and noisy data are streamed into the process. This problem directly impacts the reliability of the detection result. Furthermore, the methods that reveal the causes of the anomaly are also in the development stages. Therefore, it is necessary to have a methodology that can perform two tasks in sequence and handle large-scale streaming data from multiple sources.

Fortunately, the deployment of ensemble, two-stage anomaly detection and root-cause analysis methods has attracted researchers' interest in recent years. Unfortunately, those methodologies cannot be applied directly because of the following gaps.

1. The data are from many sources that contain noise and outliers. In addition, they also have different shapes and dimensions and are not linearly separable (e.g., the distance between points and the reconstruction error of the anomaly event is similar to that of the normal event). Therefore, it is not easy to detect anomalies using general single-level approaches (e.g., clustering, statistical analysis, or classification techniques) directly.

2. Detecting anomalies that are correlated with different data contexts are limited by the current methodologies. The anomalies are correlated by spatial-temporal contexts (e.g., time and location) and behavior attributes. For simplicity, behavior attributes are a record of actions performed in the environment or by the computer system (e.g., speed, usage, and incidents).

3. The components that discover the cause of the anomaly are still under development, as mentioned in [4].

4. The pattern of data continues to change over time. Therefore, when new data patterns are inputted into the model, the model must be rebuilt to consider these new changes.

If these issues still exist, it is difficult for the logistics agency to obtain a reliable vehicle route optimization solution. Moreover, if the decision is made based on inaccurate information, it might reduce the agency's profit and opportunities to compete in the market.

This study proposes a novel methodology for anomaly detection and root-cause analysis to fill the gaps in the current studies. This methodology was initially designed for transportation logistics domains. However, it can also be expanded to other research domains to detect anomalies and their root causes with minimal changes.

1.5 Dissertation Organization

This dissertation contains 7 chapters, as follows:

- **Chapter 1** introduces anomaly detection and its application, the problem statement, the purpose of this study, its significance, and the chapter organization of this dissertation.

- **Chapter 2** provides a literature review to introduce the research background and literature review of behavior and root-cause analysis (e.g., the temporal anomaly detection problem, multilevel anomaly detection problems, and root-cause analysis). A summary of the remaining research gaps and motivations is included in this chapter.

- **Chapter 3** presents a methodology for detecting a temporal anomaly in logistics agency operations. At the end of the chapter, the experimental results are discussed.

- **Chapter 4** presents a methodology for anomaly detection and root-cause analysis using joint learning and ensemble methods. At the end of the chapter, the experimental results are discussed.

- **Chapter 5** presents a case study that applies behavior and root-cause analyses in vehicle route optimization. At the end of the chapter, the experimental results and case studies are discussed.

- **Chapter 6** presents the contributions of this study to this domain, including the practical and theoretical implications and knowledge science contributions.

- **Chapter 7** concludes the dissertation and discusses future works.

Chapter 2

Literature Review

2.1 Introduction

In this chapter, the related literature is reviewed based on the defined problem statement. The works are grouped into two subcategories: 1) works that review anomaly detection from a data series and 2) those that review methods and approaches for root-cause analysis to determine the causes of the anomalies. A review of these related works highlights the remaining gaps and areas for improvement that are essential for this research.

2.2 Previous Work

Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior. These patterns are referred to as anomalies, outliers, observations, and disturbances. The most famous terms are anomalies and outliers that are used in anomaly detection studies. Anomaly detection can also be called behavior analysis. This technique analyzes the behavior of data points that flow into the system. Anomaly detection is also widely used to support system administration in monitoring information technology (IT) [6] and industrial manufacturing infrastructures.

In Figure 2.1, the critical differences between anomaly detection and behavior analysis are visualized.

In addition to anomaly detection, an anomaly explanation study is also presented. Note that the author uses the term root-cause analysis for the term anomaly explanation in this dissertation.

2.2.1 Anomaly Types

According to [7], the authors stated that anomalies can be classified into three types: point, contextual, and collective anomalies. The details of each anomaly type are described as follows:

Point Anomalies

If a data point is different from the remaining data points in terms of distance and density, this data point is then defined as a point anomaly. A real example of this type of anomaly is credit card fraud. In this research field, fraud detection is developed to detect data points that show a character that is different from others.

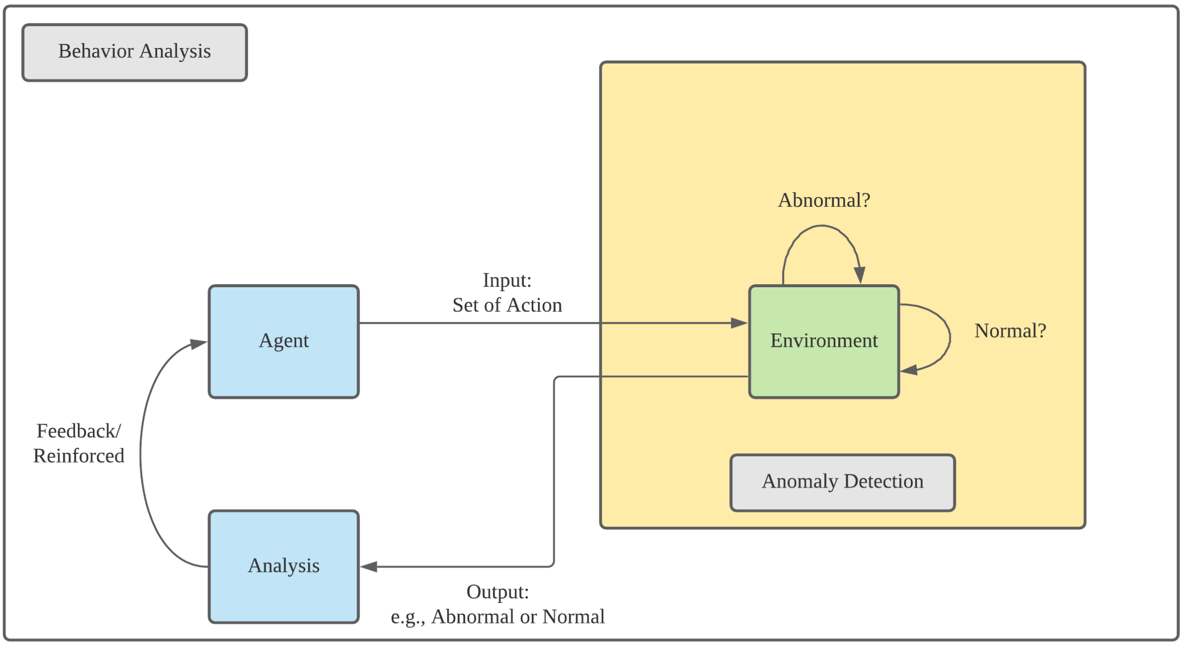


Figure 2.1: Demonstration of the differences between anomaly detection and behavior analysis approaches.

Contextual Anomalies

If a data point is an anomaly in a specific context, but for another context, it is not, then this kind of data point is defined as a contextual anomaly. This can also be called a condition anomaly. This type of anomaly uses two attributes to determine anomalies. It consists of contextual and behavioral attributes.

The contextual attributes are used to determine the context of the data point. For example, in spatial datasets, the coordinates of latitude and longitude are contextual attributes. For time-series datasets, time is a contextual attribute that determines the position of the data point in the time series.

The behavioral attributes define the noncontextual characteristics of the data point. For example, in a spatial dataset describing the average rainfall of the entire world, the amount of rainfall at any location is a behavioral attribute [7]. Therefore, if the rainfall is over the average of the specific threshold, then it is defined as an anomaly. Usually, the threshold is defined as $\mu + n\sigma$, where μ is a mean of the data sequence's value, n is the size of the standard deviation, and σ is the standard deviation of the data sequence's value.

Collective Anomalies

If a collection of related data points is anomalous concerning the entire dataset, it is classified as a collective anomaly. The individual data instances in a collective anomaly may not be anomalies by themselves, but their occurrence together as a collection is anomalous [7]. To determine this kind of anomaly, the most efficient method is to perform the sliding windows technique.

Therefore, if a group of data points in a sliding window has the same pattern and differs from other data records, then this group of data points is considered an anomaly. An example of this kind of anomaly is shown in Figure 2.2. Note that Figure 2.2 is taken from [7]'s study.

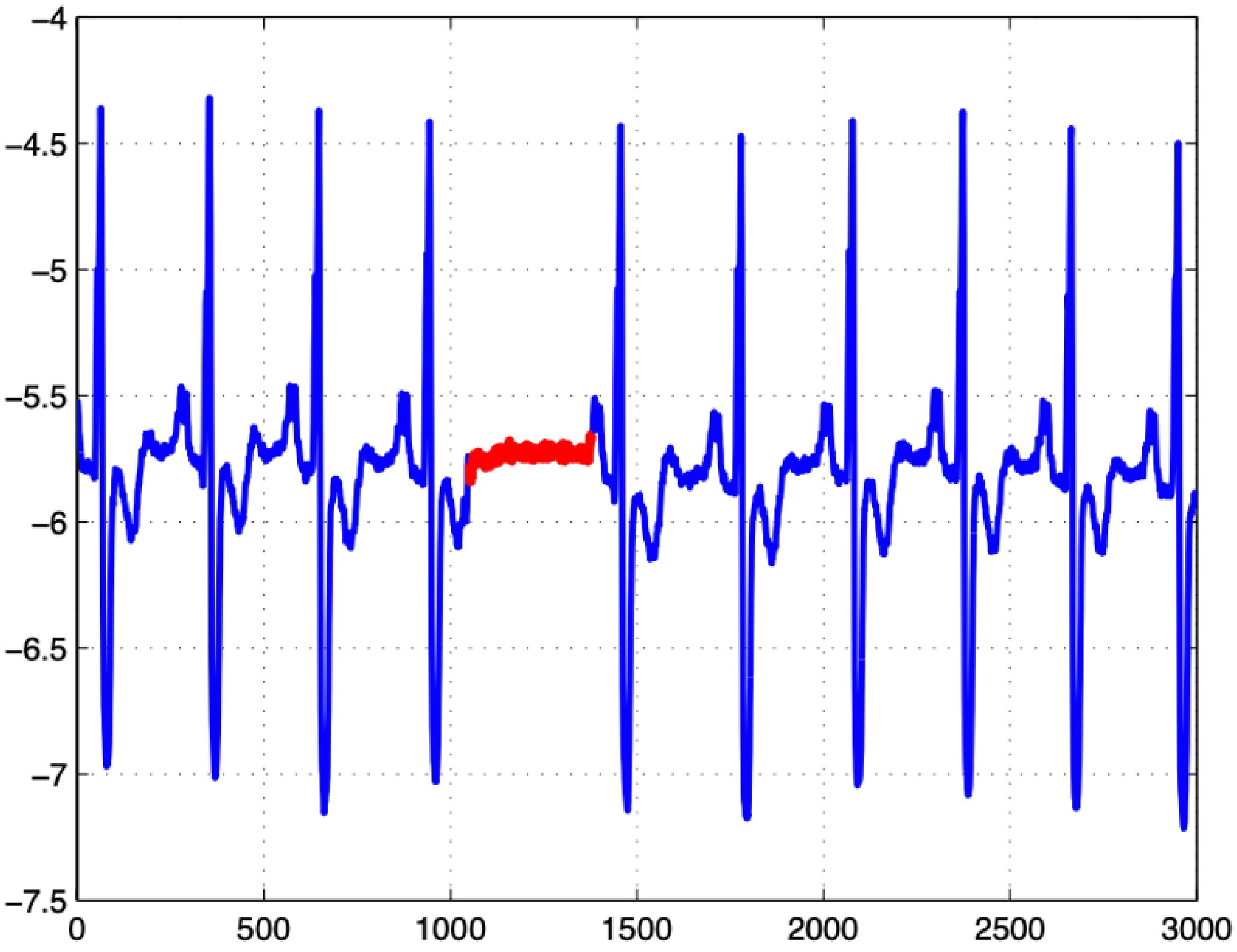


Figure 2.2: Demonstration of collective anomalies; the red line illustrates the collective anomaly, which has more frequency and differs from the other data patterns.

2.2.2 Behavior Analysis in Large-scale Data

According to a survey of the current research in terms of behavior analysis, the results and findings can be grouped into two categories: efficiency and reliability.

Generally, businesses use behavior analysis to detect abnormalities in business processes. For instance, management systems [6], industrial and manufacturing [8, 9], financial [10, 11, 12], health-care monitoring, and transportation systems [13]. Behavior analysis is also used for smart city planning [14] and IoT and networking control [15, 16].

Behavior Analysis for Solving Detection Efficiency

Recent studies have applied supervised, semisupervised, and unsupervised detection approaches. Real-world data have limitations related to labels. Therefore, unsupervised approaches are more suited for this research problem because they do not require information about the data point to perform detection.

Ko and Comuzzi proposed a statistical leverage method to detect anomalies in business system log files. The authors showed that the proposed anomaly detection thresholds can handle variable case anomaly ratios more effectively than other recent literature methods [17].

For time-series data, Hundman et al. applied LSTMs to detect unexpected behaviors from a temporal correlation and proposed a dynamic threshold. The results showed that the defined threshold can detect most of the abnormalities in the data. However, it still exhibited a high rate of false detection [18].

Similar to Hundman et al.'s study, Elsayed et al. proposed an LSTM autoencoder to detect anomalies in a computer network. However, instead of using LSTMs, the author proposed an LSTM autoencoder (LSTM-AE) because it can learn the representations of the network dataset better than using LSTM-based models with prediction error thresholds. Threshold-based LSTMs do not easily manually handcraft and extract discriminatory features [19, 20].

Xu et al. proposed a model called "DONUT". It is an unsupervised anomaly detection model based on a variational autoencoder (VAE) [21]. The VAE was leveraged to model the reconstruction probabilities of normal time series. Assuming that the reconstruction error was larger than a threshold, the data points were then defined as abnormal points. Xu et al. proposed a concept drift adaptive method for enhancing the LSTM detection result [22].

Furthermore, Wang et al. discovered that examining anomalies from LSTM prediction error does not always guarantee that the error can represent the anomaly. Therefore, the authors proposed an encoder-double decoder model that uses an attention technique. The attention model enables the encoder to effectively summarize the information of the time series input by automatically detecting parts of the input data that are more relevant in reconstructing the data at different time steps. Unfortunately, this model also has a disadvantage. It relies on human experience in tuning the criteria to monitor the trends of reconstruction errors. Statistical analysis is required to analyze the reconstruction error [9].

Ding et al. proposed anomaly detection based on an LSTM and the Gaussian mixture model (GMM). The author called their algorithm LSTM-BP and showed that using LSTM alone cannot achieve a good result for all datasets [23]. Therefore, LSTM-BP was proposed for improving the detection results using the benefit of GMM. The GMM is inputted by the prediction result from the LSTM to cluster the data points and reveal the uncorrelated data points. The author also proposed the system health factor α to denote whether joint detection between the LSTM and GMM is required. They reduced the time complexity of the computation. Assuming that the LSTM-based model is a perfectly detected anomaly, it is unnecessary to use the GMM model to detect anomalies as redundant tasks. As a result, the detection result improves compared to the traditional LSTM-based model—unfortunately, the author focused on the low-dimensional data.

More approaches that used LSTMs for detecting abnormalities from time-series data are shown in [3, 4, 24, 25, 26, 27]. Kim et al. also mentioned that using only statistical analysis is not efficient for detecting anomalies as it contains different data distributions among data points. The author also stated that using unsupervised learning has a drawback because unsupervised learning methods cannot properly detect anomalies with the same statistical distribution as normal data. Thus, the author proposed C-LSTM to overcome this issue [28]. The C-LSTM combines benefits from convolutional neural networks (CNNs), LSTMs, and deep neural networks (DNNs). In addition to the LSTM and its variants, Ahmad et al. used hierarchical temporal memory (HTM) for dealing with real-time anomaly detection in streaming data. The model detection result is impressive. However, it also has room for improvement. The author showed that the prediction error is not always correlated to the anomaly.

Therefore, the model suffered from providing a more accurate detection than the ensemble method [29]. This technique was also used by Wu et al. The author used HTM to deal with real-time changes in the data pattern since it incorporates contextual information from the past to improve prediction results [30]. Some studies used reconstruction-based frameworks, for instance, deep autoencoder (DAE), stacked autoencoder (SAE), and LSTM autoencoder, for anomaly detection. Because the data are derived from multiple sources and are highly dimensional, various autoencoding (AE) models have proven to be more suitable for performing dimensional reduction tasks than principal component analysis (PCA) models or their extended types. AE was also found to be efficient in detecting system abnormalities [31, 32, 33]. Unlike PCA-based models, it assumes that the abnormalities do not always appear with a high degree of differences from another cluster.

DAEs have also been used to represent data in low dimensions before performing joint learning with clustering methods (e.g., K-means) [34] and density estimations (e.g., Gaussian mixture model (GMM) [5] and density-based spatial clustering of applications with noise (DB-SCAN) [35]).

To increment the model with every new data pattern, Zhou et al. [36] proposed contextual hidden Markov models (HMMs) for temporal-spatial data analysis. The model continues updating and predicting the next event regarding the temporal dependencies, current situation, and context. The computations are based on the HMM definition.

Behavior Analysis for Solving Detection Reliability

Recent research shows the development of a hybrid model with a multilevel detection approach to bridge the gap caused by some of the research questions. Selim et al. [37] trained a neural network and used the output in a decision tree to adopt the intrusion detection model. The experimental results showed that combining the advantages of each model can lead to higher accuracy and that detection becomes more reliable by connecting these two models. In other words, it is a two-step approach. However, if we compare the two-step approach with joint learning approaches, especially when performing dimensional reduction before clustering, the performance is less efficient than simultaneously training and performing classification or clustering, as shown in the study by [7].

To combine multiple models, which aims to increase the model reliability, an ensemble method is a suitable choice. The most famous ensemble methods are the majority vote [38], median, and weighted average [39, 40]. The ensemble methods combine a weak learner classifier prediction to decide the final detection results, as shown in [41]. They combined an average-based classifier, autoregressive linear predictor-based classifier, and neural network based on the weighted majority voting algorithm to achieve a reliable result. This procedure is also similar to Krawczyk et al. [42]. The authors proposed a clustering-based ensemble to utilize individual classifier advantages based on feature space partitioning. In addition, [43] used the ensemble method to combine a stacked autoencoder with a probabilistic neural network.

The results showed high accuracy and reliability. [44] also proved that ensembles are generally helpful to avoid the influence of bad decision makers but do not help to maximize ranking performance. They also contrasted with the use of ensemble strategies in machine learning, where the explicit goal is to achieve better performance than that of any individual classifier. Chen et al. proposed an ensemble randomly connected AE for detecting a disturbance. Multiple AE model architectures were constructed and used to perform the detection. After all AE models were executed, the process ensembled the result for the final detection result. The author measures the degree of disturbance from the reconstruction error of the AE. A higher degree means a high possibility that this data object is a disturbance [45]. The hidden AE layers consist of fewer neurons. These neurons are used to reconstruct the input as closely as possible. The weights in the hidden layers capture only the most representative features of the original input data. However, they discarded the detailed specifics of the input data, such as outliers [46]. Zhang et al. [47] proposed a multistage ensemble to improve outlier detection output. The authors showed that the hierarchical process in feature extraction is efficient in interpreting the features and reducing noise from the dataset.

Moreover, the bagging strategy enhances the efficiency in detecting outliers. Finally, the author proposed a stacking-based ensemble learning method to improve the reliability of the detection result. They selected appropriate base classifiers by performing the self-adaptive parameter optimization method. The classifiers that have the most optimized parameters are selected. Ensemble learning is also used at double levels for unsupervised anomaly detection as presented by [48]. The authors reduced the loss of information produced by the single-level ensemble method when it generates multiple subspaces from the dataset.

Furthermore, in this research area, a dynamic ensemble anomaly detection approach dynamically chooses the models for final detection from the change in the data object. Wang et al. [49] proposed a dynamic ensemble outlier detection model based on an adaptive k-nearest neighbor rule. The results showed an improvement over the static ensemble method. The rationale behind this is that the dynamic ensemble methods dynamically select suitable classifiers by majority voting and store them in a list. It stores the model until there are no further updates. After that, it averages the output of the classifiers from the list as the final detection result. They also improved the methodology of [50]. The authors proposed a procedure that transforms the outputs of all base classifiers to the form of a Bayesian probability rule. Krawczyk et al. [51] proposed three measures of classifier competence (e.g., minimal difference measure, full competence measure, and entropy measure) for performing dynamic classifier selection. They aimed to prevent the situation of choosing the weakest model.

Anomaly Detection Challenges

An analysis of the previous studies shows 6 challenges. This finding is also supported by [4]. The challenges in the anomaly detection research area are listed as follows:

1. Low anomaly recall rate: Since anomalies rarely occur, it is difficult to detect all types of anomalies. Many normal data points are accidentally detected as anomalies. As a result, the false alarm or false positive rate is increasing.

2. Anomaly detection in high-dimensional and/or nonindependent data: Anomalies always reveal evidence of abnormal characteristics in a low-dimensional space. However, it becomes hidden and unnoticeable in a high-dimensional space. High-dimensional anomaly detection has been a long-standing problem.

3. Data efficiency for learning normality and abnormality: Due to the cost of producing data labels for a large-scale dataset, fully supervised anomaly detection is not practical for daily usage because it requires labeled training data with both normal and anomaly classes. Therefore, in recent years, the research direction has focused on unsupervised anomaly detection that does not require any labeled training data. However, unsupervised methods do not have any prior knowledge of true anomalies. Therefore, the reliability of the detected results remains an issue.

   Moreover, the data patterns change constantly. Therefore, it is necessary to retrain the model. This issue is impacted by all anomaly detection models (e.g., supervised, semisupervised, and unsupervised). It is necessary to have a new model that can adapt to new data patterns and not rely on massive data labels. The interesting studies that addressed this issue are Wu et al. [30], Hundman et al. [18] and Zhou et al. [36]. They also include other studies that address temporal anomaly detection, such as [29, 28].

4. Noise-resilient anomaly detection: Many models assume that the inputted data are clean. In this case, there is a risk that the model is mistakenly labeled for each class. At this point, it is impacted by the accuracy of the detection result. The main challenge is that the amount of noise can be significantly different from the datasets. Moreover, the noisy data points may not be regularly distributed in the data space [4].

5. Detection of complex anomalies: As shown in the reviews section, most of the detection models are for point anomalies that cannot be used for contextual anomalies and collective anomalies since they have completely different behaviors from point anomalies. One main challenge is to include the concept of contextualized and group anomalies into anomaly detection models. Additionally, current methods mainly focus on detecting anomalies from single data sources, while many applications require anomaly detection with multiple heterogeneous data sources [4].

6. Anomaly explanation: When black-box models are used for anomaly detection, the data have labeled either normal data or abnormalities. In this case, assigning data labels can be biased. This is because in some cases, data are rarely assigned as anomalies, but are normal data points. Therefore, it is crucial to have an algorithm to explain the reason or give some clues as to why a data point is assigned as an anomaly. This algorithm can also enhance the detection result to be more accurate and directly address the location of the anomaly in the data.

Unfortunately, the explanation algorithm for anomalies is still under development and is still limited. Most existing anomaly detection studies focus on only accurate detection models. They ignore the capability of explaining the identified anomalies. The main challenge is also to balance the model's detection performance with the explanation.

The literature review motivates us to develop a methodology to fulfill the gaps in these six challenges. In the next section, the review of anomaly explanations, or root-cause analysis in this study, is presented.

2.2.3 Root-Cause Analysis

The root-cause analysis consists of the explanation of the abnormal behavior, that is, the root cause that makes it happen. In the study by [52], the weight of the activation function in a generalized neural network was used to describe the root cause of the detected abnormality from the influence of the factors in a Spark cluster. It could determine the type of abnormalities and the influence of the factors. Root-cause analysis was also employed to analyze alarms occurring in a thermal power plant using the Bayesian network, as shown in [53].

Furthermore, the analysis of false processes in industrial manufacturing using an extreme learning machine (ELM) was adopted and presented in the study of [54]. [55] detected stragglers by Spark speculation and analyzed the root causes by the extracted features. It leveraged the experience rule (rule-based) to extract features for each task from the application log and monitoring data. Recently, Cauteruccio et al. [15] proposed a framework for anomaly detection and root-cause analysis. The authors proposed a distance-based and calculated node arc degree technique to detect anomalies in the IoT.

After an anomaly is detected, in the second stage, forward and inverse problems are introduced. This stage analyzes the cause of the detected anomalies. For instance, given a source IoT node, the connected node is analyzed by the degree of the anomaly. In other words, there is a network and the cause of action is observed, and computing continues until the anomaly node is reached. For the inverse problem, instead of starting with a node connected to the anomaly node, the process starts from the anomaly node and traces back to the origin nodes; it starts from the effect (observable) a continues to the cause of the event (model). From these techniques, the cause of the anomaly is revealed. The author showed that these techniques are not widely used for anomaly detection and root-cause analysis.

2.3 Problem Solving

Refer to the anomaly detection and root-cause analysis mentioned in the previous literature. The theory behind this problem-solving process is the so-called forward problem and inverse problem. These two problems have different methods and objectives for these tasks. For instance, if the author already has a model for detecting anomalies and wants to observe which data point is an anomaly, then the forward problem formulation is applied. However, if the author already observed the anomaly and wants to know what caused it, then the inverse problem formulation is applied. A summary of this problem formulation is shown in Figure 2.3.

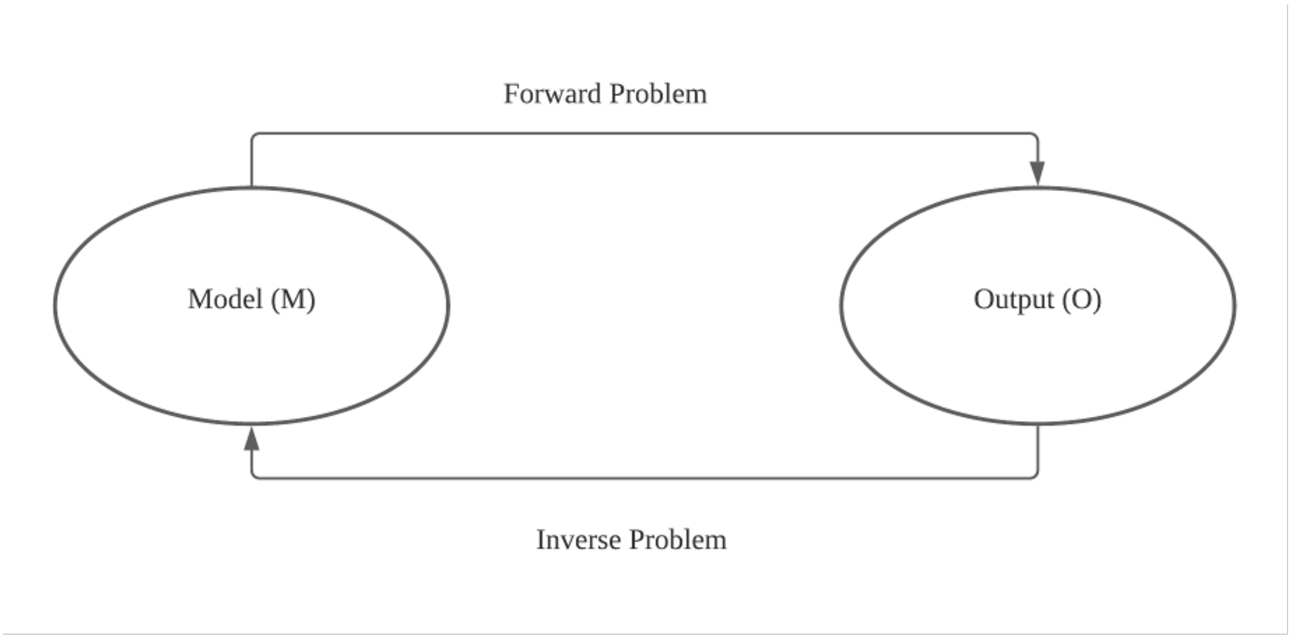


Figure 2.3: Problem formulation

2.3.1 Forward Problem

The forward problem is formulated from the model parameters (m) and sources (s) to observe output (o) as computed in Equation (2.1):

$$o = A_s(m) \tag{2.1}$$

where A_s is the forward problem depending on a source (s). This technique is regular problem solving in which we have a question, and we want to find the answer to that question.

2.3.2 Inverse Problem

In the context of root-cause analysis, the inverse problem is formulated from output (o) to obtain suitable source (s). In other words, this theory attempts to find the suitable source of output (o) [56], as computed in Equation (2.2):

$$s = A_s^{(-1)}(o) \tag{2.2}$$

where $A_s^{(-1)}$ is the inverse problem operator. This assumes that the source parameters are known. Next, in the solution of any inverse problem, there are 3 important questions that should be considered [57].

- Does a solution exist?
- Is it unique?
- Is it stable?

The formula of the solution that exists is related to Equation (2.2). The uniqueness of the solution can be demonstrated as Equation (2.3):

$$A(m_1, s_1) = o_0, A(m_2, s_2) = o_0 \tag{2.3}$$

where m_1 and m_2 are two different models and have two different sources denoted as s_1 and s_2 that produce the same output. In this case, it is difficult to determine the uniqueness of the model. Therefore, uniqueness is crucial for the inverse problem.

The last part is the stability of the solution. The stability is formulated as Equation (2.4):

$$A(m_1, s_1) = o_1, A(m_2, s_2) = o_2 \tag{2.4}$$

where m_1 and m_2 are two different models and have two different sources denoted as s_1 and s_2 that produce different outputs. Assume that the two models and sources are completely different; however, the difference in output is between the noise level ϵ. $|(|\delta m|)| = |(|(|m_1 - m_2|)| > C, |(|\delta s|)| = |(|s_1 - s_2|)| > C, |(|\delta o|)| =)||o_1 - o_2|| < \epsilon, C \gg \epsilon$ where $|(|.|)|$ denotes some norm or measure of difference between two models, sources and outputs.

In this case, it is not possible to distinguish these two models from the observed output. This is the reason why the inverse problem is more complicated than the forward problem. Therefore, the inverse solution is computed based on an approximation, not an exact solution [15, 57].

2.4 Preliminary Work

At the outset of this research, it is crucial to understand the business problem. Thus, a survey was performed at one of the most famous logistics agencies in Thailand. Accordingly, crucial information was acquired about the company's needs and goals in terms of planning for vehicle route optimization to deliver goods to customers. It also included how they monitored these processes when they delivered goods to customers. This study collaborated and exchanged ideas with an employee representative in the logistics management division. Therefore, a better understanding of the business was obtained and used for this study.

Then, the business problem was modeled as an academic problem. The historical data were analyzed, and the author discovered that it is necessary to understand the vehicle route optimization processes' characteristics to help in decision making. Therefore, to evaluate the efficiency of operations, the author sought to understand the data pattern. Accordingly, the author performed a behavior analysis of large-scale data. Subsequently, previous studies were reviewed to determine a solution that could deal with the multidimensional data involved in this research.

However, this review elucidated that most methods cannot be applied directly to monitor the vehicle route optimization process because most of the previous works were based on detecting point disturbances and lacked an explanation of the detected result. Unfortunately, the logistics management data are correlated to each other and are dependent on the location–time context and event behavior (e.g., vehicle usage, a cycle of vehicle maintenance, absence of human resources, or incidents in order cancellation and postponement). Therefore, if the previous methodologies are applied to these data, it can lead to incorrect decision making.

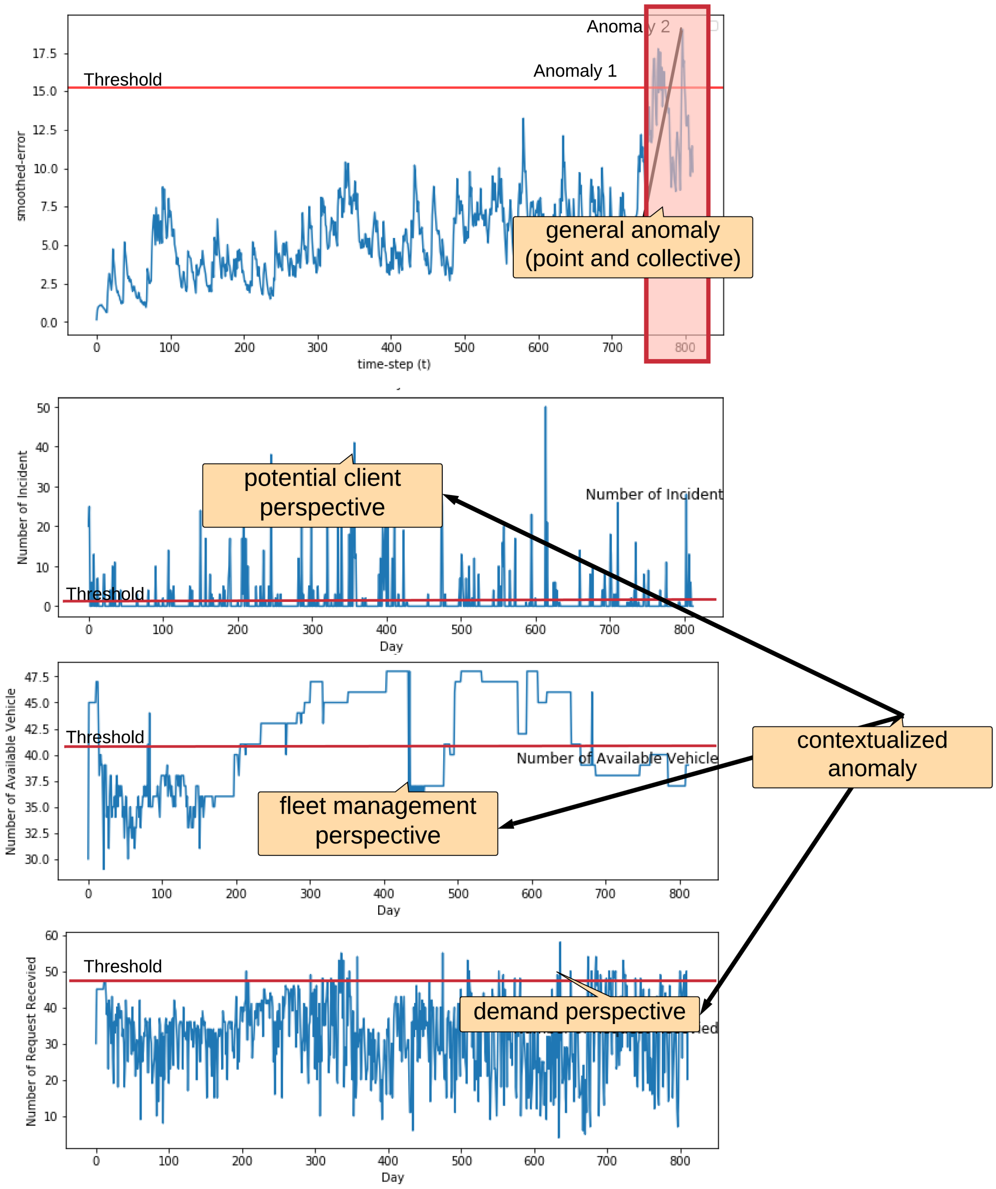


Figure 2.4: The types of anomalies in transportation logistics.

Figure 2.4 demonstrates the different types of anomalies that are involved in the logistics agency workflow. In addition to the general type of anomalies (e.g., point and collective), the anomalies in the transportation domain are also classified as “contextual anomalies” and considered in 3 perspectives. The first is the fleet management perspective. This kind of anomaly occurs when there is numerous vehicle shortage than usual because of the maintenance cycle or absent of drivers. It also includes the overused of vehicles from the vehicle route assignment. This phenomenon is considered as abnormal when compared to the routine vehicle’s fleet status.

The second is the demand perspective. This anomaly occurs when there is an unusual increase in demand from customer input into the system.

The third is from the potential client perspective. This anomaly occurs when there is an unusual surge in order cancellation and postponement from potential customers.

Figure 2.5 demonstrates the Andrews curves for transportation logistics data. The Andrews curves present a feature space of high-dimensional data. It is based on the Fourier series and provides information about the dataset structure. Each curve shows an observation (e.g., representing each row in the data record) in the dataset. For more detail about the Andrews curves, please refer to [58, 59].

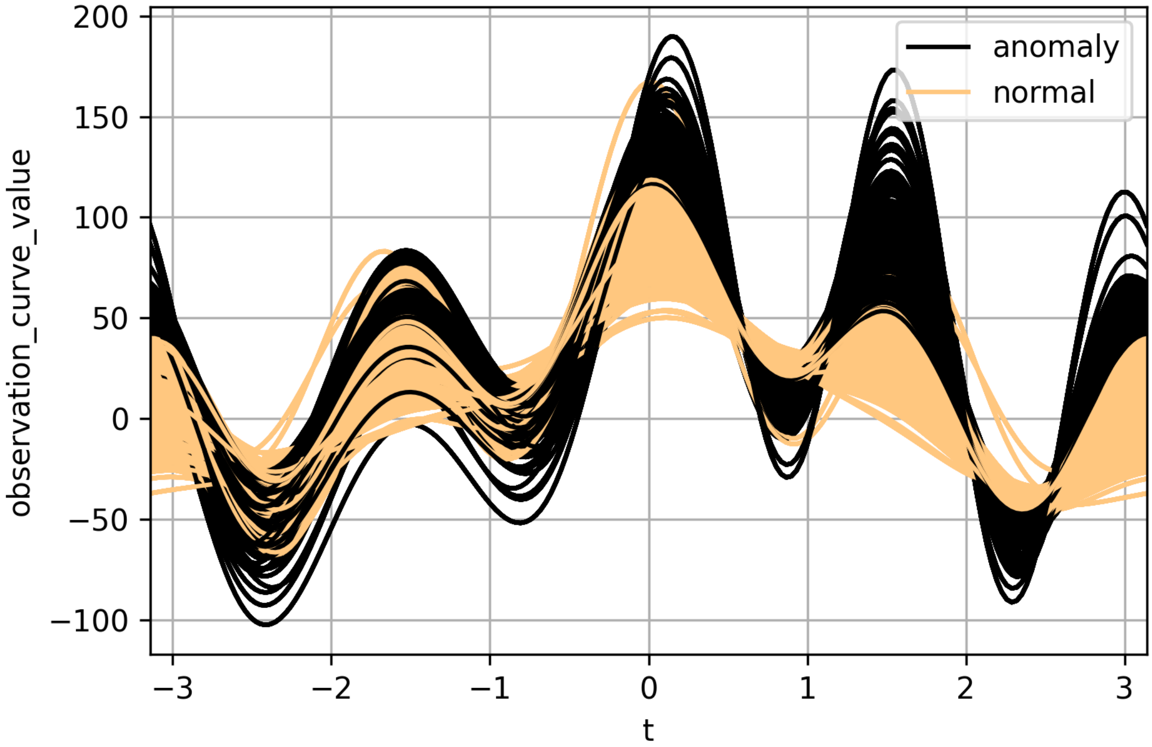


Figure 2.5: Demonstration of the separability between anomalous (black) and normal (copper) events in the transportation logistics dataset.

Figure 2.5 demonstrates that the two events in the dataset are not clearly separable, as shown by the curves that intersect with each other. This indicates the high degree of inherent nonlinearity in the feature space. Therefore, anomaly detection from conventional ML approaches cannot be used directly in this case.

This problem motivated the author to propose a novel methodology to fill the gaps in previous studies. It is evident that in this research domain, significant improvements can be made to support the development of logistics management.

2.5 Remaining Question

Refer to the problem statement. Generally, logistics agencies use these data to support their decision making, such as transportation planning, human and fleet allocation, production, and manufacturing tasks. In addition, they also want to understand the transport operating characteristics of the data that flow into the system.

Unfortunately, it was mentioned that the logistics management data come from multiple sources. They are also conditionally dependent. Therefore, analysis cannot be performed manually by a staffer or using black-box software to perform these tasks. As a result, the remaining question is, what kind of methodology should be used to support logistics agencies in transportation planning?

2.6 Restatement of Research Question

The related work shows that anomaly detection and root-cause analysis are required to assist the logistics agency in better understanding the transportation planning situation, especially vehicle route optimization tasks, from the data. Performing anomaly detection and root-cause analysis aims to distinguish normal data from anomaly data effectively. As a result, the anomalies are revealed making it feasible to eliminate them before they cause a critical problem in transportation planning and directly impact the route optimization process.

However, anomaly detection methodologies have some limitations in dealing with high-dimensional and conditional data. As a result, it raises a reliability issue for the detection result. Furthermore, they can only detect anomalies. Unfortunately, the explanation of the cause of the anomaly that can be traced back to its origin anomaly is still under development. Therefore, it is not easy to prevent anomalies before they occur. Anomalies can only be detected after they have occurred. Last, the lack of explanation of anomalies can cause the data to be labeled based on bias. This limitation also increases the chance of incorrectly labeling anomalies as normal and vice versa.

Therefore, the remaining questions are restated as “What kind of anomaly detection method is suitable for dealing with high-dimensional and condition-dependent data?” and “How can the cause of the anomalies be explained?” If there are no suitable models to detect the anomaly, then what is the appropriate solution?

2.7 Limitation and Delimitation

According to the problem statement and challenges in anomaly detection and root-cause analysis, two limitations to this study exist. First, the study has limitations in automatically dealing with unstructured data, such as text reports and spreadsheets from different types of data formats. Therefore, more preprocessing steps are required to format the data to be ready before analysis. Second, the noise in the data remains; however, it remains at a low percentage. Therefore, it results in false alarms in the detection result.

The delimitation of this study includes monitoring and detecting anomalies in transportation planning. Thus, this study only considers the transportation planning task, especially the route optimization process. However, this is also somehow linked to manufacturing and production tasks. These tasks use a multicriteria study on selected suitable preferences to adjust for the route optimization model.

However, multicriteria decision making in manufacturing and production is beyond the scope of this research. Instead, the scope of this study is limited to understanding vehicle route optimization behavior and eliminating the actions that degrade the transportation planning solution. The author plans to explore the multicriteria decision-making domain to assist in this anomaly detection task in future work.

Chapter 3

Detecting Abnormal Behavior in the Transportation Planning using Long Short-Term Memories and a Contextualized Dynamic Threshold

3.1 Introduction

This chapter presents a methodology for detecting temporal and conditional anomalies from time-series data. As mentioned in Chapter 2, the gaps and challenges are essential to detect anomalous dependence on conditional attributes. The conditional attributes in this context are the spatiotemporal (e.g., location and time) and behavioral attributes (e.g., usage, speed, and incident statistics) that were performed in the environment. These actions are also recorded in the data. Unfortunately, general models mainly designed to detect point anomalies cannot be used for this purpose [3, 4]. Moreover, when general models are used (e.g., LSTM or LSTM with thresholds), they cause a false alarm or an increase the false-positive rate [18].

Furthermore, point-type anomalies seem to be a simple type of anomaly that depends on distance- and density-based data points. They are easy to detect using time-series analysis or a clustering-based approach. However, the logistics anomalies do not contain only point anomalies. As mentioned in Chapter 2, besides the general type of anomalies (e.g., point and collective), the anomalies in the transportation domain are considered in 3 perspectives (e.g., fleet management, demand, and potential client). Based on the principle of anomaly detection domain, anomalies in transportation are defined as "contextual anomaly." Nevertheless, instead, they are heavily related to conditional and occurrence attributes. It is because the data come from multisources. Therefore, each attribute relies on the other. In this problem, it is necessary to have a new methodology that can simultaneously detect various types of anomalies. As a result, the detection result is enhanced and more accurate than using previous approaches.

In this chapter, a novel anomaly detection model for detecting temporal and conditional anomalies is proposed. This model contributes to [18]'s study. The contextualized dynamic thresholds are added to analyze the time-series data. The aims of adding contextualized thresholds are to detect conditional anomalies, which are related to various contexts. For instance: time, location, and frequency. The second purpose is to reduce the false-positive rate. These procedures also support the review paper's findings in [3, 4]. The authors mentioned that adding conditions to analyzing the data can enable the model to detect more complicated anomalies.

These anomalies are related to conditional and behavior attributes.

The author has an observation for the study in Chapter 3 as follows. Suppose the prediction error from the forecasting model cannot efficiently represent anomalies from the time-series data. It implies that the anomalies are concordant to external real-world elements, and contextualization is required to support the analysis.

If this observation is valid, when the author adds a method for analyzing the environmental conditions will enhance the model's detection efficiency and improve the detection rate.

The procedures of the proposed anomaly detection model of this chapter consist of two stages for detecting anomalies from time-series data. In the first stage, the author used a forecasting model to detect any data points that exceed the defined threshold. This step analyzed the prediction error of the Long Short-Term Memory (LSTM) model using the 3-sigma threshold method. The outcome from this analysis is normal data points, points, and collective anomalies.

Unfortunately, using the 3-sigma threshold method for analysis also has a disadvantage. For example, suppose that the anomalies are dependent on environmental conditions. Referring from the definition, this kind of anomaly is so-called contextual anomaly. Then, the prediction error approach might not be practical to detect such anomaly. Similarly, the issue was also discovered in [49]. Therefore, the author resolved this problem by adding the second stage analysis, which defines the thresholds for context and behavior attributes of the data.

The context consists of time, and behavior attributes consist of vehicle availability, incident, and customer demand. This second stage analysis was done separately from the first step analysis. Therefore, for detecting anomalies from the input time-series data, the model detects anomalies from the prediction error together with the conditional contexts. As a result, the efficiency of the model is significantly improved, as shown in the results and discussion section. The author believed that the ability to detecting anomalies for all aspects of the real-world elements could be enhanced by doing so. In the last part of this chapter, the practical applications and limitations of the proposed model are presented. The author aims to demonstrate how general the proposed model was for expanding to other application domains. For further details, they are presented in the "Methodology" section.

3.2 Methodology

3.2.1 Data Collection and Pre-processing

First, the author collected data from GPS tracker equipment and agencies' operational reports. The collected data from these two modules consists of a GPS probe from sixty trucks with installed GPS tracker equipment and a client order. Note that the data collection period is 3 years (from the year 2017 - 2019). Then, the author divided it into four data formats. These data formats consisting of order confirmation, vehicle statistic, driver statistic, and order. These datasets were used to preprocessing and feature engineering using a business intelligence framework, as the data come from multi-sources. At this stage, the data were standardized and reshaped to match with ML prediction model requirements.

Second, to acquire all necessary features, a data acquisition is performed to extract features and prepare for analysis. The author then loaded the data to store it in data storage. The author used a Power BI to determine the data relationships. These relationships were used for measuring the fundamental statistic of each feature. They also created a linkage to other data attributes within various perspectives, as shown in Figure 3.2. Power BI is a software used to discover relationships in multi-sources data. It also helps us to understand multi-sources data coherency and can be used to perform data visualization tasks.

The statistics are obtained from the previously described operations. The described operations consist of the number of vehicles used, available vehicles, drivers, relationship between vehicle and assigned driver, including their orders, and more. These statistics are shown in the form of multivariate time-series, as in Table 3.1 and Figure 3.3.

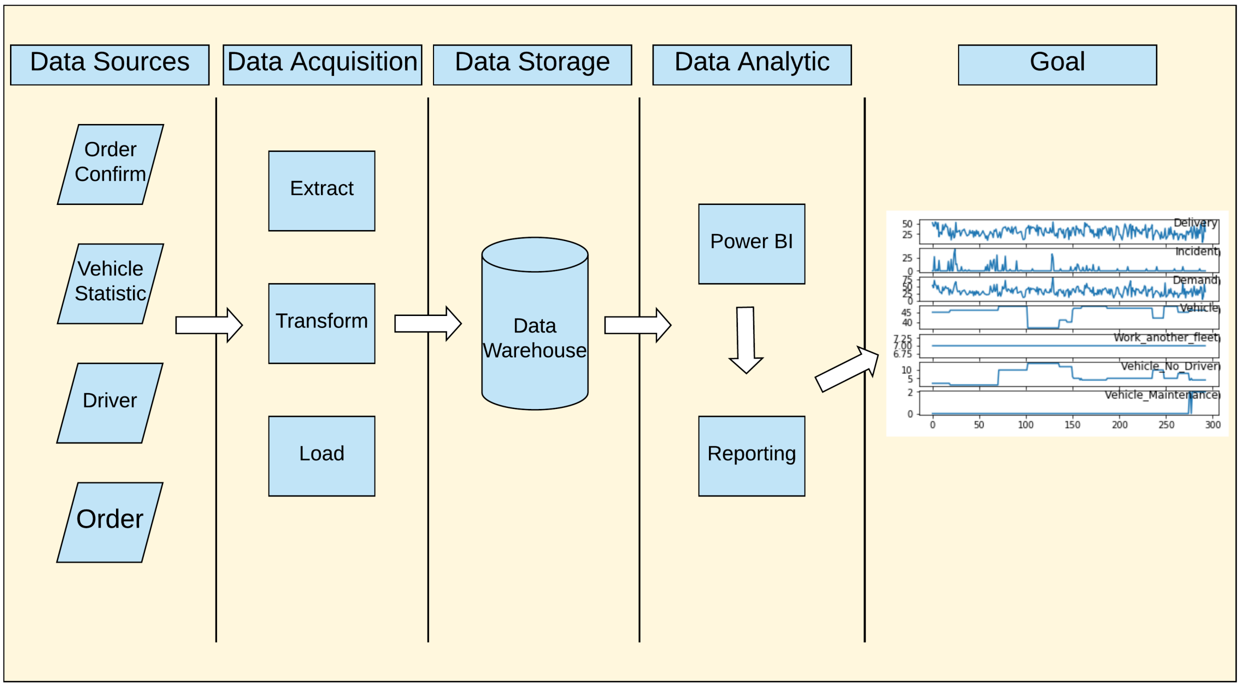


Figure 3.1: A framework of business intelligence

Table 3.1: Data specification used in this study.

Attribute(s)	Value	Unit
Date	2/10/2018	-
Number of available vehicles	45	Vehicles
Number of occupied vehicles	7	Vehicles
Number of vehicles with no assigned driver	8	Vehicles
Number of vehicles with back order work	0	Vehicles
Number of vehicles in maintenance	0	Vehicles
Number of vehicles with driver taking leave	0	Vehicles
Number of total requested from client	45	Orders
Number of requests received	45	Orders
Number of orders canceled or postponed (import)	0	Orders
Number of orders canceled or postponed (export)	0	Orders
Quarter of the year	Q4	-

From Table 3.1, the author described the definition of import and export as follows:

- Imports are tasks that deliver goods or products from local agencies to overseas destinations.

- Exports are tasks that deliver goods or products from overseas to local destinations.

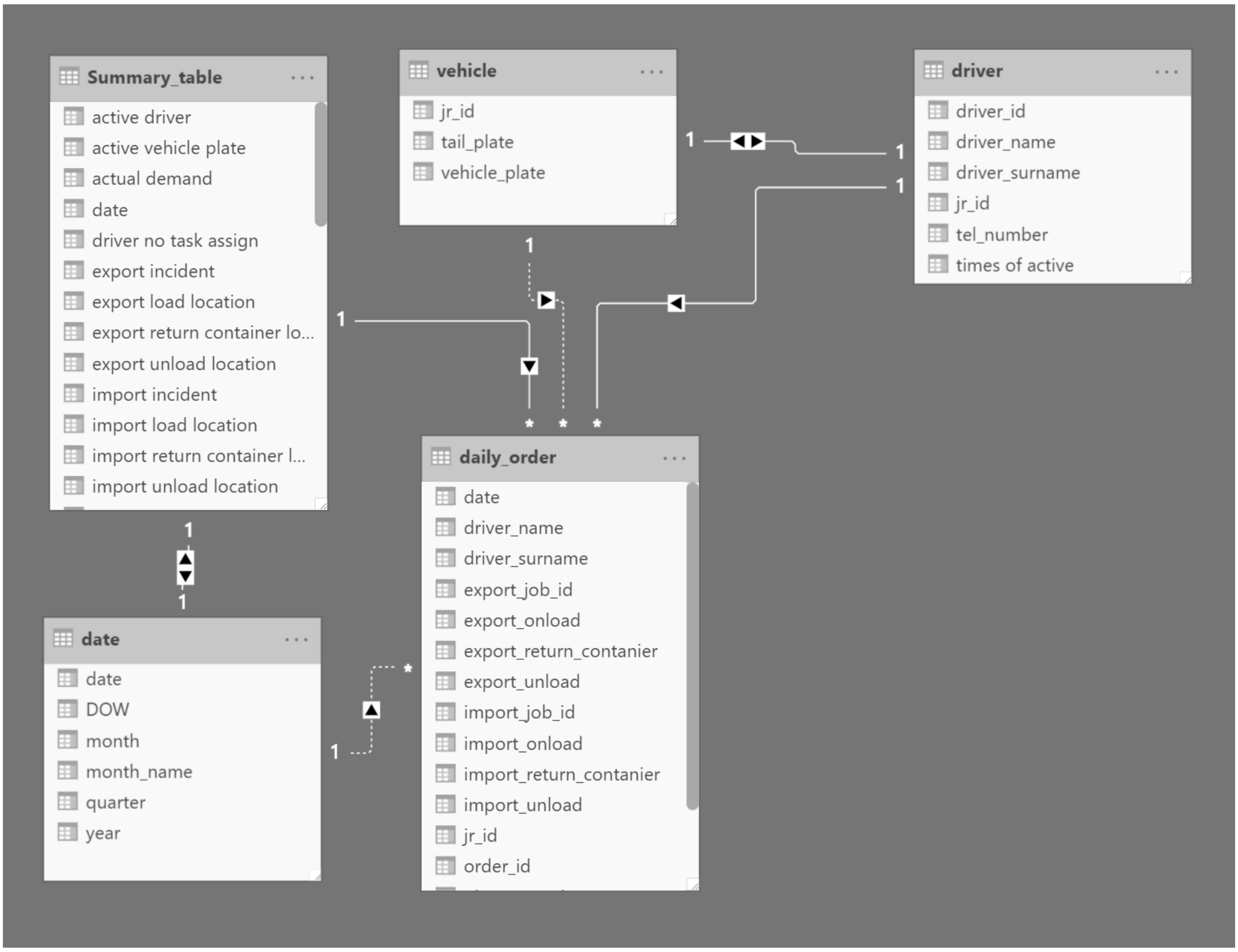


Figure 3.2: Demonstration of data relationship from multi-sources data.

In addition, the variables relationships of the collected data are shown in Figure 3.4.

After obtained all necessary features, the data have also been organized into groups to train and test the models presented in section 3.2.3. Note that the author separated the input data (e.g., train and test) into two subsets. The first data subset was for input into the Long Short-Term Memory (LSTM). The second data subset was used for analyzing the behavior attributes (e.g., vehicle availability, incidents, and customer demand) of the contextual anomalies. The required data attributes for the second data subset were the number of available vehicles, orders canceled or postponed, and total requested from customers. To analyzing these attributes, contextualization is required. Therefore, the contextualization process is presented in the next section.

3.2.2 Preliminary Experiment on Anomalies Contextualization

From [7]'s study, the authors showed that the contextual anomalies required a context link with a behavior for detection. Therefore, this finding motivates us to conduct this experiment further. The author discovered that the method that can fulfill this task was the Bayesian network. The Bayesian network can define the linkage between data attributes. It is among the type of methods in the graphical model used to represent the dependency between the event and evidence in the dataset. The model formulation is described in section 3.2.2. The assumption is that suppose the linkage has a low probability, then that data connection is rarely

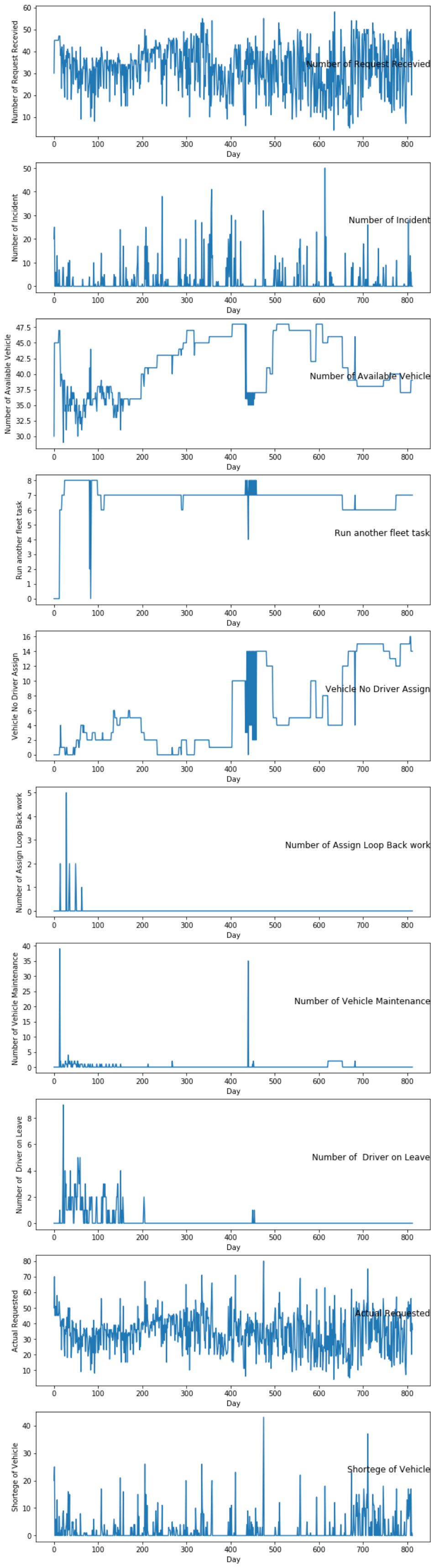


Figure 3.3: Visualization of collected data.

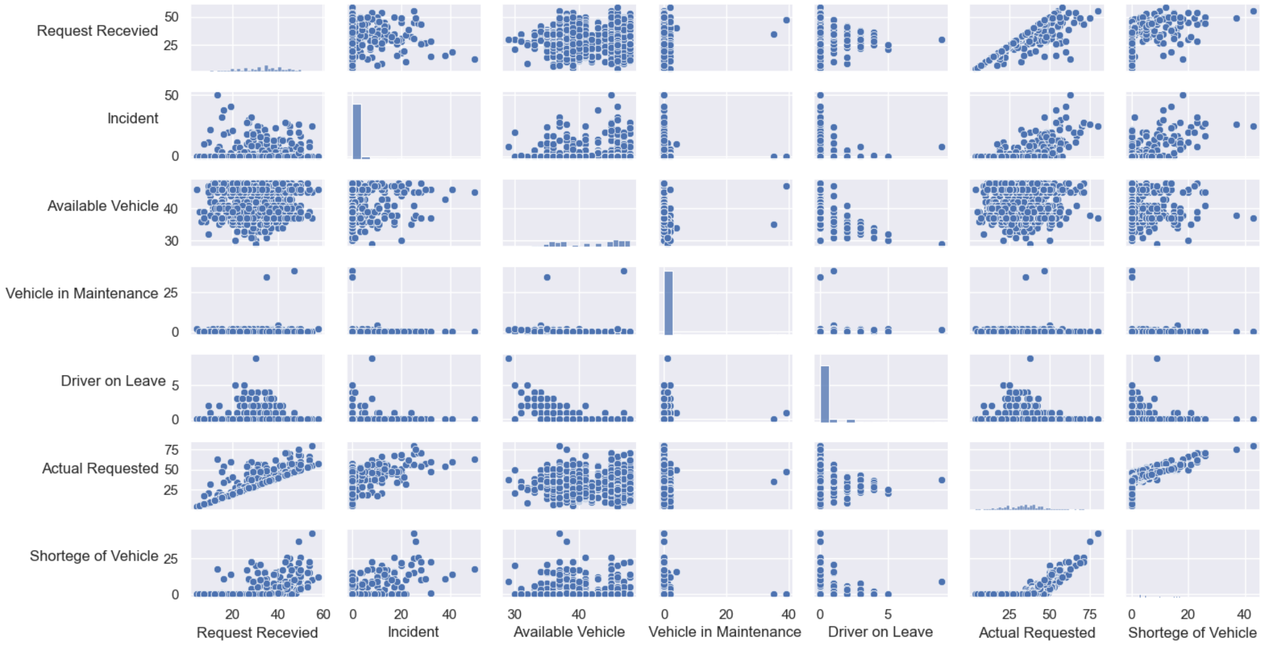


Figure 3.4: Demonstration of the variables relationships from multi-sources data.

occurring. Thus, the origin and destination of that linkage are used to be the context and the threshold. Therefore, the anomalies are defined by this context and threshold.

Define data attribute dependency using Bayesian Network

After the data was obtained, the dependencies and set of rules are determined to define anomalies from the dataset. The author used Bayesian Network-Based Approaches for anomaly contextualization as in Equation (3.1):

$$\Pr(e|m) \tag{3.1}$$

where e is an event (or evidence for an event) and m is the model. To determine the anomaly's context, the threshold (t) is needed to specify, as represented in Equation (3.2):

$$\Pr(e|m) < t \rightarrow anomalous \tag{3.2}$$

For time-series or data that have a sequence of events, a process to contextualize the anomaly (aggregated) is required. The equation is modified based on [60] as Equation (3.3):

$$\frac{1}{N}\sum_{i}\Pr(e|m) < t \rightarrow anomalous \tag{3.3}$$

where N is a time-step i and m is the model. Suppose that we would like to determine conflicts within a set of evidence, the "conflict measure" [60] is used to detect possible non-coherence in evidence $\mathbf{E} = \{E_1 = e_1, ..., E_m = e_m\}$ as in Equation (3.4):

$$C(\mathbf{E}) = \log\frac{\Pr(E_1 = e_1) \times \cdots \times \Pr(E_m = e_m)}{\Pr(\mathbf{E})} \tag{3.4}$$

After using the Bayesian Network, the context and behavior for the defined anomalies are obtained. The context and behavior are set based on the low probability of the linkage and defined as Equations (3.5) – (3.7). For more detail about the benefit of these contexts, the author provided further discussion in the results section.

For the condition, vehicle (V) usage has exceeded the threshold of a vehicle usage range. This threshold denotes the $threshold_v$. It is computed as Equation (3.5):

$$y_1 = \begin{cases} 1, & \text{if } V > threshold_v \\ 0, & \text{otherwise} \end{cases} \tag{3.5}$$

The condition is activated as anomalies when the value of the received request (R) exceeds a vehicle usage threshold. This threshold denotes the $threshold_v$ for vehicle usage. It is computed as Equation (3.6):

$$y_2 = \begin{cases} 1, & \text{if } R > threshold_v \\ 0, & \text{otherwise} \end{cases} \tag{3.6}$$

The condition is activated as anomalies when the incident (I) is more than 0 and exceeds the incident threshold. It is computed as Equation (3.7):

$$y_3 = \begin{cases} 1, & \text{if } I > threshold_I \\ 0, & \text{otherwise} \end{cases} \tag{3.7}$$

If any of these conditions are satisfied, then contextual anomalies have occurred during the operational process. From these contexts, the definition to define possible anomaly and not-anomaly points are grouped, as shown in Table 3.3.

After the contexts were defined for detecting contextual anomalies, now ready to develop the anomaly detection model, the development is presented in section 3.2.3.

3.2.3 Proposed Model for Anomaly Detection

The author used the proposed novel unsupervised anomaly detection method to determine and detect a fleet management transaction. It was used to check whether or not it results in any critical errors. It also included the defined contextualized threshold that represents an anomaly's operation from the multivariate time-series data. This study was motivated by a previous study's challenge. It consists of addressing temporal dependency data and reducing the false-positive rate. From [18]'s study, the author discovered that a dynamic 3-sigma threshold for analyzing prediction error of LSTM was able to detect the majority of the anomalies. Unfortunately, the false-positive rate still remained high. The model was efficient in detecting some anomalies that can be represented by the prediction error but are not capable of other variants such as conditional anomalies. This kind of problem was also discovered by [9]. Furthermore, numerous environmental factors can also influence the creation of anomalies. Hence, the LSTM with a dynamic threshold cannot detect all types of anomalies in the system.

To solve aforementioned issues, the contextualization thresholds are proposed for the model. It increases the capability of detecting specific anomalies from a high-level perspective of an urban transportation logistics operation. These anomalies could not determine by a prediction error and statistical threshold. However, they depend on context and behavior attributes, as mention earlier. Therefore, conditional attributes' thresholds are required in this case. For example, anomalies can be revealed from the client(customer), operation, asset, and human resource contexts. Furthermore, these contexts helped to identify the cause of the anomalies in the initial step. Thus, the proposed model is shown in Figure 3.5.

From this point, the elements shown in Figure 3.5 are presented in detail in the following sections.

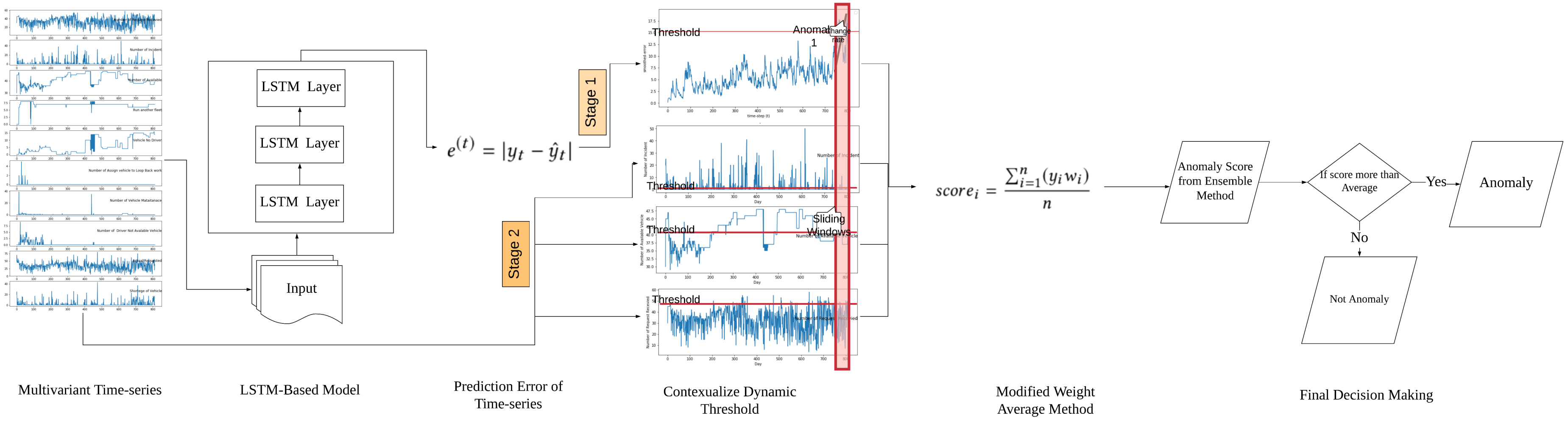

Multivariant Time-series
LSTM-Based Model
LSTM Layer
LSTM Layer
LSTM Layer
Input
Prediction Error of Time-series
$e^{(t)} = |y_t - \hat{y}_t|$
Stage 1
Stage 2
Contexualize Dynamic Threshold
Threshold
Threshold
Threshold
Threshold
Anomaly 1
Change rate
Sliding Windows
Modified Weight Average Method
$score_i = \frac{\sum_{i=1}^{n}(y_i w_i)}{n}$
Final Decision Making
Anomaly Score from Ensemble Method
If score more than Average
Yes
No
Anomaly
Not Anomaly

Figure 3.5: Demonstration on the flow of data which input into the model, first it is determined the temporal dependencies and prediction error from LSTM. Second, it evaluates the prediction error and supported contexts with the proposed contextualize thresholds. The evaluation results were then combined with the weight average method. Finally, the final decision for anomaly detection was returned.

3.2.4 Multivariate Time-series Prediction using LSTM

At this stage, the data were taken from section 3.2.1 as inputs. The fundamental principle of a recurrent neural network (RNN) is to predict the following information's input. The author used the LSTM to train the sequences of the time-series data. However, in a conventional neural network, inputs and outputs are independent of each other. Therefore, when the LSTM made a prediction, it is important to recognize the previous time steps. This type of neural network is called recurrent. The same computation is performed for all elements in a sequence of inputs, and the output of each element depends on stored state data [61].

The basic principle is to improve the network by providing it with a clear memory. The special hidden units are also equipped in these frameworks. The resultant behavior is that prior input can be memorized for a long time. Therefore, the author used multivariate time-series forecasting with LSTM to predict the received request from the customer and vehicle availability in the fleet. The author evaluated the model's root mean squared error (RMSE) and the absolute error, computed as Equation (3.8). The prediction reflected the temporal dependencies based on the observation as follows. If the time sequences have a high error rate of prediction, then it denotes that the new time-step did not regularly occur. The author assumes that it did not have any learning patterns from the prior time-step attributes to predict new data. As a result, the model makes a wrong prediction.

$$e^{(t)} = |y_t - \hat{y}_t| \tag{3.8}$$

where e^t represents the prediction error of each time-step, y_t represents a true observation, and $\hat{y}_t$ represents the prediction result from the input features.

Once the LSTM model is constructed, the thresholds for analyzing the output of the LSTM and environment information are presented in the next section. These thresholds are used to analyze the LSTM prediction error for detecting point and collective anomalies together with contexts and behavior attributes for detecting contextual anomalies.

3.2.5 Contextualized Dynamic Threshold

After the prediction result was obtained, in the first stage, the dynamic threshold approached proposed by [18] was utilized and modified for use in this chapter. The prediction error in each time-step represent one-dimensional vector errors:

$$e = [e^{t-h}, ..., e^{(t-l_s)}, ..., e^{(t-1)}, e^{(t)}]$$

where h is the historical error of the prior time step. The author smoothed the set of errors e. The author aims to reduce the spike error generated from the LSTM model. In some cases, the LSTM model was not correctly predicted when the not-abnormal state data point was inputted, as shown in the [18] experiment. The exponentially weighted average (EWMA) is used to generate smoother errors. Each data point's weight was determined from the prior time-step $(t-1)$. The smoothed error is then obtained:

$$e = [e_s^{t-h}, ..., e_s^{(t-l_s)}, ..., e_s^{(t-1)}, e_s^{(t)}]$$

To evaluate whether or not that the values are non-anomaly, the threshold for smoothed prediction errors is set using 3-sigma principle. The smoothed errors' values above the threshold were classified as anomalies. In this chapter, the proposed thresholds defined as the context from section 3.2.2 and the smoothed error described previously were used to fill the gap of specific anomalies. These specific anomalies required a context of behavior to make a detection.

Therefore, the author proposes an unsupervised model that does not require any data labels. First, the $threshold_p$ (smooth error threshold) was defined as Equation (3.9):

$$threshold_p = \mu(e_s) + \mathbf{z}\sigma(e_s) \tag{3.9}$$

where $\mathbf{z}$ represents a positive value of standard deviation above $\mu(e_s)$. From the experiment, the author discovers that the number of $\mathbf{z}$ greater than two was increasing the detection rate. It also reduces the number of positive errors.

In the second stage, the author defined thresholds for context and behavior attributes of the data. This stage was done separately from the first stage. Therefore, the author defined the threshold for an incident context in each time step (t), as Equation (3.10). The author used this threshold to determine the period that the number of incidents exceeds the regular transaction in transportation logistics operations.

$$threshold_I = \mu(i) + \mathbf{z}\sigma(i) \tag{3.10}$$

where $\mathbf{z}$ represents the positive values of standard deviation above $\mu(i)$. Furthermore, the author defined the threshold for the vehicle usage context in each time step (t), as Equation (3.11). The author used this threshold to determine the period that several vehicle usages exceed the regular transaction in the transportation logistics operation. From the defined thresholds, the decisions of defining an anomaly are computed as Equations (3.5), (3.6), and (3.7).

$$threshold_v = \mu(v) \pm \mathbf{z}\sigma(v) \tag{3.11}$$

Finally, the author utilized Equations (3.5), (3.6), (3.7) and (3.9) to determine the anomaly score. The anomaly score was computed from the average outcome from those thresholds. A weighted average method [62] was used and computed as Equation (3.12):

$$score_i = \frac{\sum_{i=1}^{n}(y_i w_i)}{n} \tag{3.12}$$

where n represents the number of contexts for a specific anomaly, w represents the assigned weight for each context and y is the decision variable in each context represented in Equations (3.5), (3.6), and (3.7). Note that the author assigned the weight equal to one for this chapter because each context was equally important. Suppose that the score exceeded the threshold, then the data point is classified as an anomaly.

A sliding window approach is also applied. It was used to make a detection in the series of anomalies that occurred in the time-sequence. It was based on the observation as follows. Suppose that the majority of data points in the windows have a $score_i$ exceeding the average score of all data points, then the author classified all of the points as anomalies. In the experiment section, the author will present how the detection performance impacted the window's size. In addition, an experiment is also conducted on the dependencies of the previous data points. The author has an observation that previous data points can influence the current data to become anomalies. Therefore, the different percentage (d) of $score_i$ and $score_{i-1}$ was computed as Equation (3.13):

$$d_i = \frac{|score_i - score_{i-1}|}{score_i} \tag{3.13}$$

where i is the time-step, suppose d_i exceeded the average of all sets of d_i in the specific windows, then all of the data points are classified as anomalies.

From the statement above, the procedure's summary is shown in Figure 3.6. The author first slides the window from the beginning of the series. The anomaly score is then calculated as in Equation (3.12). This equation was supported by threshold and decision conditions. These threshold and decision conditions were previously described. Then, the change between the time-step (t) and the previous time-step ($t-1$) is determined. It assessed whether or not each factor's values violate the threshold. After that, an anomaly score was assigned to each data point in the windows. The procedure is repeated until the end of the time sequence. The outcome of this process is to reveal the anomalies and normal data points in the time sequence.

In addition, the author also updated the model every time that the sliding windows approach was performed, as Equation (3.14):

$$Ref_t = \begin{cases} Ref_{t-1} - obv_{new-w} + obv_{new}, & \text{if } e^{w}_{obv_{new}} > threshold_p \\ Ref_{t-1} + obv_{new}, & \text{otherwise} \end{cases} \tag{3.14}$$

where Ref_t is a reference group of data at time (t), w is window size, and obv_{new} is newly observed data that streamed into the system. Thus, the previous model has transferred knowledge combined with new information to train the new model.

In this section, the process and methods of performing the anomaly detection were presented. After the detection result is obtained, it is time to evaluates the model efficiency using the evaluation metrics. These metrics are presented in section 3.2.6.

3.2.6 Evaluation Metrics

In this section, the detection capability of the proposed model is evaluated using an Area Under the Roc Curve (AUC), precision, and recall. The evaluation also including F_1-score. The author used these metrics due to the unbalanced data (e.g., with numerous percentage of actual negatives, such as data with normal and abnormal are not equally distributed). The accuracy metric is not feasible to evaluate in such a case. The AUC measures the whole 2-dimensional area under the whole receiver operating characteristic curve (ROC). The ROC was computed by applying a different threshold comparing True Positive Rate (TPR) and False Positive Rate (FPR). The TPR and FPR are computed as Equations (3.15) – (3.16).

$$TPR = \frac{TP}{TP + FN} \tag{3.15}$$

where TP represents the number of true positives and FN represents the number of false negatives.

$$FPR = \frac{FP}{FP + TN} \tag{3.16}$$

where FP represents the number of false positives and TN represents the number of true negatives.

The author also validated the experimental result using ground truth data. It was validated in terms of its detection accuracy, precision, and recall. The precision and recall evaluation metrics are computed as Equations (3.17) – (3.18).

$$Precision = \frac{TP}{TP + FP} \tag{3.17}$$

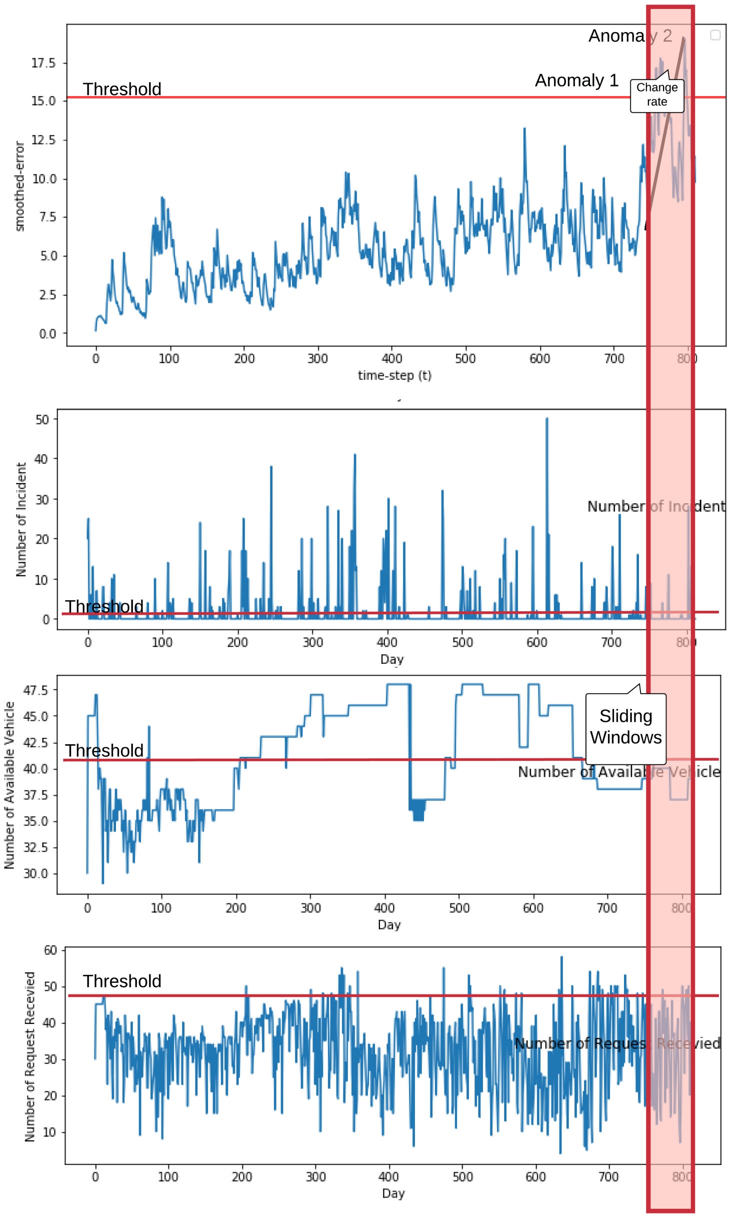


Figure 3.6: Demonstration of the sliding window with contextualized dynamic threshold processes.

where TP represents the number of true positive and FP represents the number of false positives.

$$Recall = \frac{TP}{TP + FN} \tag{3.18}$$

where TP represents the number of true positives and FN represents the number of false negatives.

Once the anomaly detection model for the transportation domain was developed, the author also wants to demonstrate how the developed model can be extended to other application domains. Therefore, the procedures for doing so are presented in section 3.2.7.

3.2.7 Practical Applications and Limitations

In this chapter, the author also used other application data (e.g., credit card transactions, water manufacturing system transactions, and computer network) to performed anomaly detection. Anomalies in this context are the system transactions that were different from the routine usage of the user (e.g., high amount of usage than usual or different period of usage).

Besides, the data visualization (e.g., variables relationships and dependencies) of the credit card, water manufacturing system, and computer network transactions are shown in Figures 3.7 - 3.9.

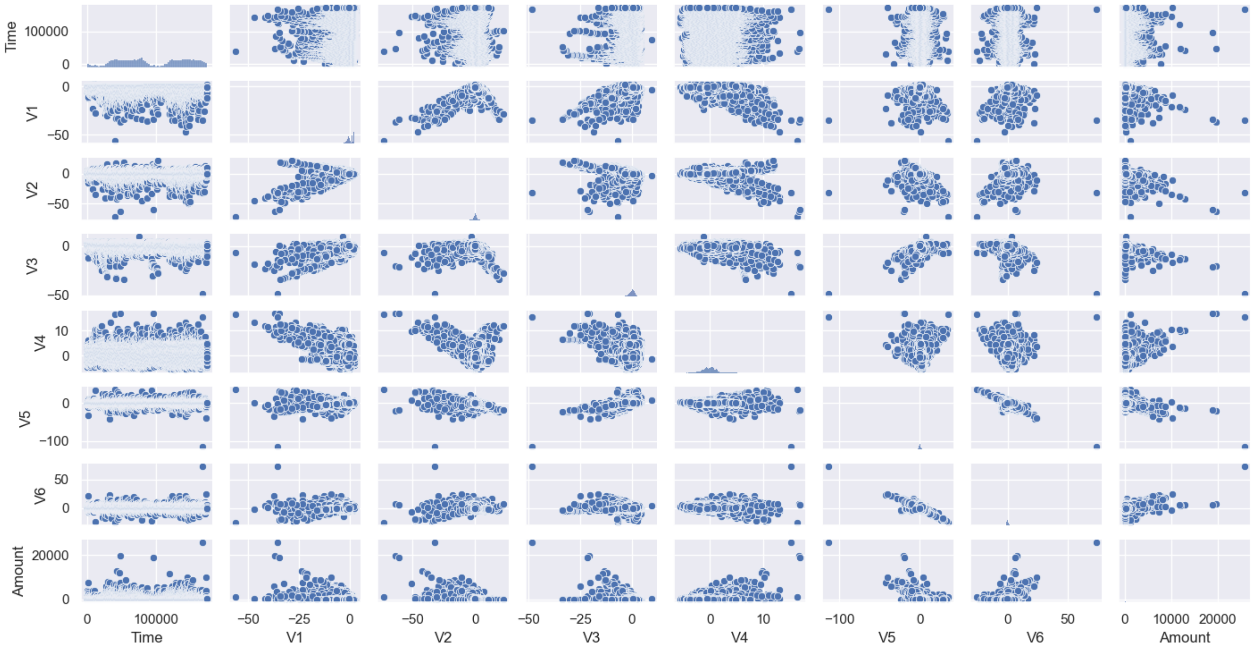


Figure 3.7: Demonstration of the variables relationships of the credit card dataset.

In the beginning, the proposed model was designed for the transportation domain. However, the author also wants to demonstrate the generic of the developed model that can extend to other application domains. If one wants to apply the proposed model with other applications, some minor modifications are needed.

First, the input data is preprocessed and standardized. It is also required to order the data record by time. The essential data attributes that are required are date-time and usage from the targeted system.

Second, the feature engineering process is required before running the model—for instance, the amount and the frequency of usage ratio. These features are used to define conditional thresholds for detecting contextual anomalies. If more features are to be input into the model, the LSTM input layer should be adjusted to meet with the new data. Also, the hyperparameters (e.g., epochs, batch size, learning rate, and more) should be optimized concordant to the new data input.

Finally, for the proposed model's restriction and limitation, the model can handle the input data ordered by time. Hence, the common attributes that are vital for this proposed model are the time context and behavior attributes. The behavior attributes consist of usage, frequency of transaction. In addition, suppose that the data comes in the form of unstructured data; the data should be preprocessing before inputting into the model. It is because the model support only structures data.

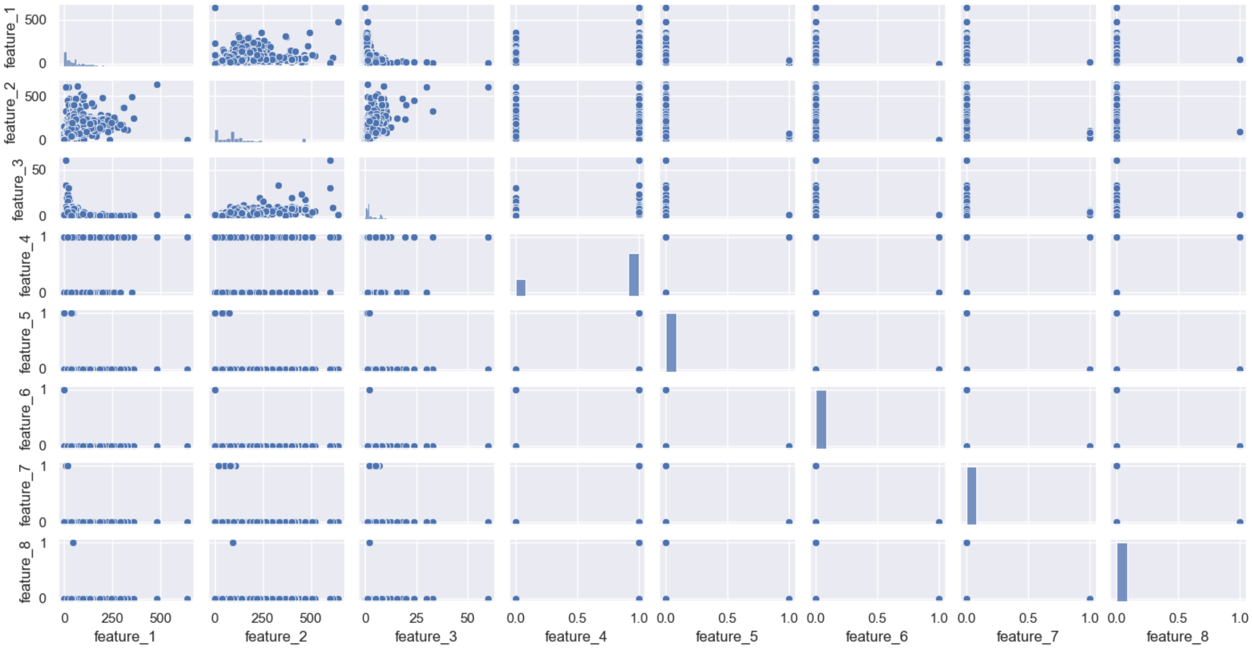


Figure 3.8: Demonstration of the variables relationships of the water manufacturing system dataset.

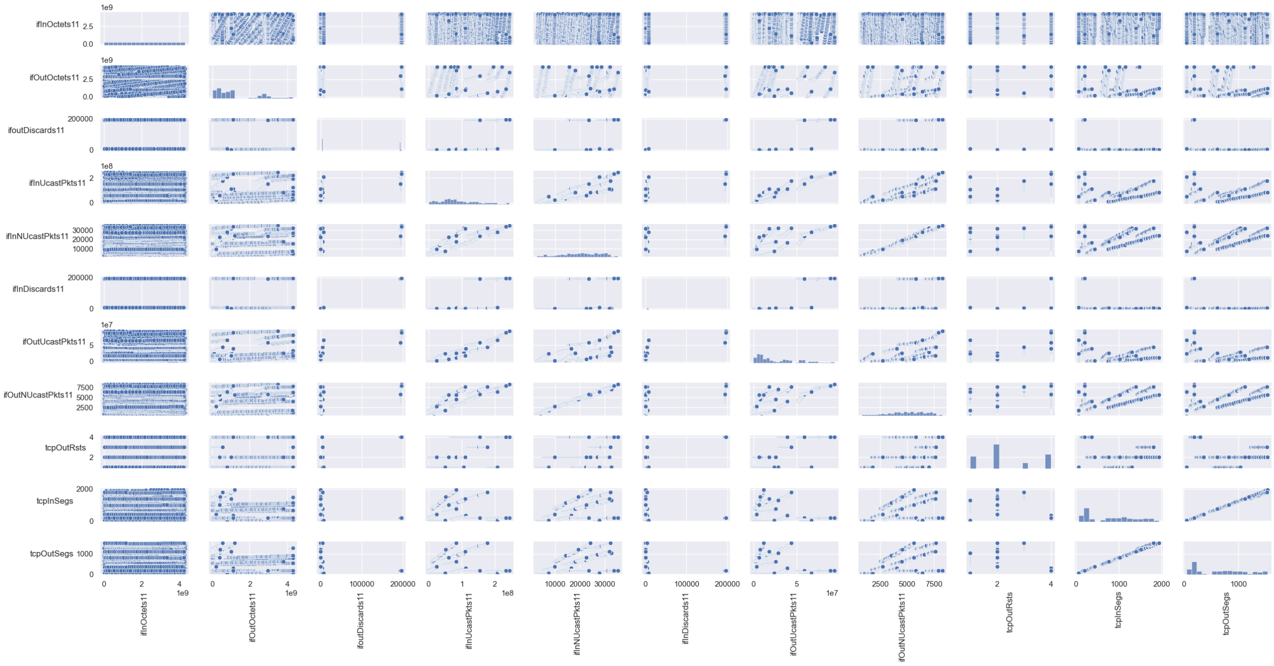


Figure 3.9: Demonstration of the variables relationships of the computer network dataset.

From this point, the author has demonstrated that each dataset is using the same data attributes and format for detecting anomalies. Therefore, the anomaly detection models are compared based on their efficiency by the area under the RoC curve metric. Thus, the practical significance of the proposed model is demonstrated. In section 3.3, the results and discussion of applying the proposed methodology are presented.

3.3 Results and Discussion

According to the data labeling limitation in the dataset, it is crucial to determine contexts and behavior attributes for detecting the anomaly in the urban logistics operation. In section 3.2.2, the result using the Bayesian Network is derived, as shown in Table 3.3.

From the experiment result, the author discovered that anomalies in the fleet management system mainly occur from vehicle issues. Table 3.2 demonstrated that the connection between the vehicle's node to the received request(Actual delivery)'s node and point toward the incident's node are most likely cause anomalies in the urban logistics operation. The results showed that a low possibility to have a regular link between these nodes.

Furthermore, when Equations (3.3) and (3.4) are calculated, it cause a high anomaly score. This phenomenon was showing that the anomaly scores were higher than those of other connections. The criteria to determine anomalies are as shown in Table 3.3. Thus, the cause of the anomalies in urban transportation logistics mainly occurs from vehicle issues. The event led to abnormal business behavior. This phenomenon occurred when the logistics agency response to potential requests from customers. After that, incidents occurred as a consequence. From this outcome, the vehicle usage is set as a behavior attribute in each time step (time context). Then, the author combined it with behavioral decision in Equations (3.5), (3.6) and (3.7), respectively. Thus, the summaries of criteria for defined anomalies are shown in Table 3.3.

After the contextualization process was done, the next step is the LSTM model execution. The model parameters are defined in the next section.

3.3.1 Model Construction and Parameter Evaluation

After the contexts and behavior attributes for anomalies are defined, the LSTM model is then constructed and set up its parameters from hyper-parameter tuning. These parameters are shown in Table 3.4.

To train and test the model, the author used real operational data between January 2017 to April 2019 for training and May 2019 to July 2019 for testing. Also, each experiment is given 10 runs to make sure the sustainability of the results. Therefore, the mean values of the evaluation metrics are presented. Note that all experiments of this chapter were performed using Python 3.6 with the Keras library. The TensorFlow backend was used and run on a google colab GPU.

The author input processed data into the model for training and testing. The model's training and testing stage is illustrated as Figure 3.10. In the next section, the results of running the model are presented.

3.3.2 Experimental Results

After the LSTM model was already constructed and executed, the procedures described in section 3.2.5 are performed. They are used to calculate prediction error and defined a set of thresholds. In the experiment, the experiment is divided into sub-experiments. They consist of the experiment of the model that detected the only point, contextual, and collective anomalies. The last experiment was combined with both types of anomalies.

Table 3.5 shows the experiment result of detecting point and collective anomaly. The result is obtained when applying only a prediction error threshold and sliding windows approach.

The experiment result shows that when the window size increased, the AUC also increased. However, the values of the AUC were not significantly high. It is a consequence of the false positive and true positive rates. This phenomenon shows that the data points contained dependencies between time-steps. The change of the previous time-step, as computed as Equation

Table 3.2: Summary of anomaly scores using the Bayesian Network.

Attribute(s)	Received req.	Incident	Request	Vehicle	Swap fleet	No driver	Maintenance
Received req.	0	1	1	**9**	2	3	1
Incident	2	2	2	**12**	2	1	3
Request	1	0	0	0	2	0	1
Vehicle	**11**	**12**	3	0	3	4	4
Swap fleet	2	2	3	1	0	2	2
No driver	2	2	2	0	2	0	2
Maintenance	1	2	2	0	2	1	0

Table 3.3: Criteria and condition for determining anomalies.

Description	Status
Received request(R) and Vehicle (V) not exceed $threshold_v$ and incident not exceed $threshold_I$	normal
Received request(R) and Vehicle (V) exceed $threshold_v$ and incident exceed $threshold_I$	anomaly
Received request(R) exceed $threshold_v$ and Vehicle (V) not exceed $threshold_v$ and incident not exceed $threshold_I$	anomaly
Received request(R) not exceed $threshold_v$ and Vehicle (V) exceed $threshold_v$ and incident exceed $threshold_I$	anomaly

Table 3.4: Parameter setting for the LSTM model.

Model parameter(s)	Values
Hidden layer	2
Unit in hidden layer	80
Sequence length	730
Training Iteration (epoch)	1,000
Dropout	0.3
Batch size	72
Learning rate	10^-3
Optimizer	adam
Input dimension	12

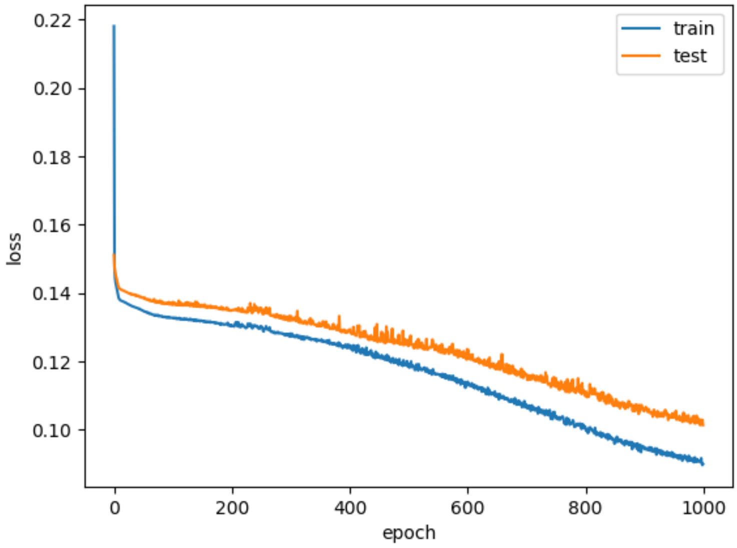


Figure 3.10: Training the model with time-series data.

Table 3.5: Experiment results when the changes between each time window and the dynamic threshold are applied.

Window size	AUC	Precision	Recall	FPR	F_1-score
1	**0.572**	**0.714**	**0.189**	**0.047**	**0.299**
2	0.573	0.673	0.209	0.063	0.319
3	0.582	0.673	0.234	0.070	0.347
4	0.588	0.667	0.253	0.078	0.367
5	0.588	0.646	0.266	0.090	0.377
6	0.590	0.638	0.278	0.098	0.387
7	0.588	0.646	0.266	0.090	0.377
8	0.597	0.632	0.304	0.109	0.411
9	0.593	0.615	0.304	0.117	0.407
10	0.593	0.615	0.304	0.117	0.407

(3.13), influences the status of the current time-step data point in the window for defining anomalies. The change is represented by the anomaly score in Equation (3.12). The anomaly score was based on a threshold defined earlier in section 3.2.3. Thus, it was combined as anomaly scenarios as in Table 3.3. Unfortunately, when the author increased the window size, it also causes a high false-positive when anomaly detection is performed. The window is set with a size equal to 1. This window size has a high chance to correctly detect anomalies compared to other window sizes.

From this experiment, the context and estimated the different changes are required. It is because they can effectively reveal whether or not the next time step is anomalous. They also enhance capability in the detection. In the next experiment, only the context is specified. The author did not consider changes of the previous time-step values, as shown in Table 3.6.

Table 3.6: Experiment results when the contextualized threshold of each time window is applied.

Window size	AUC	Precision	Recall	FPR	F_1-score
1	**0.504**	**0.88**	**0.696**	**0.059**	**0.777**
2	0.689	0.619	0.608	0.230	0.613
3	0.658	0.549	0.639	0.324	0.591
4	0.597	0.471	0.627	0.434	0.538
5	0.563	0.437	0.614	0.488	0.511
6	0.534	0.409	0.614	0.547	0.491
7	0.525	0.402	0.633	0.582	0.492
8	0.526	0.403	0.595	0.543	0.481
9	0.522	0.399	0.614	0.570	0.484
10	0.504	0.385	0.614	0.605	0.473

Table 3.5 demonstrated that each time-step data point does not contain any change dependencies. Therefore, in this experiment, the model's setting is changed to the context of behaviors. These contexts and the decision condition are described and shown in Table 3.3. They were supported by the threshold defined in section 3.2.5. It also included the anomaly score in Equation (3.12).

Table 3.6 shows that after changing the model setting. It did not return a significant change except when the window size is equal to 1. When the window size is set to 1, the precision increased to 0.88 and the recall to 0.696. Thus, the author concluded that the optimal window size for this experiment was equal to 1. It is because it had a high rate of precision and recall. The FPR was also the lowest among others.

Unfortunately, the AUC result decreased by 4% on average from the previous experiment. This phenomenon supported this study's assumption that transportation logistics contained conditional anomalies. Moreover, They required a specific context and behavior attributes to detect it. This assumption led to the subsequent experiments. This time, the dependencies between the time-step and the context for detecting data point behavior are combined. The experiment result is shown in Table 3.7.

From the experiment result, the author discovered that combine these approaches improved the AUC by 17% compared to the first experiment and 20% for the second experiment. Similar to the previous experiment, the window size of 1 was suitable for detecting anomalies. It provided the highest precision and AUC compared to other window sizes. The result also shows that suppose the window size increases, then it reduces the detection performance. It is because each data point does not depend on the other. The rationale behind this model is that the mean anomaly score within the window is taken to define the threshold. Suppose that the data points value in the window exceeds the threshold, then those data points are classified as collective anomalies if the window size more than 1. Otherwise, it is classified as a

Table 3.7: Experiment results when the contextualized and the dynamic threshold are applied combine with each timestep's changes.

Window size	AUC	Precision	Recall	FPR	F_1-score
1	**0.799**	**0.831**	**0.684**	**0.086**	**0.750**
2	0.728	0.633	0.709	0.254	0.669
3	0.730	0.599	0.785	0.324	0.680
4	0.691	0.556	0.753	0.371	0.640
5	0.690	0.553	0.759	0.379	0.640
6	0.697	0.554	0.785	0.391	0.650
7	0.677	0.567	0.671	0.316	0.615
8	0.684	0.571	0.684	0.316	0.622
9	0.680	0.561	0.696	0.336	0.621
10	0.670	0.551	0.684	0.344	0.610

point or contextual anomalies. However, it still has room to be improved the recall metrics. It also a piece of evidence that the prior time-series did not influence the data point. Therefore, the experiment is performed when the dependencies of time-step (t) and time-step ($t - 1$) are removed, using Equation (3.13).

Table 3.8: Experiment results when the contextualized threshold and the dynamic threshold are applied together.

Window size	AUC	Precision	Recall	FPR	F_1-score
1	**0.870**	**0.836**	**0.842**	**0.102**	**0.839**
2	0.754	0.647	0.766	0.258	0.701
3	0.713	0.646	0.646	0.219	0.646
4	0.692	0.574	0.709	0.324	0.634
5	0.693	0.583	0.690	0.305	0.632
6	0.698	0.614	0.646	0.25	0.630
7	0.699	0.587	0.703	0.305	0.640
8	0.693	0.592	0.671	0.285	0.629
9	0.688	0.596	0.646	0.270	0.620
10	0.672	0.561	0.665	0.320	0.609

Table 3.8 shows that the prior time-step data point did not influence the current prediction time-step. Also, the precision and recall metrics were significantly improved over the prior experiment. Similar to the previous experiments, the optimal window's size was equal to 1. It provided the most efficient detection when compared to other window sizes. This phenomenon proved that each data point independently occurs. Therefore, anomalies in transportation logistics are considered as a point or contextual, not collective anomalies in this case.

The recommendation to improve this chapter's model would be the consideration of factors. The author discovered that anomalies could originate from various perspectives. However, this study focused on operational anomalies. Thus, in the future work, the capability to detect anomalies in more aspects is expanded. This expansion also includes discovering the causes that influence anomaly occurred during the vehicle route optimization process. In addition, another essential part is the data points that appear with regular data points. They can cause a low-recalled rate. This issue is one of the limitations of this chapter's detection model.

Moreover, the second limitation is the reliability issue. From Chapter 2, it was showing that the statistic thresholds are not always appropriate to detecting abnormal patterns. Suppose that the value does not increase over the threshold, then the anomalies were not detected. Sometimes, a false positive detection happened when the data value is over the threshold.

To prevent these issue to happened, the contextualization of the data is proposed in this chapter. However, to improve the result further, some specific contexts other than transportation planning—for instance, manufacturing and production timelines should be taken into consideration also. Therefore, a connection between multi-factors was required to evaluate the urban transportation logistics operational behavior.

In addition to the data contextualization and thresholds experiment, the proposed model was also compared with the state-of-the-art models. It also includes the model that is optimized the detection capability with the genetic algorithm (GA), up to 50 iterations. The results are shown in Table 3.9. Note that the bold text model in Table 3.9 is the model that applied the methodology of this study.

Table 3.9: Experiment results for temporal behavior analysis model.

Model	AUC	Precision	Recall	FPR	F_1-score
Hundman et al. [18]	0.572	0.714	0.189	0.047	0.3
Xu et al. (Donut) [21]	0.504	0.385	0.614	0.605	0.473
Average of Baselines	0.538	0.549	0.401	0.326	0.367
LSTM+GA with context.	0.845	0.881	0.753	0.063	0.812
LSTM with context.	**0.870**	**0.836**	**0.842**	**0.102**	**0.839**

In Table 3.9, after applying this study methodology, the precision and recall significantly improved over the previous study model.

Furthermore, the context of the data point also crucial. The contexts and behavior attributes must appropriately define. The rationale behind this is that in the traditional LSTM with a dynamic threshold model from [18]'s study, it only takes prior input to predict the next time-step. However, if the sequences always have a similar pattern, then the LSTM with a dynamic threshold model is accurately predicted. Nevertheless, suppose the sequence's pattern has a variant and does not conform to any pattern. In that case, the LSTM might return the wrong prediction and concluded that this data point is an anomaly. This problem also happened to [21]. Note that to increasing capability of detection in [21] model. It is required to have additional data labels. However, the author compares the models in the unsupervised way. Therefore, the data labeled is discarded.

Unfortunately, this conclusion not always be valid. The reason to support this statement is presented in terms of an example. For instance, at time-step (t), there is a high vehicle usage in the fleet. This never happens before. If we used only the LSTM model to make predictions, it would detect that this event is an anomaly.

However, suppose we add some contexts to the model, such as high demand from the customers. In that case, this event is not the anomaly anymore because of the context high customer demand is correlated with high usage of the vehicle. Therefore, it enhances the prediction accuracy and lowers the false positive rate.

Moreover, the author also compared the proposed model with the model titled "LSTM+GA with context.", which was optimized the detection capability by GA. The experiment results showed that our proposed model is more effective in detection because the GA approach performs feature selection tasks. Therefore, some features that are important for detecting anomalies are removed.

Besides, the author also performed an additional experiment by testing the proposed model with other data. The data consist of credit cards transactions [63], networking [64], water manufacturing [65], and transportation logistics system. This experiment aims to reveal the practical implication of the model on detecting anomalies. The experiment is presented as Figure 3.11.

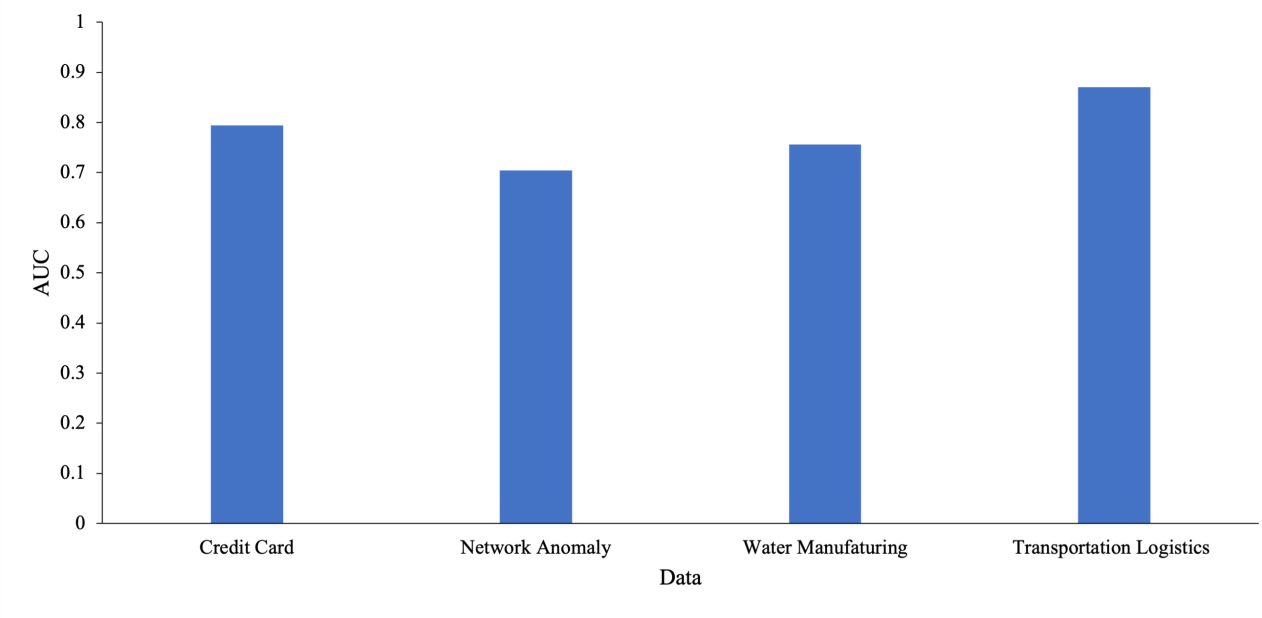


Figure 3.11: Experiment results of anomalies detection using the proposed model on other applications data.

Figure 3.11 illustrated that the proposed model is practical for performing anomaly detection in various applications with some modifications, as described in section 3.2.7. The model has an AUC between 0.7 – 0.9, which is an acceptable result. The rationale behind this impressive result is that most application data are dependent on a location-time context and behavior attributes such as resource usages and transactions throughout the day. Therefore, the LSTMs with contextualized thresholds are able to adapt to changes in the data usage pattern at a specific time.

These procedures are done by computing prediction errors and anomaly thresholds/degrees in each sliding window. The current window is compared with the next time windows. Suppose there is a significant difference within each window. It is then denoted that there is an anomaly detected. The computation started with the first data series. It continues to update the model until the end, and the process is terminated. Lastly, the abnormality and normality are then revealed.

In the future work, the author suggests combining the LSTM-contextualized dynamic threshold approach with the multi-level-density-based approach. This suggestion is to increase the model reliability. This improvement can also leverage the benefit of distance and statistical approaches together to improving the model's detection performance. The author discovered from the literature review and the experiments that anomalies did not always result in a high prediction error. It is also hidden and mix with regular data points. Thus, this is why the anomaly study was vital to this research domain and comprehensive to find a feasible solution.

We reach the last part of this chapter, where the author concludes the study findings and limitations. The conclusion is presented in the next section.

3.4 Conclusion

This chapter presents an important challenge in detecting abnormal behavior in a vehicle route optimization process. It was obtained the benefits from novel anomaly detection approaches. This chapter demonstrated that the LSTM-based approach was suitable for detecting abnormal operational behaviors. At the same time, the author addressing the challenge involved. It also included the remaining research question associated with unlabeled multi-dimensional datasets.

As a result, the author proposed a novel contextualized with a dynamic threshold approach. This approach does not depend on any labels and it is entirely unsupervised. This approach's capability was expanded to detecting complex anomalies. They are anomalies that have dependencies between multi-dimensional factors. Consequently, the root causes are also identified.

The experimental result shows that the data points in the time-series do not depend on each other. However, the anomalies are depending on environmental conditions. Therefore, specific contexts and behavior attributes were required to detect these complex anomalies. The experiment showed that the AUC, precision, and recall metrics increased significantly to 0.870, 0.836, and 0.842. This phenomenon occurred after the model included the spatial-temporal contexts and behavior attributes. The author also identified the necessary factors behind the improvement and furthered the model's evaluation.

Lastly, the proposed approach's applications were developed not limited to transportation logistics areas but also include all areas that want to eliminate abnormal events and developing optimal policies. This improved methodology will enable more reliable and efficient decision-making.

In the future work, the author wishes to expand the proposed approach's capabilities. This expansion also includes implementing the framework, which takes other operational areas into account. This area also involves the streaming of data.

Chapter 4

A Novel Unsupervised Behavior and Root-Cause Analysis Framework for Transportation Logistics

4.1 Introduction

This chapter presents the procedures for performing the behavior and root-cause analysis. As mentioned in the literature review chapter, it is essential to learn about a company operation's behavior, especially in the transportation planning process using data from multiple sources. Additionally, it is essential to comprehend how the environment impacts the transportation planning results from this obtained information.

The behavior and root-cause analysis results are used as information for staffers in the company to improve their assigned tasks. Moreover, the information is also used to train autonomous transportation planning agents in performing vehicle route optimization tasks. This task has become a crucial task influenced by the growth of the economy.

The autonomous transportation planning agent is similar to a human agent. It can perform actions within a given environment. Generally, it is not feasible to recognize the actions of the appropriate agents and those that should be avoided to achieve the optimal solution. This chapter also investigates how an agent's behavior can be effectively detected and its impact on the environment can be judged. For example, the causes of the action and event include their normality and abnormality. The outcome is then updated for the agent to preserve its cumulative reward when performing the actions. Therefore, the appropriate action was chosen.

The behavior analysis in this chapter comprises 3 parts, i.e., temporal, static disturbance detection, and anomaly explanation. After performing all detections, the analysis of the root cause is performed. Finally, these procedures are combined into a hybrid behavior and root-cause analysis model.

The purposes of developing this hybrid model are as follows. The first purpose is to reduce the detector selection bias, as discussed in Chapter 2. [44] indicated that the detection methods have different capabilities for detecting abnormalities from data points. Therefore, combining the results can reduce the model selection bias and weak classifiers.

Moreover, the second purpose is to improve the detection results. It was mentioned in the anomaly detection challenge in Chapter 2 that the noise-silence anomaly and anomaly explanation are crucial tasks. This is because the data come from multiple sources and have multiple dimensions. Therefore, the anomalies might look similar to a normal event, and they do not reveal their identity. This phenomenon is called nonlinearly separable between data points. As a result, the detection model returns a low detection rate. For instance, anomalies are detected as normal events, and normal events are detected as anomalies. This issue is a limitation of conventional ML and threshold-based methods. They cannot effectively perform the detection tasks, as reconstruction error does not easily calculate the separation boundary. This statement is also supported by the finding of [19]'s study that simple thresholds are not sufficient for detecting anomalies in nonlinear data.

In Chapter 3, the author also mentioned that thresholds are necessary to adapt to data changes. This is the reason why the sliding window approach is proposed. Furthermore, the reason and hint of the abnormality are given and therefore useful for further decision making. The anomaly explanation (root-cause analysis) reduced the bias on labeling the datapoint [4] and provided reasons for deciding whether this datapoint is normal or abnormal. Therefore, the detection results are enhanced and accurately detected.

In the last part of this chapter, the practical applications and limitations of the proposed model are presented. The author aims to demonstrate how general the proposed model was for expanding to other application domains. For further details, they are presented in the "Methodology" section.

4.2 Methodology

The overall framework of this study is shown in Figure 4.1. Therefore, in the upcoming sections throughout this chapter, the processes in Figure 4.1 are explained in detail.

4.2.1 Data Preprocessing and Feature Engineering

As mentioned previously in Chapters 2 – 3 that logistics agencies' operations consist of impact features. They are required to be taken into consideration when performing behavior analysis to detect disturbances and their root cause. Therefore, the pre-processed features (e.g., standardize, reshape, and transform) and extracted from spatial-temporal data consist of route distance, traveling time, speed, location (latitude, longitude) based on time are essential for this analysis. Table 4.1 shown the example from spatial–temporal data. They were used to detect the driver's behavior and determine the efficiency of the daily vehicle's route assignment process. Note that the dataset used in this chapter is the same as Chapter 3. In this section, the data have been organized into groups. These groups of data are used to train and test the models presented in section 4.2.2.

Table 4.1: Demonstration of the spatial-temporal data.

Name	Value	Unit
DeviceID	"213L2017000968"	-
Latitude	"13.68691"	-
Longitude	"100.46526"	-
Speed	45.4	km./hr.
Distance	69.2	km.
Time-spent	79	minutes
Time-stamp	2018-01-01 18:50:23	Hr:MM:SS

In addition to the spatial-temporal features, the raw operation dataset is shown in Table 4.2. When these data attributes are visualized. It is shown as Figure 4.2. It is possible to

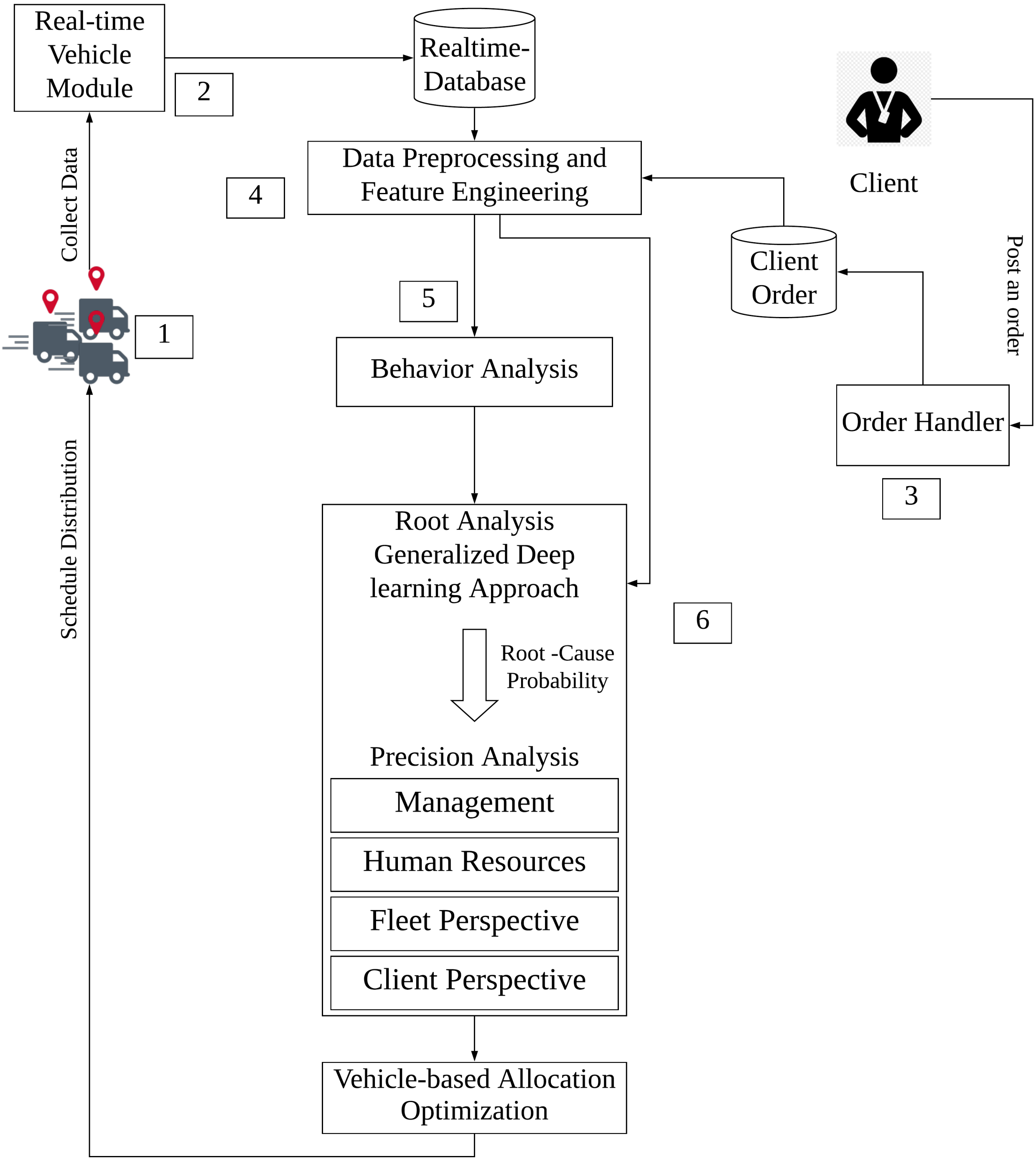


Figure 4.1: The proposed framework for behavior and root-cause analysis with applications in urban route logistics optimization.

derive additional features in terms of operation perspective. Referring to supply chain management strategy [66], where the authors suggest those factors that are important to evaluate the performance of the operation, the following new features were calculated as follows:

Human Resources Perspective

- Driver workload: This ratio aims to determine the workload assigned to each driver. A ratio greater than 1 shows that the capacity of a particular driver is exceeded, which is

Table 4.2: Demonstration of the raw operation report.

Name	Value	Unit
Date	2/10/2018	-
Number of available vehicles	45	Vehicles
Number of occupied vehicles	7	Vehicles
Number of vehicles with no assigned driver	8	Vehicles
Number of vehicles with back order work	0	Vehicles
Number of vehicles in maintenance	0	Vehicles
Number of driver has taken leave	0	Vehicles
Number of total requested from client	45	Orders
Number of requests received	45	Orders
Number of orders canceled (import)	0	Orders
Number of orders canceled (export)	0	Orders
Quarter of the year	Q4	-

shown in Equation (4.1):

$$\frac{\sum_{j=1}^{n}(client_{requested_j})}{\sum_{i=1}^{n}(driver_{assigned_i})} \tag{4.1}$$

where $client_{requested_j}$ denotes the request of service from each client, $driver_{assigned_i}$ denotes the individual driver assigned to handle the request, and n denotes the total population. This ratio is represented for each day of the time series.

- Driver availability: This ratio shows the available drivers compared to the overall demand from the client assigned to drivers. A ratio greater than 1 shows that some drivers have no assigned work compared to the demand flow into the company. However, if the ratio is less than 1, then the company has a shortage of drivers to deliver goods to the existing and potential customers, as shown in Equation (4.2):

$$\frac{\sum_{i=1}^{n}(driver_{available_i})}{\sum_{j=1}^{n}(driver_j)} \tag{4.2}$$

where $driver_{available_j}$ denotes the driver available to handle the work assignment, $driver_j$ denotes an individual driver in the fleet and already has a task assigned, and n denotes the total population. This ratio is represented for each day of the time series.

Resources Utilization Perspective

- Fleet utilization: This ratio aims to determine the usage of vehicles compared to the overall work, as shown in Equation (4.3):

$$\frac{\sum_{i=1}^{n}(complete_{vehicle_i})}{\sum_{j=1}^{n}(vehicle_j)} \tag{4.3}$$

where $complete_{vehicle_i}$ denotes the individual vehicle that completed the assigned task, $vehicle_j$ denotes the individual vehicle available in the fleet, and n denotes the total population.

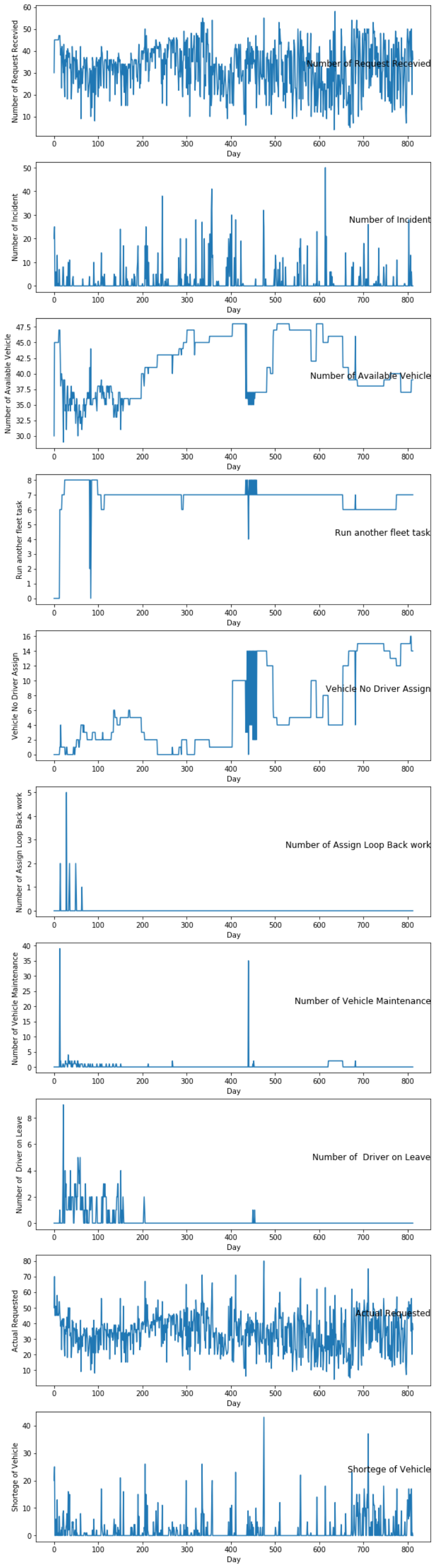


Figure 4.2: Data visualization

- Productivity ratio: This ratio reveals how well the fleet is working by comparing the success rate with the overall work.

$$\frac{\sum_{i=1}^{n}(complete_{vehicle_i})}{\sum_{i=j}^{n}(client_{requested_j})} \tag{4.4}$$

where $complete_{vehicle_i}$ denotes the individual vehicle that completed the assigned task from the planning division, $client_{requested_j}$ denotes the individual vehicle requested by the client, and n denotes the total population.

Management Perspective

- Vehicle shortage ratio: This ratio shows how well the companies have planned to handle the demand from the client. If the ratio is more than 1, it means there is a vehicle shortage, and the work could not be handled as per the customer's request.

$$\frac{\sum_{i=1}^{n}(incomplete_{task_i})}{\sum_{j=1}^{n}(client_{requested_j})} \tag{4.5}$$

where $incomplete_{task_i}$ denotes the required individual vehicle that overflows the capacity of the fleet, $client_{requested_j}$ denotes the individual vehicle requested by the client, and n denotes the total population.

- Incident ratio: This ratio shows the difficulty in transportation planning that may arise from a potential client, such as postponement or cancellation request. Therefore, the ratio is more than or equal to 1.

$$\frac{\sum_{i=1}^{n}(incident_{order_i})}{\sum_{j=1}^{n}(client_{requested_j})} \tag{4.6}$$

where $incident_{order_i}$ denotes the individual order that is postponed or canceled by the client, $client_{requested_j}$ denotes the individual vehicle requested by the client, and n denotes the total population.

4.2.2 Anomaly Detection

Unsupervised Hybrid Anomaly Detection Model

The unsupervised hybrid anomaly detection model is motivated by the previous research challenge to deal with high dimensional data and to reduce false-positive rates that cause from not linearly separable, as mentioned in the literature survey. The conventional ML and thresholds-based are not efficient in detecting this kind of anomalies. Therefore, the author experimented on well-known methods used for data dimension reduction, such as Principle Component Analysis (PCA) and Autoencoder (AE). For PCA, the output is taken from the model and applied density- and distanced-based methods. These procedures are used to detect disturbances from the data. The AE is used as a Long Short-Term Memory Autoencoder (LSTM-AE) because the data is in the form of time-series.

Furthermore, AE is capable of learning and preserving features representation of the dataset. This feature representation is vital for adequate classification. Thus, the author changed the conventional AE architecture from the feed-forward neural network into the LSTM recurrent network. The model was trained with the collected data from section 4.2.1 using the procedure as follows:

The LSTM-AE was implemented, which used the LSTM layer run as an encoder. The encoder encode the input (x), as computed in Equation (4.7). Also, another LSTM layer runs as a decoder. The decoder reconstructed x' from input (x) in the decoding step, as computed in Equation (4.8). The squared error was also used to determine the difference between the model's input and output values. In this context, the square error is denoted as reconstruction error, as computed in Equation (4.9). It should be noted that W_x and W_y are the weight matrix of the encoder and decoder, while b_1 and b_2 are the bias vectors in each phase.

The LSTM-AE has equal input and output in a hidden layer. This layer is used to handles the input units. The input is set equal to 12 for this study. In addition, the LSTM-AE model's architecture consists of one encoded layer and one decoded layer. According to the LSTM-AE is based on a neural network, the reconstruction error was optimized. A back-propagation algorithm is applied to optimize the reconstruction error. An error signal is computed by the back-propagation algorithm. It then transfers the value back through a network. The model was tuned at 5,500 epochs, 64 of batch size, adam optimizer, MSE loss function, and learning rate of 10^{-4}. The proposed LSTM-AE is shown in Figure 4.3.

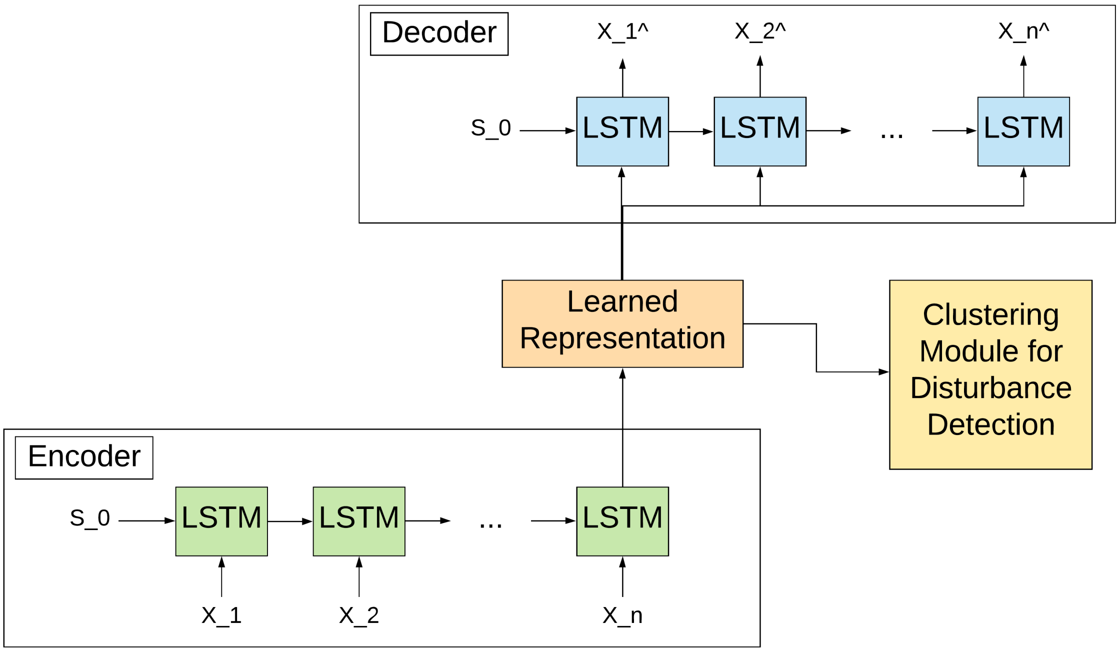


Figure 4.3: Demonstration of LSTM-AE architecture.

$$h(x) = f(W_x + b_1) \tag{4.7}$$

$$\hat{x} = g((W_y \times h(x)) + b_2) \tag{4.8}$$

$$L(x, \hat{x}) = ||x - \hat{x}||^2 \tag{4.9}$$

where $h(x)$ denotes the hidden encoder vector calculated from input vector x and $\hat{x}$ denotes the reconstruction vector of the output layer. In addition, f and g denote the encoding and decoding function.

While the model was running, the output was taken from the encoding layer. This encoded layer's output is used as the input of density- and distance-based models. For instance: Kmean, One-Class Support Vector Machine, Isolate Forest, and Gaussian Mixture Model (GMM). In this chapter, the GMM was proposed to perform the clustering task on the encoded value. This encoded value is returned from the encoded layer. This procedure was shown by [5, 34] that the joint learning approach was much more effective for encoding features. It also capable of projecting highly dimensional data into low dimensional space. Furthermore, it can adequately define the different clusters' regions when performing dimension reduction with non-linear data. This model's outcome aims to solve the detecting static disturbances problem from high dimensional data. Thus, the data points that do not belong to any cluster are labeled as a disturbance.

Dynamic Ensemble Weight Average Method

As seen so far, models have been developed to deal with temporal and context disturbances. This model was presented in Chapter 3 and the detection model for point disturbances presented in section 4.2.2. Note that in this chapter, the author replaces Chapter 3 LSTM-based model with LSTM-AE. The reason is that the LSTM-AE is more capable of learning the features representation of the dataset.

Each model has different advantages and disadvantages when used for behavior analysis to detect disturbances. Suppose that we want to combine these models together, an ensemble weighted average method is suggested. It shown in previous study [45, 43, 50] that it provided efficient results. Therefore, the proposed weighted average method was computed the model's $ensemble_{weight}$, as defined by Equation (4.10). The equation was computed based on the average model's prediction performance.

$$ensemble_{weight} = 1 - \frac{1}{m}\sum_{i=1}^{m}\left(\frac{false^{i}_{positive} + false^{i}_{negative}}{n}\right) \times w_i \tag{4.10}$$

where n represents the total amount of data points, m represents the number of models in the ensemble, $false^{i}_{positive}$ represents an actual value that is not a disturbance. However, the model i-th falsely detected as disturbance, $false^{i}_{negative}$ represents the actual value which is a disturbance, but the model i-th detected as not a disturbance, and w_i represents the weight or how important of the model i-th. For this study, w_i is set to 1 as every model is equally important.

In contrast, the model which has a high $ensemble_{weight}$ is the model that has a low prediction error. Therefore, these model outputs are chosen to merge with others. However, suppose that the model obtained a low $ensemble_{weight}$; this phenomenon denotes that the model has a high prediction error. Therefore, the outputs of weak detectors are discarded for the final prediction.

When the new data object is input into the system, the dynamic model selection process is performed. This process of dynamic model selection was adopted from [67, 49]'s study. The procedures are described as follow:

- **Step one**: The models' inputs in section 4.2.2 are compared between the prior information and new inputted information with sliding windows. Suppose that these two pieces of information are different and cause a high prediction error. The models are then updated and executed with respect to the new information. After that, the ensemble weight average methods are computed using Equation (4.10).

- **Step two**: The method chooses the model's output with the topmost $ensemble_{weight}$ (e.g., Top 2 to 6) to be stored in the ensemble list.
- **Step three**: Check whether all model's weight is updated. Suppose not, go back to step two; otherwise, go to step four.
- **Step four**: The process averages the output from the selected models as a final detection result. If the average output of the ensemble model is more than 0.5, this value was obtained by computing the mean in the ensemble list. In this event, the data point is classified as an anomaly (1); otherwise, it is classified as normal (0).

In summary, the process which performed in Sections 4.2.1 – 4.2.2 are summarized as Figure 4.4 and are transformed to the computation algorithm, as Algorithm 4.1 to Algorithm 4.3.

Algorithm 4.1: Temporal behavior analysis

```
Input  : Training set $T_r = \{F_1, F_2, \ldots F_n\} \in \Re$; test point $F_i \in \Re$
Output: The prediction result of anomaly detection ($Pred^n$)
$\hat{y}_t$ = LSTM($T_r$) → Train LSTM model on $T_r$ to derive the prediction error $e^t$ %Chapter 3.2.4
$e^t = |y_t - \hat{y}_t|$
$e_s = EWMA(e^t)$
for i = 1:N do
    $Threshold_p = \mu(e_s) + z\sigma(es)$
    $Threshold_v = \mu(v) \pm z\sigma(v)$
    $Threshold_i = \mu(i) \pm z\sigma(i)$
    $Score_i = \frac{1}{n}\sum_{i=1}^{n}(y_i w_i)$
    if $Score_i > \mu(\sum_{i=0}^{n} Score_i)$ then
        $[LSTM_{anomaly}, index_i] \leftarrow 1$
    else
        $[LSTM_{anomaly}, index_i] \leftarrow 0$
    end if
end for
$Pred^n = \{LSTM^1_{anomaly}, \ldots, LSTM^n_{anomaly}\}$
return $Pred^n$
```

4.2.3 Root-Cause Analysis

In the anomaly detection section, the hybrid model of anomaly detection was proposed. The model aims to detect a temporal and static anomaly that is raised as challenged in Chapter

Algorithm 4.2: Joint learning model of LSTM-AE and clustering method.

```
Input  : Training set $T_r = \{F_1, F_2, \ldots F_n\} \in \Re$; test point $F_i \in \Re$; Set of Prediction
         error $e_s = \{e_1, e_2, \ldots e_n\}$; Set of Clustering $C = \{C_1, C_2, \ldots C_n\}$
Output: The prediction result of anomaly detection $(LSTMAE_{anomaly})$
$Model_{LSTMAE}$ = LSTM-AE$(T_r, e_s) \rightarrow$ Train LSTM-AE model on $T_r$ to derive the
encoded value %section 4.2.2
$[encoded, index] = Model_{LSTMAE}(F_i)$ %use trained model to predict the test set.
for i = 1:N do
    for j = 1:N do
        $[Cluster_i, index] = C_i(encoded_j)$ %used clustering technique to cluster the data into
        groups
        if $Cluster_i$ is $false$ then
            $[LSTMAE_{anomaly}, i, j] \leftarrow 1$
        else
            $[LSTMAE_{anomaly}, i, j] \leftarrow 0$
        end if
    end for
end for
$LSTMAE_{anomaly} = \{LSTMAE^1_{anomaly}, \ldots, LSTMAE^n_{anomaly}\}$
return $LSTMAE_{anomaly}$
```

Algorithm 4.3: Ensemble model for anomaly detection.

```
Input  : Set of detection result $D_t = D_1, D_2, \ldots, D_n$; Set of evaluation metric $E_m = E_1, E_2, \ldots, E_n$
Output: The prediction result of anomaly detection ($Pred^n$)
for i=1:N do
    Compute $[ensemble_i, index] = 1 - \frac{1}{m}\sum_{i=1}^{m \in E_m}\left(\frac{false^i_{positive}+false^i_{negative}}{n}\right) \times w_i$ %section 4.2.2
    tmp ← 1 - $argmin(ensemble_i)$ Find the model which has ensemble weight close to 1 %top 2 to 6
    $[AVG_{result}, index] \leftarrow average(tmp)$
    if $AVG^i_{result} > 0.5$ then
        $[ensemble_{anomaly}, index_i] \leftarrow 1$
    else
        $[ensemble_{anomaly}, index_i] \leftarrow 0$
    end if
end for
$Pred^n = \{ensemble^1_{anomaly}, \ldots, ensemble^n_{anomaly}\}$
return $Pred^n$
```

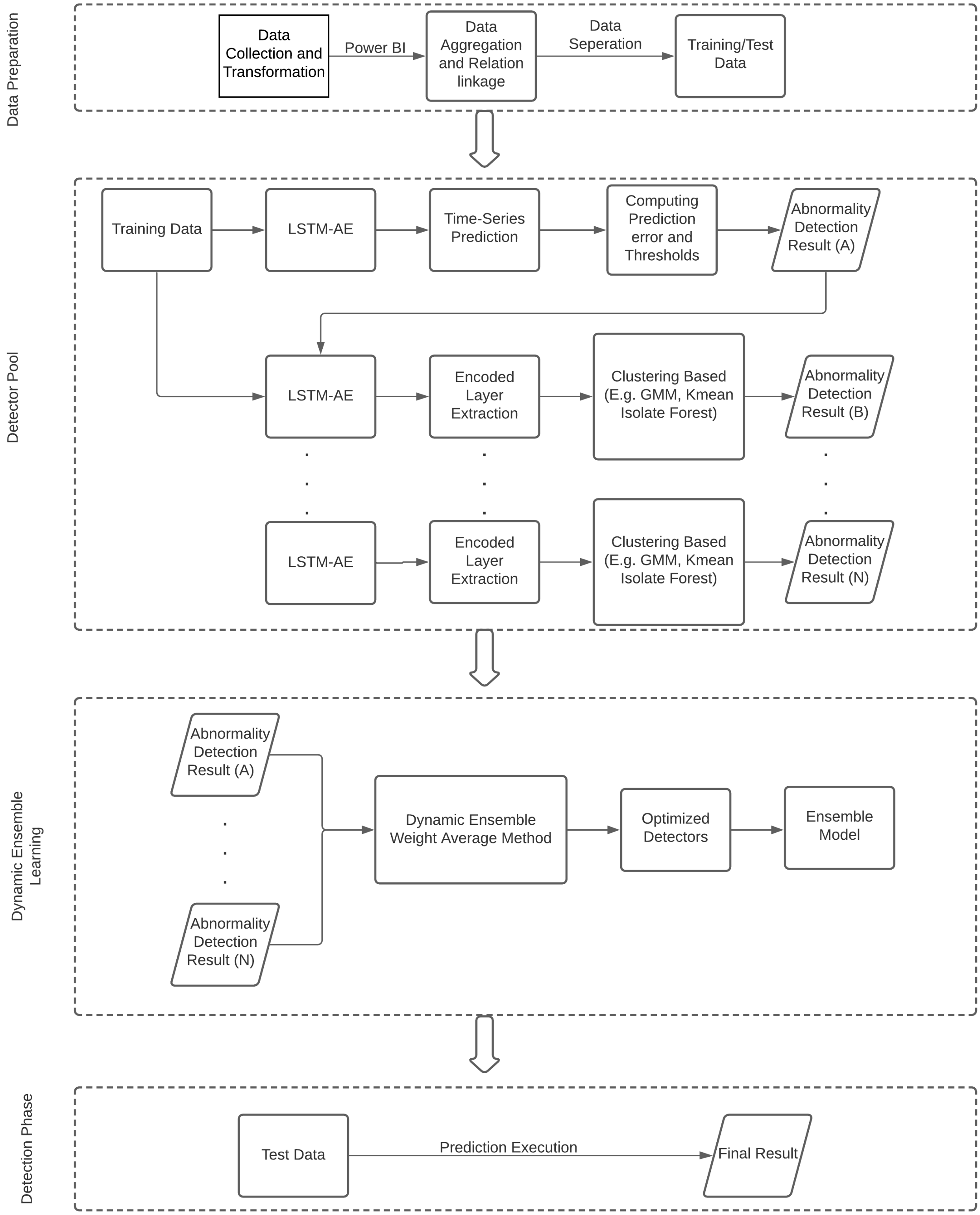

Data Preparation
Data Collection and Transformation
Power BI
Data Aggregation and Relation linkage
Data Seperation
Training/Test Data
Detector Pool
Training Data
LSTM-AE
Time-Series Prediction
Computing Prediction error and Thresholds
Abnormality Detection Result (A)
LSTM-AE
Encoded Layer Extraction
Clustering Based (E.g. GMM, Kmean Isolate Forest)
Abnormality Detection Result (B)
LSTM-AE
Encoded Layer Extraction
Clustering Based (E.g. GMM, Kmean Isolate Forest)
Abnormality Detection Result (N)
Dynamic Ensemble Learning
Abnormality Detection Result (A)
Abnormality Detection Result (N)
Dynamic Ensemble Weight Average Method
Optimized Detectors
Ensemble Model
Detection Phase
Test Data
Prediction Execution
Final Result

Figure 4.4: The proposed dynamic ensemble models for disturbance detection.

2. Up to this stage, it is necessary to have a mechanism to explain the cause of the detected anomaly. The one crucial reason is to reduce the bias of the detection and to increase the detection model reliability. The review shown in Chapter 2 shows that the methodologies for

performing root-cause analysis consist of three ways. For instance, forward problem analysis, inverse problem analysis, and using machine learning. The explanation of these methodologies in detail is shown in the following sections.

Forward Problem Analysis

As presented in Chapter 2, the definition of the “forward problem” is formulated from the model parameters (m) and sources (s) to observe output (o). This technique is similar to how the human solves the problem from the given environment input. The overall statement is formalized as Equation (4.11):

$$o = A_s(m) \tag{4.11}$$

where A_s is the forward problem depending on a source (s).

To solve the forward problem, the greedy algorithm and stack operator are used to construct the algorithm. This section’s methodology starts with the model using a greedy algorithm to pick up the daily routing plan (P) from the logistics agency. The anomaly degree and cost are then computed, respectively. Note that the procedures in this section are adapted from [15]’s study. The anomaly degree is computed as Equation (4.12):

$$Anomaly^{i}_{degree} = \frac{\sum_{i=0}^{n} Operation^{i}_{attr}(t)}{|S^{i}_{POA}|} \tag{4.12}$$

where Operation attribute ($Operation_{attr}$) denotes as utility and productivity record from the routing planning of the logistics agencies and S^{i}_{POA} is the set of Operation attributes in plan P. The operation attribute for this study consists of six attributes that are referred from the “Supply Chain Strategies” book written by [68]. The operation attributes consist of driver workload ratio (dwl_{ratio}), driver availability ratio (da_{ratio}), utilization of the fleet (uof), productivity of the fleet (p_{ratio}), incident ($Incident_{ratio}$) and shortage ($shortage_{ratio}$) respectively. Also, these attributes are presented as presented in Equation (4.13) to Equation (4.18).

The driver workload ratio (dwl_{ratio}) is a computation of the portion between the received request from customers and the vehicle’s availability in the fleet at a current time. This attribute demonstrated how well the drivers are assigned to work compared to the number of unassigned drivers represented via a number of available vehicles. Therefore, the dwl_{ratio} is computed as Equation (4.13):

$$dwl_{ratio} = \frac{Received_{reg}}{Vehicle_{avali}} \tag{4.13}$$

where $Received_{reg}$ denotes as a number requested of a vehicle from the customers and $Vehicle_{avail}$ denotes the number of remaining vehicles in the current fleet at time (t). The driver availability ratio computes the portion between the vehicle in use and the total vehicle in the fleet. This attribute demonstrated how much drivers are unassigned to work respected to the workload. Therefore, the da_{ratio} is computed as Equation (4.14):

$$da_{ratio} = \frac{Vehicle_{used}}{\sum (Vehicle_{avail}, Run_{anotherfleet}, Back_{orderwork}, Vehicle_{main.})} \tag{4.14}$$

where $Vehicle_{used}$ denotes the number of the vehicle used in the current fleet at time (t), $Run_{anotherfleet}$ denotes the number of drivers that are working in another fleet. Further, $Back_{orderwork}$ denotes the number of drivers who are not ready to receive a new task.

$Vehicle_{maintenance}$ denotes the number of the vehicle in maintenance, and $Vehicle_{avail}$ denotes as the number of available vehicles in the fleet currently at time (t).

The uof computes the portion between the total received requests from the customer and the fleet's total vehicle. This attribute aims to demonstrate how much fleet utilization performed when compared to the requests from customers. Therefore, the uof is computed as Equation (4.15):

$$uof = \frac{Received_{reg}}{\sum(Vehicle_{avail}, Run_{anotherfleet}, Back_{orderwork}, Vehicle_{main.})} \tag{4.15}$$

where $Received_{reg}$ denotes as a number requested of the vehicle from the customers, $Run_{anotherfleet}$ denotes the number of drivers that are working in another fleet. On the other hand, $Back_{orderwork}$ denotes the number of drivers who are not ready to receive a new task and has not finished their previous tasks, $Vehicle_{maintenance}$ denotes the vehicle in maintenance, and $Vehicle_{avail}$ denotes the number of the available vehicle in the fleet at the current time (t).

The p_{ratio} computes the portion between the actual requested and the successfully received request from the customers. This attribute aims to demonstrate the productivity of the fleet when performing delivery tasks. Therefore, p_{ratio} is computed as Equation (4.16):

$$p_{ratio} = \frac{Received_{reg}}{Actual_{reg}} \tag{4.16}$$

where $Received_{reg}$ denotes a number requested from the customers and $Actual_{reg}$ denotes the actual requested from the customers that submitted to the system. The incident ratio computes the unsuccessful delivery compared to all requests received submitted to the logistics agency. This attribute aims to determine the unsuccess rate and demonstrate how incidents and disturbances impact transportation planning. Therefore, the $incident_{ratio}$ is computed as Equation (4.17):

$$incident_{ratio} = \frac{incident}{Actual_{reg}} \tag{4.17}$$

where $incident$ denotes as a number of incidents and disturbances occurred while transportation planning is in process and $Actual_{reg}$ denotes as all of the requested from the customer that is submitted to the system. The shortage ratio computes the number of tasks that are not assigned to the vehicle compared to the total requested submitted by the customer. This attribute aims to demonstrate how efficient the transportation planning tasks are performed. Therefore, the shortage ratio is computed as Equation (4.18):

$$Shortage_{ratio} = \frac{Shortage_{vehicle}}{Actual_{reg}} \tag{4.18}$$

where $Shortage_{vehicle}$ denotes the number of tasks that cannot be served by the vehicle in the fleet and $Actual_{reg}$ denotes an actual request from the customers that submitted to the system.

After the anomaly degree of each operation's attribute was computed. It then used an input to compute the ranking (R) of importance to the corresponding anomaly. Note that this process is adapted from the page rank algorithm. The ranking is computed as Equation (4.19):

$$R(OA_i) = 1 - d + d\frac{\sum_{P_{day} \in P} Anomaly_{OA}^{degree} \times OA}{\sum_{day \in P} OA} \tag{4.19}$$

where $OA_1, \ldots, OA_N$ are the operation attributes under consideration. $Anomaly_{OA}^{degree}$ denoted how related of the anomaly to the operation's attribute (OA), d is the dampling factor that generally used in page rank algorithm. It ranges between the real interval [0,1].

Refer from Figure 4.5, the algorithm for performing forward problem is starting from the anomaly degree is first computed and then ranked it for the pickup OA. The author pickup OA by random pick and stored it to "*tmp*" list. Next, starting picking OA from the list and check whether or not it belongs to the routing plan (P). Second, the ranked of these two variables are compared. The comparison is split into two cases. Note that the ranking is ordered by Ascending order.

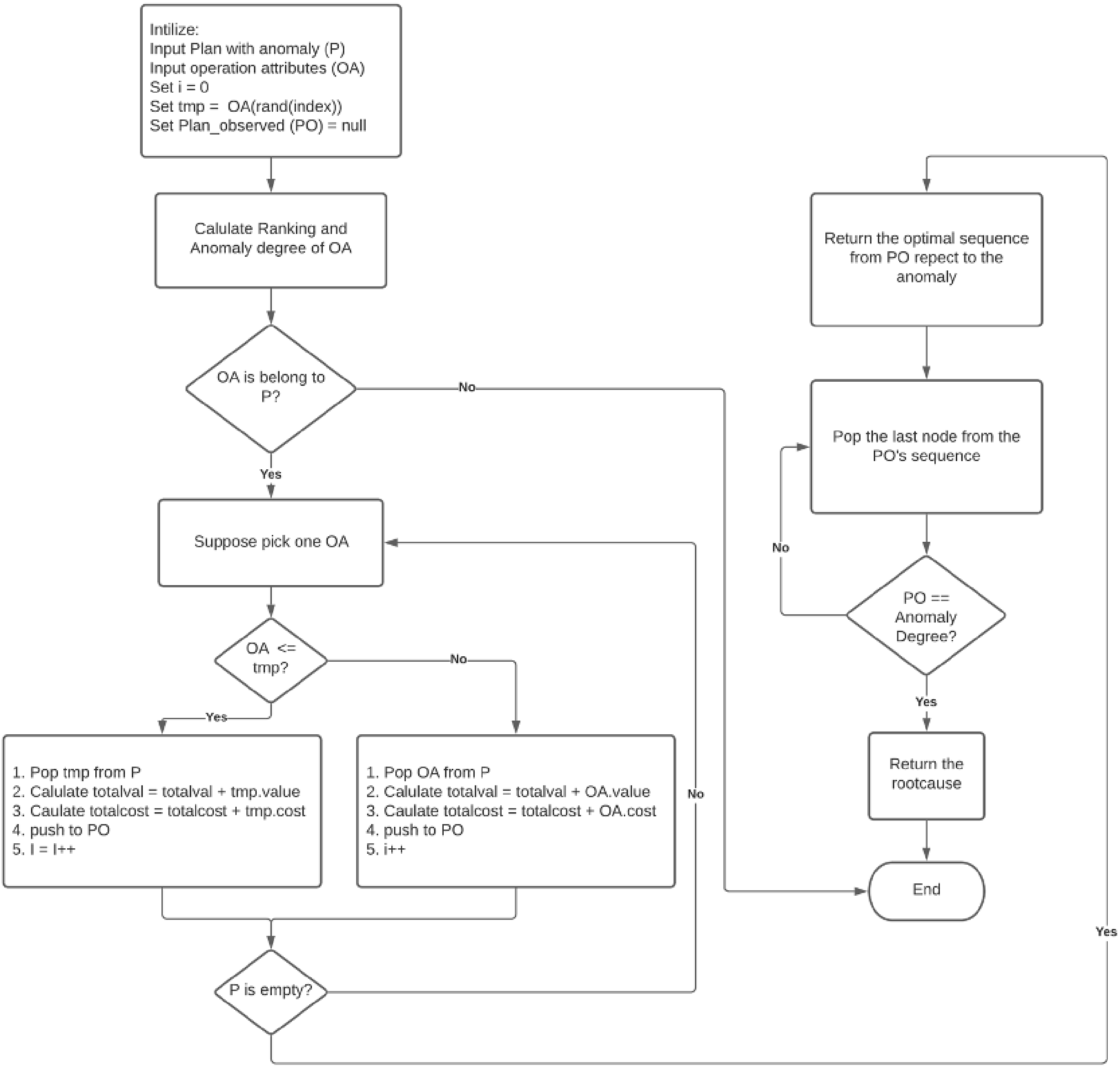

Intilize:
Input Plan with anomaly (P)
Input operation attributes (OA)
Set i = 0
Set tmp = OA(rand(index))
Set Plan_observed (PO) = null
Calulate Ranking and Anomaly degree of OA
OA is belong to P?
No
Yes
Suppose pick one OA
OA <= tmp?
No
Yes
1. Pop tmp from P
2. Calulate totalval = totalval + tmp.value
3. Caulate totalcost = totalcost + tmp.cost
4. push to PO
5. I = I++
1. Pop OA from P
2. Calulate totalval = totalval + OA.value
3. Caulate totalcost = totalcost + OA.cost
4. push to PO
5. i++
No
P is empty?
Yes
Return the optimal sequence from PO repect to the anomaly
Pop the last node from the PO's sequence
No
PO == Anomaly Degree?
Yes
Return the rootcause
End

Figure 4.5: The proposed methodology for forward problem analysis.

The first case is when the ranking of OA value is less than or equal to the *tmp* value. Therefore, *tmp* is taken from the routing plan list. The cost is then computed and insert together with *tmp*'s value into the "plan observed (PO)" list. The PO is a list that stored the attributes that makes comparison and computation already. The second case is when the OA

value is more than the value that stored in *tmp* then the OA is taken from the plan P list. The cost is then calculated and put together with OA into the PO list. The process is continuing until all OA in the plan (P) is computed. After the process is finished, the optimal sequence is returned. At this point, the algorithm took the last index of the sequences and check whether or not the value of OA is similar to the maximum of $anomaly_{degree}$ in PO. Suppose that it is equal, then the OA is then defined as a "rootcause" of the revealed anomaly. Otherwise, the next OA is taken for evaluation. Therefore, it is denoting as the origin of the sequence after the anomaly has occurred.

The overall process for performing forward problem analysis which converted Equation (4.11) into an algorithm is presented in Figure 4.5 and pseudocode in Algorithm 4.4.

Inverse Problem Analysis

In the previous section, the forward problem analysis for determining root cause was presented. This problem solving is similar to how humans tried to solve the problem from the given environment. Unfortunately, in this section, the way of problem-solving is different. Instead of analysis from the starting point until the abnormality is revealed. However, this problem solving is starting to solve the problem from the revealed abnormality and inverse back to the origin. Note that the procedures in this section are adapted from [15]'s study. As mentioned in the review section in Chapter 2, the inverse problem analysis is more complicated than the forward problem analysis. It is because of these reason as follow:

1. The returned solution is uncertain. In some cases, the solution is not returned.
2. The returned model's solution is duplicated with another model's solution. Therefore, it is not easy to differentiate the characteristics of each model from the given output.
3. The solution is based on approximation. Therefore it is required to have a benchmark to compare to increasing the solution confidentiality.

In general, the inverse problem is formulated as Equation (4.20):

$$s = A_s^{(-1)}(o) \tag{4.20}$$

where $A_s^{(-1)}$ is the inverse problem operator. This assumed that the source parameters are known. Next, in the solution of an inverse problem, there are 3 important questions that should consider [57].

From this point, the inverse problem analysis is converted to the computer programming as shown in Figure 4.6 and Algorithm 4.5. Note that the operation attributes and the anomaly degree are computed using Equation (4.12) to Equation (4.18).

Refer from Figure 4.6, the algorithm will pick up the operation attribute that has the maximum anomaly degree using a greedy algorithm and then pushed it into the data structure stack. While the plan is not empty, it is then continuing similar to the previous step until plan P is empty. After finishing these steps, the model then returns the optimal sequences that are respected to the anomaly. As a result, the root cause of the inverse problem is denoting as a last node of the sequence because it is input at the stack's first index. Therefore, it is denoting as the origin of the sequence before the anomaly has occurred. The overall process for performing inverse problem analysis, which Equation (4.20) was converted into an algorithm, is presented in Figure 4.6 and pseudocode in Algorithm 4.5.

Algorithm 4.4: Root-Cause Analysis by Forward Problem Analysis.

```
Input  : Dataset $P = P_1, P_2, \ldots, P_n$;
         Set of detection result $D_t = D_1, D_2, \ldots, D_n \in P$;
         Set of operation attributes $OA = OA_1, OA_2, \ldots, OA_n \in P$
Output: The analysis result of the root-cause ($Rootcause$); $Model_{accuracy}$
tmp ← rand($OA, Index^i$)
$Plan_{observed}$ = NULL
pred ← Array()
lastnode = 0
for $i = 1 : P.size$ do
    for $j = 1 : OA.size$ do
        $Anomaly^j_{degree} = Anomaly^i_{degree} = \frac{\sum_{i=0}^{n} Operation^i_{attr}(t)}{|S^i_{POA}|}$ %using Equation (4.12)
        $Ranking^j = R(OA_i) = 1 - d + d\frac{\sum_{P_{day} \in P} Anomaly^{degree}_{OA} \times OA}{\sum_{day \in P} OA}$
        % using Equation (4.19)
        $= OA^j$.concat($Anomaly^j_{degree}, Ranking^j$)
    end for
    $OA = OA$.orderby($Ranking$,asc)
    if $OA = 1 : P.size$ then
        for $i = 1 : OA.size$ do
            $OA = OA$.pop()
            if $OA.Anomaly_{degree} <= tmp.Anomaly_{degree}$ then
                $tmp = OA$.pop($tmp$.index)
                totalval = totalval + $tmp$.value
                totalcost = total cost + $tmp$.value
                $PO$.push($tmp$)
            else
                $tmp = OA$
                totalval = totalval + $tmp$.value
                totalcost = total cost + $tmp$.value
                $PO$.push($tmp$)
            end if
        end for
        while $lastnode.Anomaly_{degree} = max(PO.Anomaly_{degree})$ do
            $lastnode = PO$.pop()
        end while
        $pred.push(lastnode)$
    else
        $pred.push($“$No$”$)$
    end if
end for
$Rootcause = [pred_1, pred_2, \ldots, pred_n]$
$Model_{accuracy}$ = eval($P, pred^n$)
return $Rootcause$, $Model_{accuracy}$
```

Algorithm 4.5: Root-Cause Analysis by Inverse Problem Analysis.

```
Input  : Dataset $P = P_1, P_2, \dots, P_n$; Set of detection result $D_t = D_1, D_2, \dots, D_n \in P$;
         Set of operation attributes $OA = OA_1, OA_2, \dots, OA_n \in P$
Output: The prediction result of the root-cause ($Rootcause$); $Model_{accuracy}$
$tmp \leftarrow$ rand$(OA, Index^i)$
$Plan_{observed} =$ NULL
$pred \leftarrow$ Array()
$lastnode = 0$
for $i = 1 : P.size$ do
    for $j = 1 : OA.size$ do
        $Anomaly^j_{degree} = Anomaly^i_{degree} = \frac{\sum_{i=0}^{n} Operation^i_{attr}(t)}{|S^i_{POA}|}$ % using Equation (4.12)
        $OA^j$.concat$(Anomaly^j_{degree})$
    end for
    $OA = OA$.orderby$(Anomaly_{degree}$,dst$)$
    if $OA \in P$ then
        for $i = 1 : OA.size$ do
            $OA = OA$.pop()
            totalval = totalval + $OA$.value
            totalcost = total cost + $OA$.value
            $PO$.push$(OA)$
        end for
        $lastnode = PO$.pop()
        $pred.push(lastnode)$
    else
        $Pred.push($"$No$"$)$
    end if
end for
$Rootcause = [pred_1, pred_2, \dots, pred_n]$
$Model_{accuracy} = eval_{metrics}(P, Rootcause)$
return $Rootcause$, $Model_{accuracy}$
```

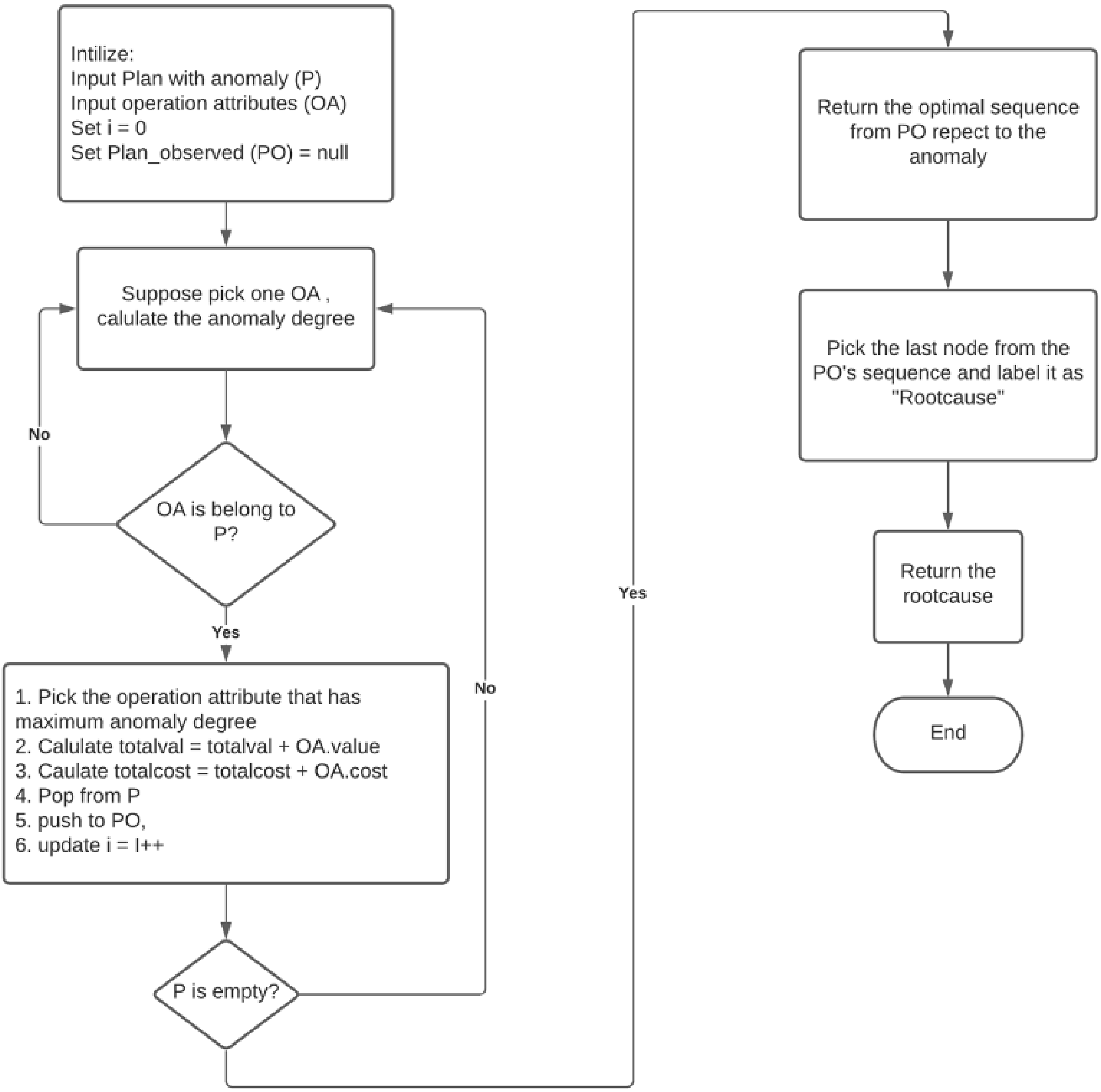

Intilize:
Input Plan with anomaly (P)
Input operation attributes (OA)
Set i = 0
Set Plan_observed (PO) = null
Suppose pick one OA , calulate the anomaly degree
No
OA is belong to P?
Yes
1. Pick the operation attribute that has maximum anomaly degree
2. Calulate totalval = totalval + OA.value
3. Caulate totalcost = totalcost + OA.cost
4. Pop from P
5. push to PO,
6. update i = I++
No
P is empty?
Yes
Return the optimal sequence from PO repect to the anomaly
Pick the last node from the PO's sequence and label it as "Rootcause"
Return the rootcause
End

Figure 4.6: The proposed methodology for inverse problem analysis.

Using Machine Learning for Root-Cause Analysis

After performing disturbance detection successfully, an experiment on a machine learning model using the probability distributions insides the model was performed. This model is used to discover the root cause of the disturbances. This method was adopted from Lu's framework [52]. They use machine learning for performing root-cause analysis. The motivation behind this experiment is to use the information organized from the disturbance detection as the root cause's input. The author also considered it as one feature in the root-cause analysis model combined with other features. The author believes that such information can help us perform root-cause analysis more feasible than perform these two tasks separately.

The reasons behind this statement are as follows. First, the data is non-linear. Second, the data points did not separate from each other. Therefore, The author uses the output from the disturbance detection model to filter the data. It also makes distinct data extractions. The assumption is that if the author inputs the data point directly without information of the disturbance into the root-cause analysis model, then the root-cause analysis model will not have any prior knowledge about the disturbance during training.

Consequently, the results will be a mix between normal events and disturbance data points. Moreover, it will not effectively distinguish between a normal event or a disturbance. To conclude, the model with the highest performance is recommended for root-cause analysis in practical scenarios. The crucial steps for this are as follows:

- The data and its features obtained from the feature engineering process were used in combination with disturbance status from the behavior analysis model. The status of the data points was trained and extracted. This status was then used to assist in the root-cause analysis process. In other words, the author pre-extracted information from the data and used it as one of the features to assist in the root-cause analysis task. The data was non-linear; thus, adding more features to the existing ones could improve the model's feasibility. It also added the capability to differentiate between disturbance and non-disturbance data points effectively in the clustering process.

  Then, numerous features are put into the Gaussian Mixture Model (GMM). These features are vital for clustering the data into different groups that have the same data characteristics. Five components in GMM were chosen for this study. These values were derived from information criterion techniques. The Akaike information criterion (AIC) was used. The AIC penalizes the model based on its complexity, as defined in Equation (4.21). Hence, there are five types of disturbances' root cause in the vehicle route optimization task.

  $$AIC(\theta) = -2 \log \Pr(X|\theta) + 2k \tag{4.21}$$

  For complete details, refer to [69]. The Bayes information criterion [70] is defined in Equation (4.22):

  $$BIC(\theta) = -2 \log \Pr(X|\theta) + k \log(n) \tag{4.22}$$

  Accordingly, the components that minimize both AIC and BIC were selected.

- The Probabilistic Neural Network (PNN) was then connected to learned and classified the data pattern stored by a PNN's pattern layer. These patterns are sum together in the summation layer. The PNN is a well-known ML model for doing pattern recognition, such as handwriting detection. Therefore, the author adapted the benefit of performing pattern recognition to the root-cause analysis problem.

  The model used in this study consists of four layers (e.g., the input, pattern, summation, and output layers). According to the structure of the data, the input layer consists of six neurons. They are used to indicates the dimension of the extracted input feature vector (F_1, F_2, ..., F_n). The pattern layer is fully connected. It consists of neurons that have six neurons similar to the input data. The summation layer followed the pattern layer. Lastly, at the output layer of the PNN, it gives a prediction result for each root cause. The prediction result was based on the probability distribution.

  The author used the softmax function to convert the output into a normalized form and selected the high-probability class as the final output. The transfer function F_i in the pattern layer is defined in Equation (4.23), where X_N is the input data, and σ is the smooth parameter. The author set σ to 0.1 according to the experiment. The hyper-parameter (σ) is used to control the model's smoothness. When the σ is very large, it is increasing the variance of the Gaussian density distribution. The Gaussian density distribution smoother the transition between different categories. In the summation layer, the Gaussian kernel of each known input is added from the pattern layer as $S_j = \sum_{i=1}^{n}(F_i)$, where i denotes the number of input data and S_j denotes the j_{th} S neuron output.

Finally, the output layer calculates the class probability as $y_j = \frac{1}{\sum_{i=1}^{n}(F_i)} \times S_j$, where n is equal to the size of input from the data and $i = 1 : n$ to obtain the classification output for the root cause. The label (output layer) is a 5-dimensional one-hot vector with two indicating normal, and the other three are the causes of abnormalities in freight transportation. Owing to the probability of the root cause's representation, at the output layer of the PNN, the author adds a softmax layer to convert the 5-dimensional output to 1. The PNN is trained with one-thousand epochs with sixty-four batch size.

For other models to compete with PNN, the author trained Deep-learning and Back-propagation Neural Networks with one-thousand epochs, three hidden layers and the same input and output as PNN. The extreme learning machine (ELM) was also trained in our experiment. The weight of the activation function has been analyzed as the data is propagated through the network to determine the root cause with possible probabilities.

$$F_i = \frac{1}{(2\pi)^{d/2}\sigma^d}\frac{1}{m}\exp[\frac{-(X_N - X_i)^{\intercal} \cdot (X_N - X_i)}{2\sigma^2}] \tag{4.23}$$

where i denotes the pattern number, m denotes the total number of training patterns, X_i denotes the i_{th} data that belongs to data category (C), σ is the smooth parameter (0.1), and d denotes the dimension of the measured space (1,2,3,4,5).

From the above defined equations, the overall model for the PNN is shown in Figure 4.7. The methodology is shown in Figure 4.8. The author also converted it to the programmable algorithm as Algorithm 4.6.

To train and test the model, the author used real operational data. The data between January 2017 to April 2019 were used for training and May 2019 to July 2019 for testing. The model of [52, 21, 5, 34, 19, 18, 43, 45, 49, 54, 15]'s were used as baselines for performance comparison and validation.

Also, each experiment is given 10 runs to make sure the sustainability of the results. The mean values of the metrics are presented. Note that in all experiments of this chapter, the author used Python 3.6 with the Keras library. The author also used the TensorFlow backend and ran on a google colab GPU.

After the experiments were conducted, the results are evaluated with the well-known evaluation metrics. Therefore, the detail of these evaluation metrics is presented in the next section.

4.2.4 Evaluation Metrics

In this section, the proposed model's detection capability was evaluated using the area under the ROC curve (AUC), precision and recall. The evaluation also including F_1-score. The author used these metrics due to the unbalanced data (e.g., with numerous percentage of actual negatives, such as data with normal and abnormal are not equally distributed) where the accuracy metric is not feasible to evaluate in this case. The AUC measures the whole 2-dimensional area under the whole receiver operating characteristic curve (ROC). The ROC

Algorithm 4.6: Root-Cause Analysis by ML.

```
Input  : Dataset $DF$ Set of detection result $D_t = D_1, D_2, \ldots, D_n$; Set of evaluation
         metric $E_m = E_1, E_2, \ldots, E_n$
Output: The prediction result of the rootcause : $Rootcause =$
         $\{Rootcause_1, \ldots, Rootcause_n\}$; $Model_{Accuracy}$
$Data_{Processing}$ = concat($DF$,$D_t$)
$Train$, $Test$ = split($Data_{Processing}$)
$Feature_{Train}$ = Makefeature($Train$) % Section. 4.2.1
$Feature_{Test}$ = Makefeature($Test$) % Section. 4.2.1
$Train$ = concat($Train, Feature_{Train}$)
$Test$ = concat($Test, Feature_{Test}$)
if $Train['status'] = 1$ then
    for i=1:N do
        [$Train^i$, Cluster] = GMM($Train$, $AIC$, $BIC$) % Section. 4.2.3
        $PNN$ = model([$Train^i$, Cluster])
        [$Rootcause_{Train}$, $index_i$] = PNN($Train$)
        [$Rootcause$, $index_i$] = PNN($Test$)
    end for
    $Accuracy_{train}$ = eval($Rootcause_{Train}$,$e_m$)
    $Accuracy_{test}$ = eval($Rootcause$,$e_m$)
    $Rootcause = [Rootcause^1_{anomaly}, \ldots, Rootcause^n_{anomaly}]$
    $Model_{Accuracy} = [Accuracy_{train}, Accuracy_{test}]$
else
    for i=1:N do
        [$Rootcause$, $index_i$] = 'Normal'
    end for
    $Rootcause = [Rootcause^1_{normal}, \ldots, Rootcause^n_{normal}]$
end if
return $Rootcause$
```

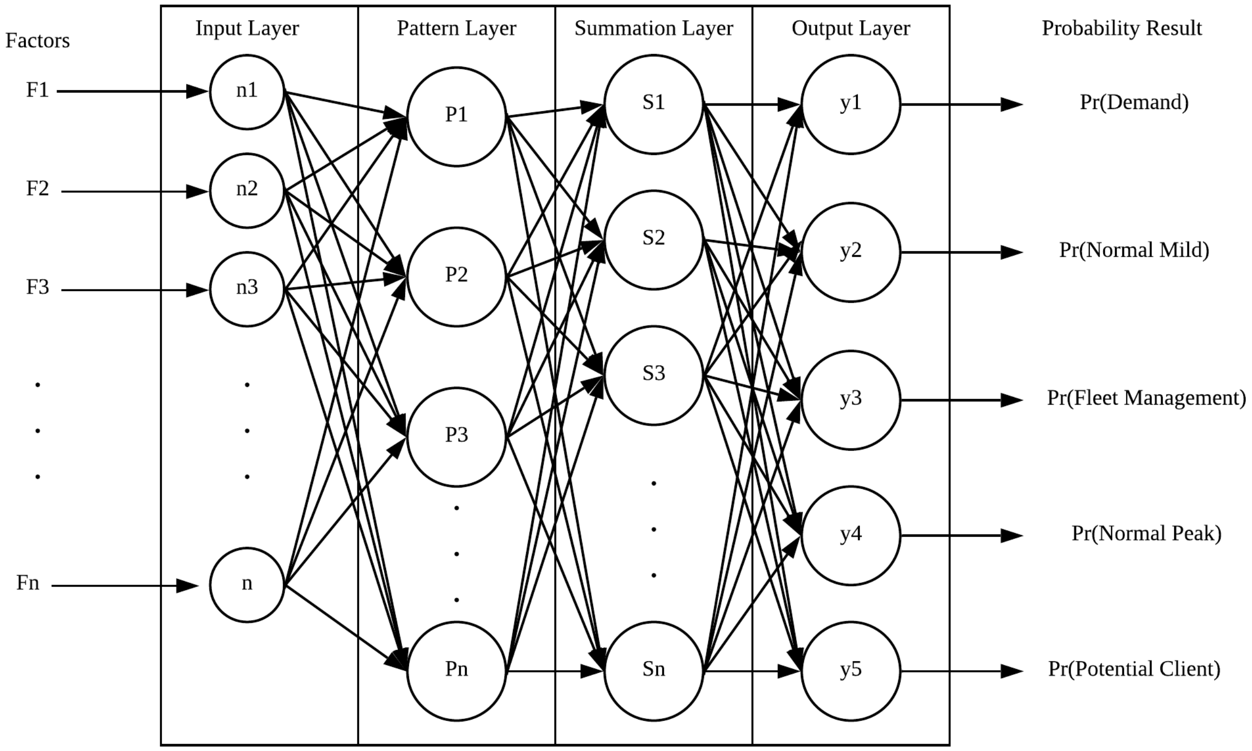


Figure 4.7: Demonstration of probabilistic neural network (PNN) for root-cause analysis with application in urban freight transportation planning.

was computed by applying a different threshold comparing True Positive Rate (TPR) and False Positive Rate (FPR). Thus, the higher of these values indicated that the model is efficient in detecting abnormal and normality events.

The author aims to demonstrate the practicality of the proposed methodology. Therefore, the experimental results were validated using the provided metrics with real data of disturbances in route optimization. These data were reported by company staff.

Once the methodology for detecting anomalies and root causes for transportation was developed, the author also wants to demonstrate how general the proposed methodology was when employed in other application domains. Therefore, the detail of the applications, modifications, and limitations is presented in the next section.

4.2.5 Practical Applications and Limitations

In the beginning, the proposed hybrid model was developed for the transportation domain. However, the author also wants to demonstrate that the proposed hybrid model was general and can be employed with minor modifications to other application domains. Therefore, other application data (e.g., credit card transactions, water manufacturing system, and computer network) were also used to performed anomaly detection and root-cause analysis. Note that these data are the same as presented in Chapter 3.

The anomalies in these applications' context are the system transactions that were different from the routine usage of users (e.g., growing usage than the usual or different usage period). These anomalies were detected by the ensemble of the detection results from the models in Chapters 3 – 4. The integration of the models is because the anomalies of credit card transactions were not effectively detected using only distance-, and density-based characteristics.

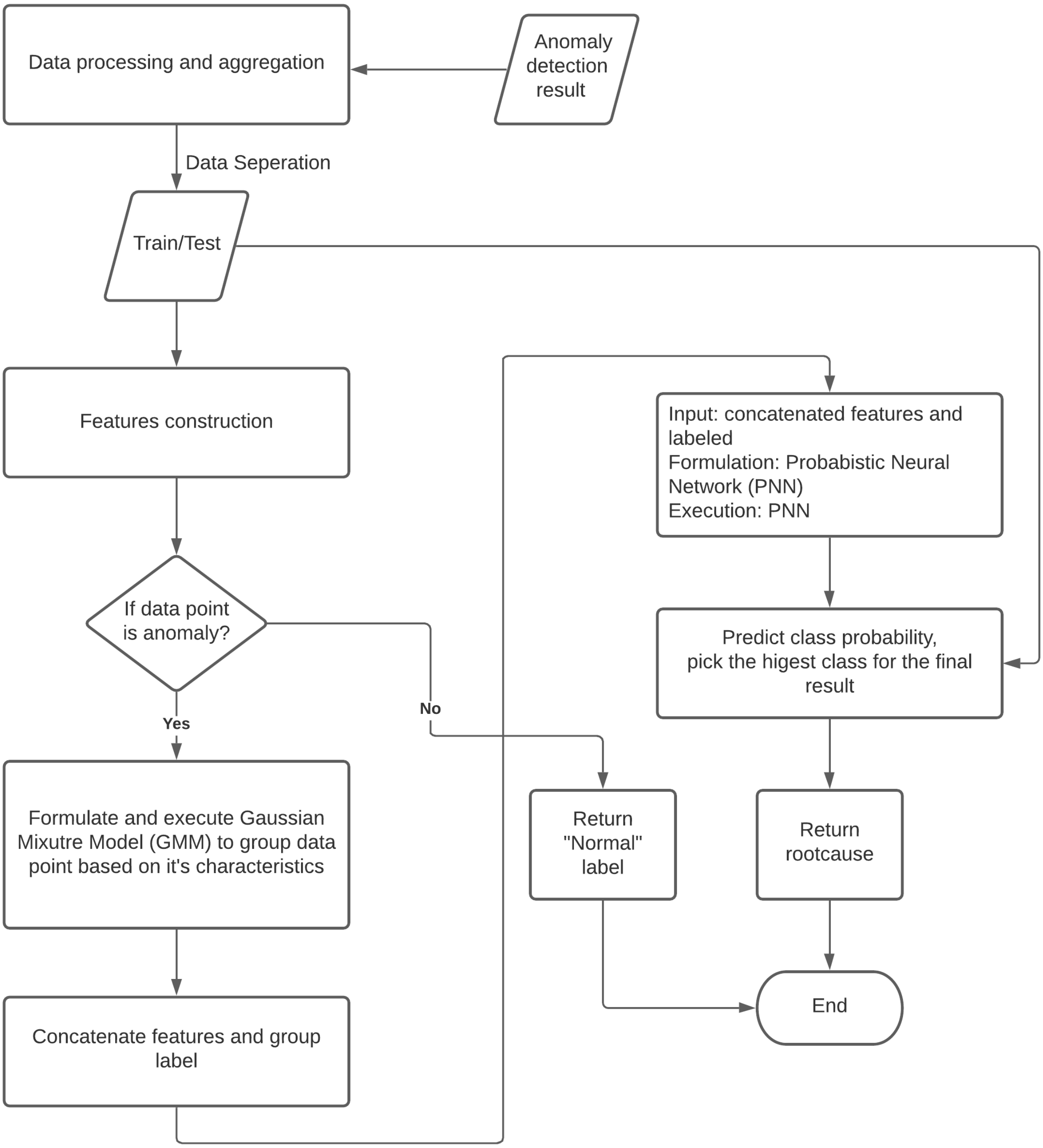

Data processing and aggregation
Anomaly detection result
Data Seperation
Train/Test
Features construction
Input: concatenated features and labeled
Formulation: Probabistic Neural Network (PNN)
Execution: PNN
If data point is anomaly?
Predict class probability, pick the higest class for the final result
No
Yes
Formulate and execute Gaussian Mixutre Model (GMM) to group data point based on it's characteristics
Return "Normal" label
Return rootcause
End
Concatenate features and group label

Figure 4.8: Methodology for performing root-cause analysis.

However, it also depends on the contexts, such as time and location, and conditional behavior (e.g., usage and frequency). Therefore, the root-cause analysis plays an essential role in dealing with the data characteristics at this stage. Similar procedures were also applied to the networking anomaly and water manufacturing dataset. The anomalies were the usage that concordant by time.

Before, the anomaly detection model detects anomalies from the data, which are subjected to preprocessing and standardizing. Then, these data were ordered based on time. The common attributes that are vital for this proposed model are the time context and behavior attributes. The behavior attributes consist of transaction usage and frequency. From this point, the author has demonstrated that each dataset is using the same data attributes for detecting anomalies. Therefore, the anomaly detection models were compared based on their efficiency by the area under the RoC curve metric.

Furthermore, the data have different characteristics. However, using only one model to detect anomalies may not be feasible because some data characteristics may be discarded. Therefore, in this study, the author combines model detection capability using ensemble learning methods and root-cause analysis to enhance detection efficiency and increase the interpretability of the result. The candidate detectors for each data characteristic consist of temporal-, distance-, density-, reconstruction-, and anomaly explanation-based.

The temporal-based detector is capable of detecting anomalies that are dependent on time. In addition, the distance-, density- and reconstruction-based detectors can detect characteristics of data points that can be measured by distance, density, and degree of reconstruction (e.g., input and output should be nearly identical). Finally, the anomaly explanation-based can detect anomalies that depend on behavior conditions (e.g., usage, frequency, and duration).

This part has driven the proposal of a hybrid model that combines the detection capability of each model. Thus, the model consists of detection and root-cause analysis stages. Furthermore, the author discovered that each model has a different capability in dealing with different data characteristics. Therefore, the integration of these capabilities can significantly enhance the final detection. This finding was also discovered by [44, 43].

The author believed that integrating these detectors' capabilities could increase the model's efficiency as it incorporates all aspects of anomalies represented in each data characteristic.

Finally, the limitation of the proposed hybrid model is that it can only handle the structured data ordered based on time as inputs. Therefore, data preprocessing is required when the data is unstructured. In addition, the feature engineering process is required before executing the model—for instance, the amount and the frequency of the usage ratio. These features are used for performing the root-cause analysis. If more features are to be input into the model, the model's input layer should be adjusted to meet the new data. Also, the model hyperparameters (e.g., epochs, batch size, learning rate, and more) should be optimized concordant to the new data input. Hence, the common attributes vital for this proposed model are the time context and behavior attributes, which consist of the transaction usage and frequency.

After the methodologies for detecting anomalies and root causes were presented, the experimental results of these methodologies are presented in the next section.

4.3 Results

In this section, the experiment results are presented in two sub-sections: disturbance detection and root-cause analysis. In addition, Figure 4.9 illustrated the training performance of the LSTM-AE based model with a learning rate 10^{-4}.

4.3.1 Behavior Analysis

Unsupervised Hybrid Anomaly Detection Model

In the previous section, the author performed anomaly detection for temporal and context disturbances. The author used the same dataset as presented in section 4.2.1. The difference is that this time the author detected the static disturbances from the dataset using joint learning between the data dimension reduction method and clustering technique for behavior analysis. The experiments were conducted with numerous models (e.g., PCA and AE). The author

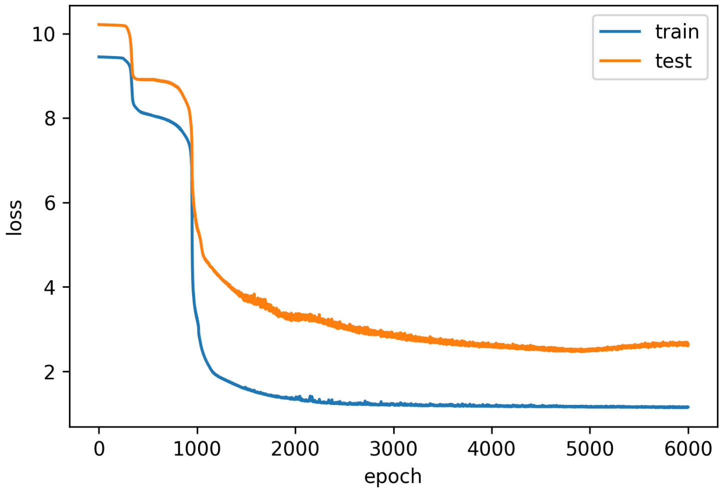


Figure 4.9: Training the model with multidimensional data.

utilized the one that returned the highest detection result for practical use. The experiment result is shown in Table 4.3.

Table 4.3: Experiment results when dimension reduction and joint learning methods were applied for detection models.

Model	AUC	Precision	Recall	F_1-score
Yang et al. [34]	0.546	0.425	0.557	0.482
Zong et al. [5]	0.790	0.76	0.722	0.741
Xu et al. [21] (Donut)	0.504	0.385	0.614	0.473
Hundman et al. [18]	0.572	0.714	0.189	0.3
Phiboonbanakit et al. [13]	0.870	0.836	0.842	0.839
Elsayed et al. [19] (LSTM-AE)	0.577	0.943	0.137	0.239
Average of Baselines	0.763	0.736	0.652	0.667
PCA with Kmean	0.543	0.585	0.152	0.241
PCA with One-SVM	0.5254	0.5	0.133	0.210
PCA with Isolate Forest	0.5613	0.667	0.177	0.280
PCA with GMM	0.658	0.523	0.722	0.607
PCA with GMM (estimator)	0.614	0.466	0.778	0.583
LSTM-AE with One-SVM	0.621	0.498	0.652	0.565
LSTM-AE with Isolate Forest	0.618	0.493	0.646	0.559
LSTM-AE with GMM	**0.790**	**0.76**	**0.722**	**0.741**

Dynamic Ensemble Weight Average Method

As shown in the previous sections, multiple models for detecting anomalies in transportation logistics operations were presented. It was shown that these models have different capabilities of detecting disturbances. Therefore, it is essential to integrate these detection results into consideration for final decision-making. Therefore, the dynamic ensemble weighted average method was used to perform this task.

In addition, the experiment on the state-of-the-art models was conducted and compared against the proposed model. The author aims to illustrate the practical significance of the proposed model. The experiment results are as shown in Tables 4.4 - 4.5. Note that the bold text models in Tables 4.4 - 4.5 are the models that applied the methodology of this study.

Table 4.4: Experiment results when the ensemble weighted average method was applied to combine detection's result.

Model	AUC	Precision	Recall	F_1-score
Chen et al. [45]	0.769	0.589	0.768	0.667
Chakraborty et al. [43]	0.756	0.882	0.536	0.667
Wang et al. [49]	0.355	0.245	0.323	0.279
Average of Baselines	0.627	0.572	0.542	0.538
Temporal-LSTM-AE with Kmean	0.734	0.806	0.551	0.655
Temporal-LSTM-AE with One-SVM	0.761	0.814	0.608	0.696
Temporal-LSTM-AE with Isolate Forest	0.778	0.866	0.614	0.719
Ensemble LSTM-AE	0.59	0.32	0.44	0.371
Temporal-LSTM-AE with GMM	**0.784**	**0.868**	**0.627**	**0.728**

After the suitable model was obtained, the author then conducting an experiment on other application data. The author aims to demonstrate the practical usage of the model in detecting anomalies with various application data. The data used in this experiment consists of credit cards transactions [63], networking [64], water manufacturing [65] and transportation logistics system dataset. Some of these data are open data from Kaggle competitions. The experiment results is illustrated as Figure 4.10.

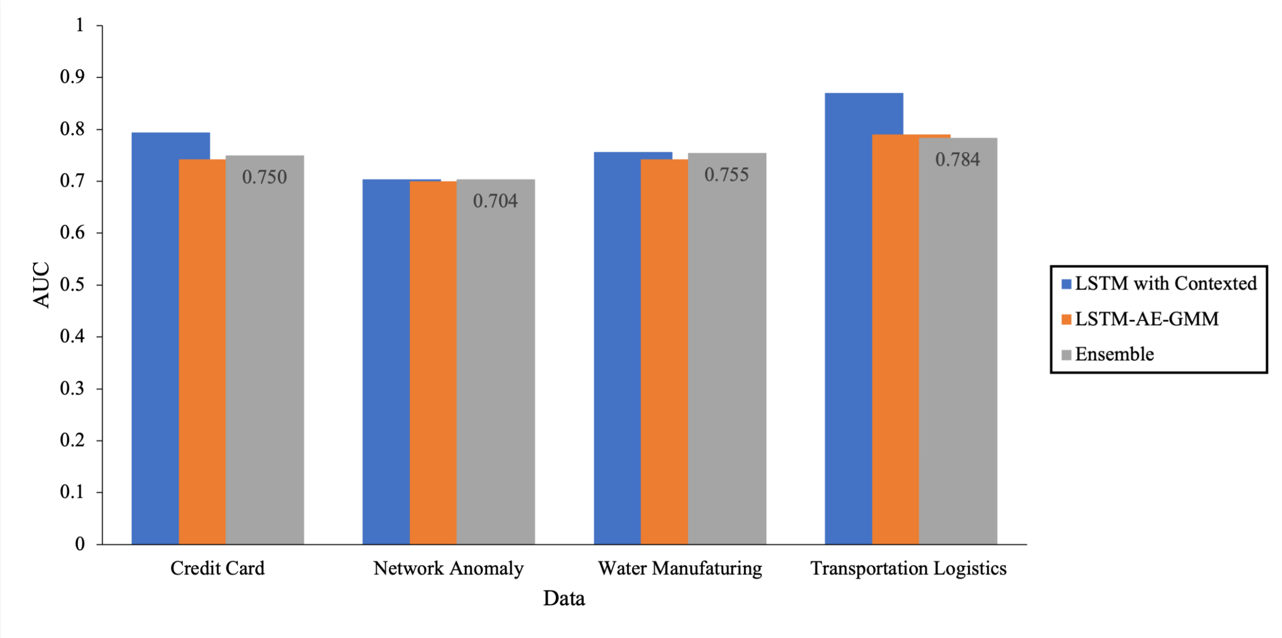


Figure 4.10: Experiment results when detecting anomalies using the proposed ensemble model on other applications data.

Table 4.5: Experiment results when the ensemble weighted average method was applied to combined detection's result with more than one classifier.

Model	AUC	Precision	Recall	F_1-score
Chen et al. [45]	0.769	0.589	0.768	0.667
Chakraborty et al. [43]	0.756	0.882	0.536	0.667
Wang et al. [49]	0.355	0.245	0.323	0.279
Average of Baselines	0.627	0.572	0.542	0.538
Detector Selection by GA	0.848	0.829	0.797	0.813
Detector Selection by PSO	0.848	0.829	0.797	0.813
Temporal-LSTM-AE with 2 GMM (EM/PCA-DATA)	0.839	0.826	0.778	0.801
Temporal-LSTM-AE with 2 GMM (EM/PCA-EM)	0.832	0.823	0.766	0.793
Temporal-LSTM-AE with 2 (GMM/Isolate Forest)	0.829	0.837	0.747	0.789
Temporal-LSTM-AE with 2 (GMM/One-SVM)	0.822	0.835	0.734	0.781
Temporal-LSTM-AE with 2 (GMM/Kmean)	0.827	0.842	0.741	0.788
Temporal-LSTM-AE with 3 (GMM(Data)/GMM(EM)/...)	0.839	0.826	0.778	0.801
Temporal-LSTM-AE with 4 (GMM/Kmean/IS/...)	**0.848**	**0.829**	**0.797**	**0.813**
Temporal-LSTM-AE with 5 (GMM/IS/SVM/...)	0.833	0.838	0.753	0.793
Temporal-LSTM-AE with 6 (GMM/IS/SVM/Kmean/...)	0.837	0.830	0.772	0.800

4.3.2 Root-Cause Analysis

Root-Cause Analysis using Forward Problem Analysis

In the previous section, the anomaly detection model was performed in the logistics operation data. The model returned the label of the data point. It was checked whether the data point is anomalous or normality. However, if the author wants to explain the reason why these data point was detecting as an anomaly, root-cause analysis is required to perform. In this section, the root-cause analysis using the forward problem analysis was performed. The result is shown in Table 4.6.

The effectiveness of the methodology was evaluated with the area under the curve (AUC), accuracy, precision, and recall, including the computational time when running the model of RA. To demonstrate the practicality of the methodology, the analysis results were validated with real data collected through interviews with the company staff. Table 4.6 summarizes the results of the AUC, accuracy, precision, and recall, including the computational time, when compared against state-of-the-art models such as [52], the deep neural network (DNN), backpropagation neural network (BP-NN), and extreme learning machine (ELM). Note that the bold text models in Table 4.6 are the model which applied the methodology of this study and choosing forward problem analysis to explain the cause of the abnormalities.

Root-Cause Analysis using Inverse Problem Analysis

In the previous section, the experiment on the forward problem analysis was presented. In this section, the approach is changed in an inverse way. The main difference is the starting of the process is starting from the detected anomaly and traces back to its origin.

In this section, the root-cause analysis using the inverse problem analysis was performed. The result is shown in Table 4.7.

The effectiveness of the methodology was evaluated with the area under the curve (AUC), accuracy, precision, and recall, including the computational time when running the model of RA. To demonstrate the practicality of the methodology, the analysis results were validated with real data collected through interviews with the company staff. Table 4.7 summarizes the results of the AUC, accuracy, precision, and recall, including the computational time, when compared against state-of-the-art models such as [52], the deep neural network (DNN), backpropagation neural network (BP-NN), and extreme learning machine (ELM). Note that the bold text models in Table 4.7 are the model which applied the methodology of this study and choosing inverse problem analysis to explain the cause of the abnormalities.

Root-Cause Analysis using ML

In the previous sections, the anomaly detection in logistics operation data was performed. Following that, the author performed clustering using Gaussian Mixture Model (GMM) to segregate the types of disturbances using their characteristics. The reason for the use of GMM is non-linear data that cannot be clearly separated [34]. Using Kmean could not counteract this issue. After clustering, all of the features, including the disturbance types, were used as inputs for the machine learning model, which uses weight propagation in the network. The information from the disturbance detection task was considered as input for the second model for determining the root cause from the flow of data. This section also presents the practical use of this methodology.

The author tested the accuracy of the proposed methodology by using a cross-validation strategy. Specifically, k-fold cross-validation was performed. The author evaluated the effec-

Table 4.6: Experiment results of root-cause analysis using forward problem analysis.

Model	Accuracy	AUC	Precision	Recall	F_1-score	Time(s.)
Groenewald et al. [54] using ELM	0.826	0.772	0.615	0.346	0.443	0.470
Lu et al. [52]	0.823	0.860	0.745	0.717	0.731	0.147
Average of Baselines	0.825	0.816	0.68	0.531	0.587	0.309
DNN-Keras	0.811	0.888	0.790	0.900	0.841	143.312
BP-NN	0.576	0.5	0.346	0.270	0.303	0.188
Forward Problem Analysis [15]	0.242	0.539	0.278	0.318	0.297	10.173

Table 4.7: Experiment results of root-cause analysis using inverse problem analysis.

Model	Accuracy	AUC	Precision	Recall	F_1-score	Time(s.)
Groenewald et al. [54] using ELM	0.826	0.772	0.615	0.346	0.443	0.470
Lu et al. [52]	0.823	0.860	0.745	0.717	0.731	0.147
Average of Baselines	0.825	0.816	0.68	0.531	0.587	0.309
DNN-Keras	0.811	0.888	0.790	0.900	0.841	143.312
BP-NN	0.576	0.5	0.346	0.270	0.303	0.188
Inverse Problem Analysis [15]	0.223	0.519	0.237	0.173	0.200	10.401

tiveness of the methodology with the area under the curve (AUC), accuracy, precision, and recall, including the computational time when running the model of RA. To demonstrate the practicality of the proposed methodology, the analysis results were validated with real data collected through interviews with the company staff.

Tables 4.8 – 4.9 summarize the analysis' results with the F_1-score, AUC, accuracy, precision, and recall, including the computational time. This experiment also compared the proposed model against state-of-the-art models such as [52], the deep neural network (DNN), backpropagation neural network (BP-NN), and extreme learning machine (ELM). Note that the bold text models in Table 4.8 are the model that applied this study's methodology.

Table 4.10, the author determined the root cause based on their characteristics and in Figure 4.11 demonstrates proportions of the root cause that cause difficulty in transportation planning.

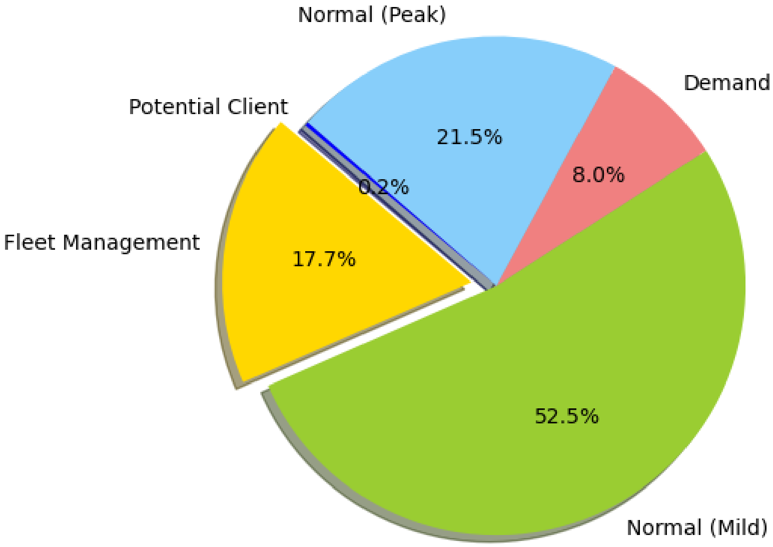


Figure 4.11: Proportions of the root cause of planning setting behavior in the freight transportation planning system classified by probabilistic neural network (PNN).

Finally, the result of using anomaly detection and root-cause analysis of the proposed model is also presented. It is presented in term of the effectiveness of detecting each type of anomaly, as shown in Figure 4.12 and Figure 4.13.

Besides, the author also performed an experiment to demonstrate the practical significance of the proposed model by adding anomaly detection in various applications. The data for testing is similar to Chapter 3. Thus, the experiment result is demonstrated as Figure 4.14.

After the experimental results were presented, it is now ready to discuss the findings from those experiments.

Table 4.8: Experiment results of root-cause analysis using using ML.

Model	Accuracy	AUC	Precision	Recall	F_1-score	Time(s.)
Lu et al. [52]	0.823	0.860	0.745	0.717	0.731	0.147
Groenewald et al. [54] using ELM	0.826	0.772	0.615	0.346	0.443	0.470
Average of Baselines	0.825	0.816	0.68	0.531	0.587	0.309
Cauteruccio et al. [15] Forward Problem Analysis	0.242	0.539	0.278	0.318	0.297	10.173
Cauteruccio et al. [15] Inverse Problem Analysis	0.223	0.519	0.237	0.173	0.200	10.401
DNN-Keras	0.811	0.888	0.790	0.900	0.841	143.312
BP-NN	0.576	0.5	0.346	0.270	0.303	0.188
Proposed Model (Hybrid root-cause analysis)	**0.833**	**0.888**	**0.829**	**0.896**	**0.861**	**0.080**

Table 4.9: The model's performance in detecting various root causes in transportation logistics operation.

Root cause	Precision	Recall	F_1-score
Fleet management	0.79	1	0.89
Demand	0.70	0.91	0.79
Potential clients	1	1	1
Normal (Mild)	0.99	0.78	0.88
Normal (Peak)	0.66	0.79	0.72

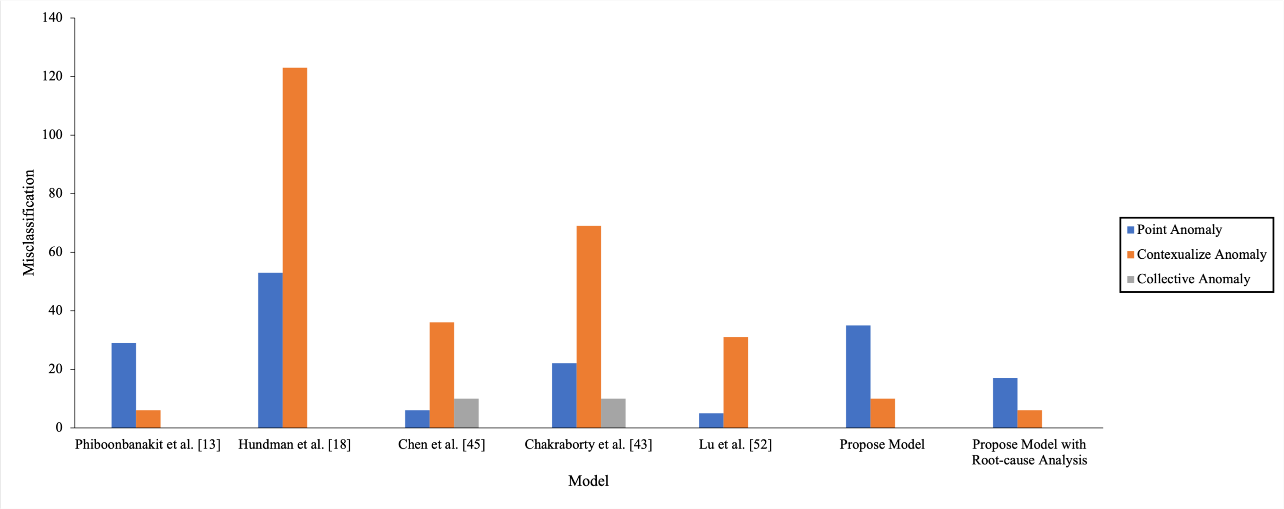


Figure 4.12: Misdetection of each anomaly types.

4.4 Discussion

4.4.1 Behavior Analysis

Unsupervised Hybrid Anomaly Detection Model

The comparison between two well-known dimension reduction techniques used to deal with high dimensional data is shown in Table 4.3. For the anomaly detection model, the transformed data were used as input. The PCA model was discovered that it did not outperform the LSTM-AE model. The reason being, in PCA, the components are essential to be optimally selected; otherwise, vital information will be lost, which may be required by the anomaly detection model. Additionally, we did not know which information is vital to the anomaly detection model as it requires a pre-training process.

Fortunately, it is not required to pre-training the LSTM-AE and remove any dimension. It is because the LSTM-AE encoded all data into a single encoded value. The encoded data represented the high dimensional data as single value data. Also, they are stored in the model's hidden layer. Hence, in each dimension, all the information was preserved. They also input into density- and distance-based detectors. Accordingly, the best fit detection model was the LSTM-AE with GMM. It provided the highest results in terms of AUC, precision, and recall, respectively. It also outperformed most baseline models. For instance: state-of-the-art models, One-SVM, and Isolate Forest. This phenomenon has reflected the benefit of dealing with non-linear data of the GMM. This capability is limited to other models (e.g., Kmean, One-SVM, and Isolate Forest). Furthermore, the proposed procedure from [5] was adapted by replacing

Table 4.10: The demonstration of characteristics of each root cause.

Factor	Fleet management	Demand	Potential client	Normal (Mild)	Normal (Peak)
Vehicle available	✓				✓
Incident			✓		
Fleet productivity	✓			✓	✓
Fleet utilization	✓			✓	✓
Maintenance cycle	✓				
Vehicle driver available	✓			✓	✓
Vehicle driver work load		✓		✓	✓

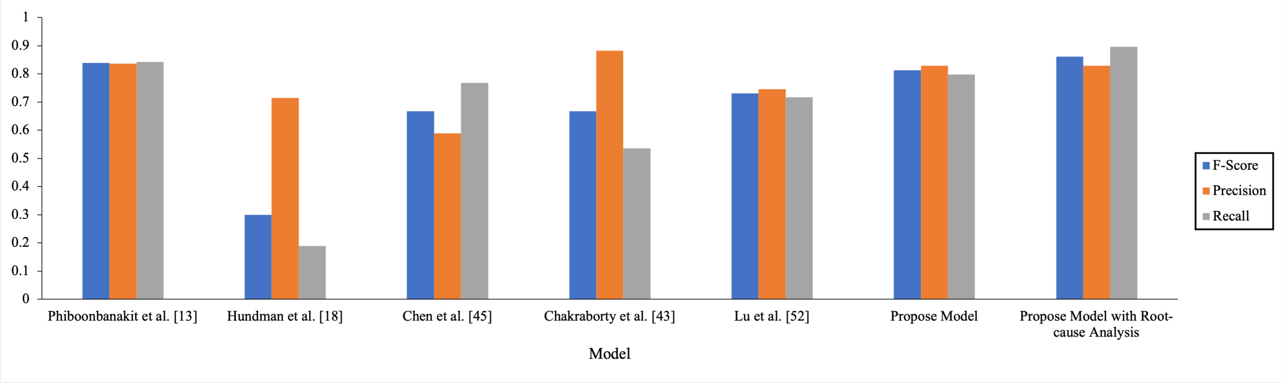


Figure 4.13: Performance comparison

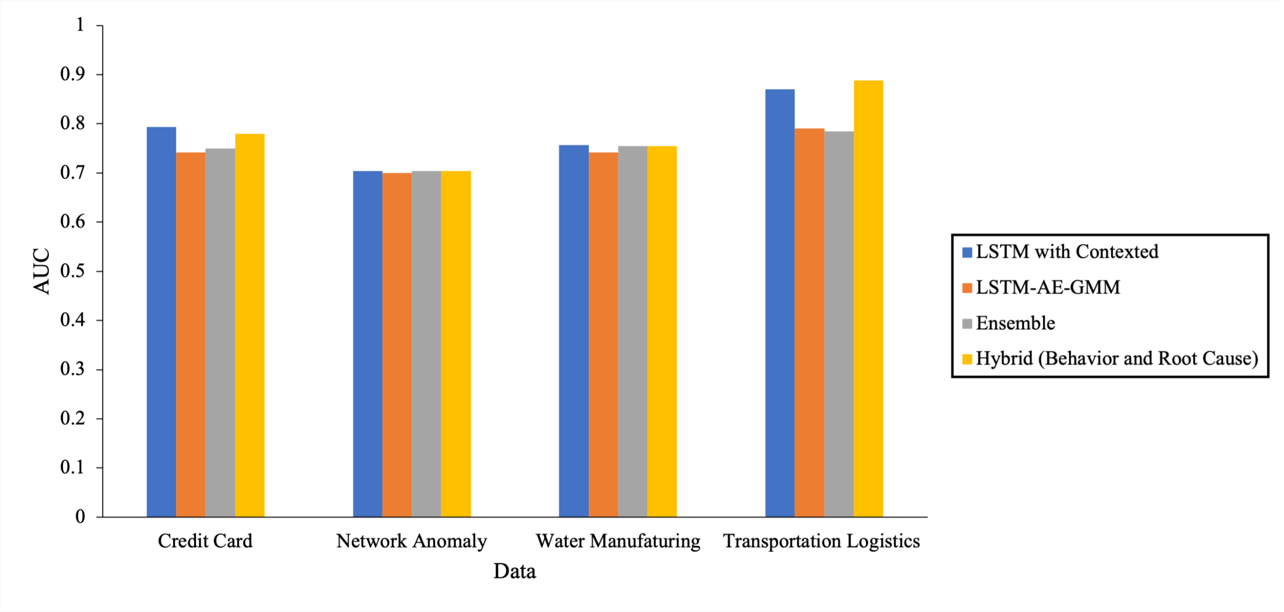


Figure 4.14: Experiment results when detecting anomalies and analyzing the root cause using the proposed hybrid model on other applications data.

Deep AE with LSTM-AE. However, the experiment result did not change significantly from [5]'s model. Unfortunately, the model of [34, 18, 19, 5] and the LSTM-AE with GMM are limited to detecting contextual anomalies. It cannot overcome the model of [13], as presented in Chapter 3. Therefore, [13]'s model is used as a based development for the dynamic ensemble model.

Dynamic Ensemble Weight Average Method

From Table 4.4, a model in which there is a balance between the temporal-context disturbance and static disturbance detection during daily transport logistics operations was obtained. The integrated output between the temporal, LSTM-AE, and GMM models is selected for practical use. The reasons to support this statement are as follows: First, GMM performed the best with the non-linear data. Second, this study used non-linear data; therefore, it is the limitation of Kmean, where it supports only linear data. On the other hand, non-linear data also has noise and unwanted data attributes. Therefore, it reduced the detection performance of the One-SVM and ensemble AE models. These models are not robust to noises from the data.

Further when the model of Wang et al. was applied to this study data. It also suffers from the problem that anomalies and normal events are not linearly separable from a high-dimensional data projection space. For simplicity, the distance of the data point from a radius of the decision boundary is similar for both abnormal and normal events, as shown in Figure 4.15. Therefore, the distance between points from centroid strategies used by most clustering methods (e.g., Kmean, k-nn, or its variants) cannot be used to determine whether the set of data points are an anomaly or not. As a result, this model has the evaluation metrics result (e.g., AUC and F_1-score) lowest among all models.

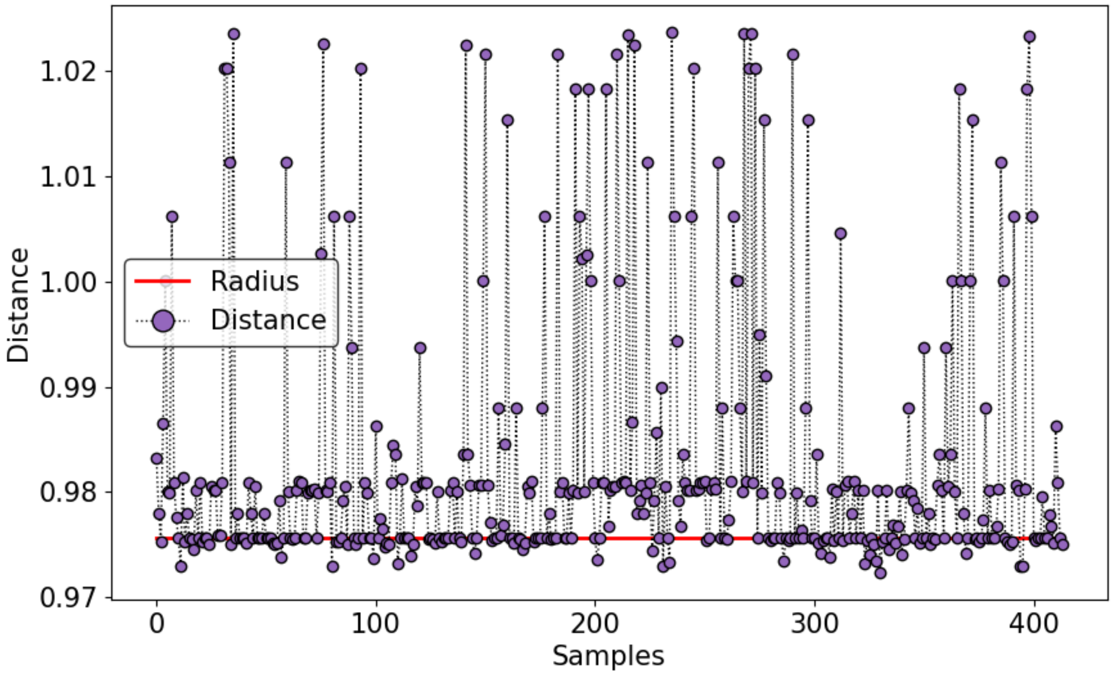


Figure 4.15: Demonstration of the use of Support Vector Data Description (SVDD) for detecting anomaly in the transportation logistics dataset.

Finally, disturbances in the logistics environment are correlated with location and time (spatial and temporal) contexts. Therefore, using the only ensemble of density- and reconstruction-based detection is not practical to detect disturbances that correlated with those contexts. These were the limitation of [45, 43]'s studies.

Fortunately, in the proposed methodology, the author ensemble each model's output together. It also includes the output of temporal detection from the LSTM with dynamic contextualized thresholds, as presented model of Chapter 3. Therefore, Table 4.4 depicted that the proposed approach was successful. It is because the detection rate increases after including temporal detection in the anomaly detection framework.

It also proved the assumption that the disturbances in transportation logistics are correlated with behavior attributes. Further, the result also overcomes the single model shown in Table 4.3 and the ensemble model of [45, 43]'s studies.

Unfortunately, the recall metric still had room for improvement. Hence, more classifiers were added to the model, as depicted in Table 4.5.

From the experiment result, the author discovered that the number of classifiers in the model is significant. As mention previously, each model has various capabilities to detect different types of an abnormal events. This finding shows the reason why some proposed model has low accuracy than some baselines in Table 4.4. Therefore, when the outcome of each model was combined, then the detection rate is increasing. As presented in Table 4.5, when the author added more classifiers, it improved all evaluation metrics. Especially the model titled "temporal-LSTM-AE with 4 (GMM/Kmean/IS/...)". It provided the best detection rate.

To prove this claim, instead of using the dynamic ensemble method for selecting detectors and combine detection results, the author proposed to used GA and PSO. The detector selection by GA and PSO and ensemble the detection result approach are presented. The experiment results show that the optimal detector recommended by GA and PSO provided the most efficient anomaly detection with 5 detectors. It is the same as a result suggests by the proposed model.

Therefore, the model titled "temporal-LSTM-AE with 4 (GMM/Kmean/IS/...)" is confidently selected for the final disturbance detection. Its detection result is also further used as input for the root-cause analysis model.

Unfortunately, Figure 4.10 shown that when the LSTMs join learning with clustering was applied, it reduces the overall detection performance for all datasets. This phenomenon occurred because the LSTMs join learning with clustering is truly unsupervised learning. It does not have any assumptions about the abnormality in terms of contexts and behavior attributes. The model is classified abnormality based on the data point density only. Suppose that the data point does not belong to any clusters; in this case, it then labels that data point as an anomaly. However, in the real-world application, the data are correlated with location–time contexts. Also, the behavior attributes such as resource usages concerning the time. Therefore, the LSTMs join learning with the clustering approach are not capable of handling this kind of problem.

To enhance the detection result, the dynamic ensemble method was applied to combine the temporal anomaly detection model from Chapter 3 with an LSTM-AE with the clustering model presented in this chapter. As a result, the detection rate is increasing. The rationale behind the improvement is that the proposed dynamic ensemble method enables the model to decide which detector is used to classify the data. For instance, temporal with context, density-based, or both of them. Therefore, the data point is labeled individually depending on its characteristics. As a result, the detectors' majority vote is used to assigns the label. For simplicity, suppose that we have five detectors, then three detectors defined the data point as an anomaly. On the other hand, two detectors defined data points as normal. Therefore, the final detection result is returned as the data point is an anomaly. It is because most detectors are detecting it as an anomaly.

However, it still has room for further improvement. In Figure 4.10, shows the disadvantage of using ensemble methods. If the detector results conflict with each other. They cause degrading the overall detection result. Therefore, in the following section, the author continued to improve the model. Also, the root-cause analysis or anomaly explanation is considered to enhancing the detection result. The author aims to reduce the misdetection rate and bias when the detectors are performed. These actions strengthen the anomaly detection results. This process also included adding the ability to detect and eliminate the anomalies until their origin before they cause any disruption to the transportation planning processes.

4.4.2 Root-Cause Analysis

Root-Cause Analysis using Forward Problem Analysis

The result in Table 4.6 shows that the forward problem analysis cannot outperform the root-cause analysis using ML is because of these two reasons. The first reason is that the model ranked the degree of an anomaly in the ascending order. Therefore, the computation is selected the data point to be expanded base on the maximum ranking value. The characteristics of selecting data points are the same as a greedy algorithm.

The greedy algorithm is efficient and fast. However, it has some limitations. The heuristics function selects the closest node that has maximum value to be expanded. This action leads to the uncertain of the result because the computation can lead to the dead-end or be caught in the infinite loop.

The second reason is that it degrades the model's efficiency, as the algorithm chooses the closest data point with maximum value to expand. In some cases, the closest maximum data point is not always the origin that causes abnormalities. Therefore, it is increasing the chance to define a root cause with an incorrect searching location.

From the disadvantages mentioned earlier, the forward problem analysis also degrades the overall detection result because of the anomaly's misclassification. The result shows that it has only 0.539 of AUC. On the other hand, the precision and recall are on 0.278 and 0.318, respectively.

The author would suggest that this model's efficiency is not practical to derive a root cause in practical usage. The reason is that the lower of the precision and recall denotes that the model has a limitation in capturing the data characteristics and behavior from the input data.

Root-Cause Analysis using Inverse Problem Analysis

The result in Table 4.7 shows that the inverse problem analysis could not outperform the model proposed by [52]. The main difference between the inverse problem analysis and the forward problem analysis is that the inverse problem analysis used approximate strategies to derive the solution. The model started from the outcome and estimated the connected causes—however, the forward problem analysis using exact strategies for deriving the problem. The model starts based on the observed environment and traces to the effect of the cause. Therefore the solution is returned based on different assumptions.

The model that is used ML is also based on an approximation of the solution. The rationale behind that makes ML performed better than the inverse problem analysis is that the ML was trained based on the real data and had a label the characteristics of the cause of the events. On the other hand, the inverse problem analysis is based on search strategies. Therefore, it does not have any clue or assumption about the events. Unfortunately, the result is purely based on searching strategies and does not guarantee that the returned solution is actually the cause of the abnormalities.

From the issues mentioned earlier, the inverse problem analysis degrades the overall detection result because of the anomaly's misclassification. The result shows that it has only 0.519 of AUC. On the other hand, the precision and recall are on 0.237 and 0.173, respectively. Similar to the forward problem analysis, the author would suggest that the efficiency of this model is not practical for derive a root cause in the practical usage for the real-world problem.

Root-Cause Analysis using ML

From Table 4.9, the experiment result shows that the PNN in the proposed methodology was the most effective in determining the cause of the abnormal event in urban transportation logistics.

The PNN is a well-known ML model for doing pattern recognition. Therefore, the author adapted the benefit from pattern recognition to the root-cause analysis problem. Instead of searching for the relationship between data attributes like inverse and forward analysis problems, the PNN stored the route optimization process pattern in the pattern layer and summed it together in the summation layer. The model further makes classification the type of root cause using a probability distribution. As a result, the root cause from the route optimization process is returned.

It is also simple, with only 0.080 s. of computational time. To discuss the root-cause analysis further, the PNN is selected and recommended for practical use. In the following statement are reasons for the selection of PNN instead of other approaches. The first reason is that the PNN is an uncomplicated and efficient network. It has a high performance in computation.

The transfer function at the pattern layer is a Gaussian function that can obtain local approximation with low computational time. It does not require any backpropagation training operations. In contrast, classic neural networks and deep neural networks require more effort to fine-tune hyper-parameters. Hence, they are not capable of small datasets. The second reason is that traditional classification and regression models, such as CART, did not return the output's probability. Therefore, it is not feasible to analyze the data probability distribution throughout the process. In other words, the PNN is more transparent to understand the model's behavior and underlying relationships between the model's input and output. These components built up the case for the root-cause analysis than other models. In theory, it is similar to the general additive model (GAM). Therefore, the output at the PNN's output layer is influenced by the input of data pattern while the classification is performed. The stored pattern is propagated from pattern to summation layer. The relation behind the effect to the output was also understandable by the users. The third reason is that rule-based statistics provided an acceptable explanation for the class probability of root cause, as shown by [55].

Unfortunately, the relationship between data characteristics is not always a linear correlation. It is not possible to create all possible rules to handle this type of data distribution. On the other hand, the Bayesian Network is also well-known for performing root-cause analysis. However, it requires supporting data and prior knowledge to determine the influence output, as shown in the study by [71]. Thus, this is the reason why the author used GMM to group the data pattern by its characteristics first before inputting it into the PNN model instead of creating all possible handcraft rules.

Furthermore, the proposed methodology also outperforms [52]'s framework in determined the abnormality root cause. It is because of the efficiency in detecting abnormalities in the series of data. In [52], the process done separately between anomaly detection and root-cause analysis. Therefore, it is difficult to obtain the root causes when the data points are overlap to each other.

In this study, the joint learning between anomaly detection and root-cause analysis is performed. As a result, the information is shared between these models to determined the root cause.

Table 4.9 presents how well the PNN classified the root cause each day from analyzing the probability of the stored data pattern throughout the network. Fortunately, the performed clustering process is able to separate data into groups.

The data are finally classified as five types of root cause and the factors that influence the root cause are shown in Table 4.10. Three of those are causes of disturbances to route optimization (e.g., fleet management, demand, and clients) and the remaining two are normal cases. The difference between "Normal (Mild)" and "Normal (Peak)" is the usage of vehicles where mild means the usage is not more than half of the fleet's capacity at the current time, whereas peak means the usage is more than half of the fleet's capacity. However, it has not yet exceeded the capacity limit. For example, on a certain day, the cause of the disturbance was the fleet management (e.g., inefficient route assignment for delivering goods to customers because of the shortage of vehicles due to maintenance or absence of the driver).

The results show that the root cause was efficiently determined with up to 0.79 precision and 1 recall. Furthermore, the author examined the case where the disturbance is caused by the demand overflow capability of the fleet. The results, in this case, were similarly effective, with up to 0.70 and 0.90 of precision and recall, respectively, including the normal case where it efficiently detected the normal situations out of the abnormalities. The results are satisfactory. However, some issues are worth noting: For the human resource management case, the result seems to be accurate. However, its frequency of occurrence is very low, as shown in Figure 4.11. Thus, further experiments need to be conducted for the cases where it happens more often.

In this section, the effectiveness of combining anomaly detection and root-cause analysis together is presented. This approach is also recommended for a practical use. It is shown that combining hybrid anomaly detection and root-cause analysis can enhance the detection result. As in Figures 4.12 – 4.13, the proposed model that performing hybrid detection (e.g., point, contextual, and collective anomaly) is outperformed other models. It is because the number of misclassification is reduced for all types of the anomaly. The rationale behind this is that the root-cause analysis part revealed the cause of the abnormality.

It led to the discovery of the origin components that are the cause of the anomalous. The anomalous can be eliminated at the root cause before it occurred and cause any disruption to the overall route optimization process. This also solved the problem of detecting data points that are depending on the behavior attributes. The contexts and behavior attributes must appropriately define. Therefore, the data point is labeled with respect to the origin of the causes.

As a result, the model returned the detection result more accurately than the model that determines the anomalous based on distance, density, and statistical analysis. On the other hand, in the detection part, the model can detect conditional behavior. Therefore, when these two capabilities are combined, the anomaly's data points are mostly detected and free from bias on labeled the data. In short, the user can have confidence in the detection result. As the decision of the detection is being explained, it makes the user to clearly understand the relation why the model reached this such decision.

It was also shown that the model based on ML is more practical and efficient in revealed the cause of abnormality than the exact (e.g., Forward Problem Analysis) and approximate (e.g., Inverse Problem Analysis) approaches. Nevertheless, the proposed model is typically general and can also be used for any other anomaly detection problems with minimal changes. For instance, minimal changes in location-time contexts (e.g., event at the location and specific time) and thresholds of behavior attributes (e.g., resources usage, amount of workflows, and transactions). Figure 4.14 demonstrated that using anomaly explanation enhances the final detection result. The experiment results improve from the prior detection models between 0.7 – 0.9 of AUC for all real-world applications. The results are admirable. It was demonstrated that the proposed hybrid model of this study is simple and only minor modifications were required when used in other applications domain. It can be utilized with other applications while the model still maintains a high computational performance.

The rationale behind this success is that the detection result is enhanced by combining numerous detectors with different capabilities in detecting anomalies. Also, the proposed hybrid model is capable to reveals the root cause of the anomalous. Therefore, the chance of assigning the anomalies labeled with bias and mistaken is reduced.

It is clear that performing anomaly detection is crucial to consider the anomalies' context and their causes. Suppose we compare to humans. This process is similar to when we consult the experts to get supporting information before making a decision.

Further, each expert has a different level of expertise in this research field. Therefore, the more result that returns in the same direction made the decision to be more confident than consult only a single expert. The decision is also free from bias.

From the experiments, the author discovered that combination of detectors and revealed its root cause could enhance the anomaly detection performance. The procedure makes the final detection result to be more accurate and reliable than the single model approaches. While the detection's decision is being explained, it makes users clearly understand the relation why the model reached this such decision. Also, the anomalies are detected and eliminated until their origin. It is done before they cause any consequence failures to the transportation planning processes. This chapter's finding provides a direction to investigate more in improving anomaly detection with anomaly explanation using ML for further studies. It also contributes further to a path for knowledge discovery.

Figure 4.11 demonstrates that the root cause of difficulty in transportation planning lies mostly in the management of the vehicle fleet, as it has a higher proportion when compared to that of other causes such as potential client or demand from the market.

For this methodology's practical usage, the logistics agency can use the proposed methodology to analyze the root cause of the detection stage's outcome. Therefore, the disturbance can be eliminated at its origin, and also, the cause of the disturbance is revealed. For instance, if the disturbance is detected, the PNN returns the weight from the activation function caused by disturbances. The weight which is higher than others denotes that the features in which the weight is associated are the cause of the disturbance.

However, there may be few limitations for analyzing the root cause, such as the overlapping of some factors. More factors have to be segregated for it to be clearer. Although the data which the author obtained contained full information, it was not sufficiently rich given that all the data needed to be linked together. However, it is not easy to analyze all root causes' types by using only sensor data and reports. This is because some root causes such as driver availability and delivery accuracy have environmental factors behind them.

Moreover, as some of the factors are dependent on humans, it results in unpredictability and difficulty in determining the root cause. This is the reason the author focuses on determining the root cause of disturbances for route optimization through the proposed model. The goal is to assist the decision-makers in their final decisions through proper evaluation of their transportation planning.

After the experimental results were discussed, it is now ready to conclude this chapter's contents, as shown in the conclusion section.

4.5 Conclusion

This chapter presents an unsupervised methodology for detecting anomalies and their root causes in urban freight transportation. This methodology aims to fulfill the conventional approach to dealing with anomaly detection problems, as mentioned at the beginning.

In this study, the author only concerning anomalies in route optimization tasks. From the results, there are two mains findings. First, the experiment result shows that the proposed methodology can detect a disturbance and involving root cause in the route optimization process with up to 0.88 of AUC. Hence, the proposed methodology can precisely determine the root cause. At the same time, it can enhance the detection result compared to state-of-the-art models. The reason is that the model took the benefit of performing dynamic two-stage analysis. This analysis perfectly revealed vital feature representations of the data. Therefore, the ability of the classifiers is enhanced. These classifiers are used to detected anomalies in the high-dimensional data.

Second, using a hybrid model provided the highest analysis results without additional monitoring and prior knowledge. The model is capable of handling non-linearly separable and conditional dependent data. It also biases-free when the detection result is finalized. The root-cause analysis part provides essential information to classify the data correctly. It also enhanced the overall detection performance. Thus, the detection result is more reliable than only doing detection of anomalous based on density and distance. As the detection's decision is being explained, it makes the user clearly understand the relation why the model reached this such decision. It also added the ability to prompt staffers to eliminate the anomalous at its origin. It is done before the anomaly occurred and caused any consequence failures. To strengthen the finding of this study, these experimental results are validated with actual operational data provided by company staff.

The experiment also showed that the proposed hybrid model is general and can also be used for any other anomaly detection problems with minimal changes.

Finally, this study has certain limitations. It only can handle structured data and delimit to the transportation research domain. Therefore, additional data preprocessing procedures and external operational factors are required for further considerations. The proposed methodology has reduced an amount of noise in the data, but it is still observed at a low percentage.

In the future, the author will improve the proposed methodology. Also, the author will consider more complex scenarios. For instance: dealing with production and manufacturing processes besides transportation planning. The author believes that doing so will make the proposed methodology more robust for behavior and root-cause analysis in real-world applications. The author will also apply this behavior analysis methodology to detect the reinforcement learning (RL) agent's behavior. The RL agent will perform the vehicle route optimization tasks in the entire operation instead of humans, as presented in Chapter 5.

Chapter 5

Case Studies of applying Behavior and Root-Cause Analysis for Transportation Planning

5.1 Introduction

In this chapter, case studies applying the methodology of behavior and root-cause analysis are presented. The case studies consist of routine vehicle route optimization tasks and when disturbances occur while vehicle route optimization is performed.

This chapter aims to present the practical and significance of behavior and root-cause analysis that assist in the transportation planning process, especially vehicle route optimization tasks. Vehicle route optimization tasks for logistics have become a crucial task since the rise of the digital eCommerce market. Therefore, it is necessary to have accurate planning and policy development. Therefore, in this chapter, the benefit of the proposed methodology is presented and demonstrated to enhance the accuracy and improve the vehicle route optimization solution.

The study assumed that a transportation planning agent is developed based on Reinforcement Learning (RL). They have equal responsibility as humans, starting from the initial learning stage. Therefore, each agent is trained based on the information provided by the methodology of behavior and root-cause analysis, described in Chapters 3 – 4. The author observes that if the information obtained from analyzing the transportation environment is instructive, it will improve the optimization solution suggested by RL when dealing with uncertain environmental changes.

The motivation for these case studies can be attributed to the increasingly complicated vehicle route optimization tasks of recent years. In addition, they are considerably affected by uncertain environmental changes (e.g., the sudden change of customer demand, road-network traffic conditions, and fleet resources).

However, traditional approaches cannot be applied because they are limited to extracting and monitoring the operation characteristics from multisource data. This results in the vehicle route optimization task not responding dynamically to uncertain situations or providing targeted solutions. Therefore, the proposed methodology of behavior and root-cause analysis presented in this study aims to enhance the accuracy of the information provided by the multisource data. The normality and abnormality of the transportation planning process are then revealed. Finally, the explanations of the cause of the abnormality are also used to assist in improving the detection result.

After the information is obtained, it contributes to the RL vehicle route optimization model by communicating through a dynamic reward function that is connected with the behavior and root-cause analysis models. This process is a hybrid model because the agent determines the solution via an RL trial-and-error strategy. Then, they evaluate the actions that interact with the environment using the behavior analysis models described in Chapters 3 – 4. Finally, the results of the behavior analysis are transmitted to the reward processing unit. Therefore, the information is used to train the agent on the chosen appropriate action and which action to avoid when performing the vehicle route optimization.

The agent's goal is to maintain the cumulative reward. In this study, the appropriate action means an agent is chosen to make delivery without violating any constraints. Therefore, a positive reward is given to the agent. Otherwise, a negative reward is given to reduce its reward. The process remains in this step until all customers are assigned to the vehicles. Finally, the optimal solution with the highest reward is returned. This chapter also presents the improvement of the result when behavior root-cause analysis is used together with the vehicle route optimization tasks and when optimization tasks are performed individually.

The author chose RL to perform vehicle route optimization instead of other algorithms because the author aims to develop the optimization model driven by the multisource data. Furthermore, it was shown that the model driven by the data could significantly reduce the complexity of defined constraints and also enhance the adaptability of the model according to the environmental changes. Thus, the method that fulfills this objective is to use the RL approach.

After the study introduction was presented, it is time to formulate the research problem definition.

5.2 Problem Definition

Recently, there were numerous vehicle route optimization solutions that might not be practical to deploy into the transportation planning operation. It causes by the disturbance and the lack of understanding of the current situation, whereas the optimized solution should be adapted. Those conventional models were not fully capable of compensating for such issues. In this chapter, a split delivery VRP (SDVRP) is solved by using RL. The RL approach is more flexible and accessible to adapt according to the environmental changes than the mixed-integer linear programming approaches.

Typically, the SDVRP is formulated as a graph $G = (\mathcal{V}, E)$ with a set of vertex $\mathcal{V} = \{0, 1, \ldots, n\}$, where 0 represents a depot, the other vertex represents a customer location. This location requires each vehicle to visit, and E represents a set of edges. Furthermore, each vehicle should start and end at the depot. In addition, each vehicle can visit a specific customer more than once until the demand is satisfied. The indices, parameters, and variables for this problem formulation are defined in Tables (5.1–5.3).

The purpose of this problem is to minimize the traversal cost obtained while making multiple deliveries to each customer until all demands are satisfied. The objective function is defined as Equation (5.1):

$$\min \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{v=1}^{m} c_{ij} x_{ij}^{v} \tag{5.1}$$

Table 5.1: Set of indices

Notation	Description
i, j	The origin i and destination $j \in \mathcal{V}$
n, p	n is the number of nodes in $\mathcal{V}$, $p = 0, \ldots, n$
K	The number of goods units that need to be delivered
k	Goods unit $k \in \{1, \ldots, K\}$
m	The number of vehicles that need to be used
v	Vehicle $v \in \{1, \ldots, m\}$
$\mathcal{V}^+$	Vertices in the set of vertex $\mathcal{V}$ except the depot, i.e., $\mathcal{V}^+ = \mathcal{V} \setminus \{0\}$
V	The subset of vertices in $\mathcal{V}^+$, i.e., $V \subseteq \mathcal{V}^+$
T	Limit of working hours per trip by limiting consecutive hours

Table 5.2: Set of parameters

Notation	Description
c_{ij}	The traversal cost of route i to j
t_{ij}	The travel time of route i to j
ct_i	The current time at vertex $i \in \mathcal{V}$
y_{iv}	The demand of i delivered by vehicle 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
Q_v	The vehicle v's capacity

Table 5.3: Decision variables

Notation	Description
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 $= \begin{cases} 1, & \text{if } v \text{ delivers goods unit } k \text{ to customer } i \\ 0, & \text{otherwise} \end{cases}$

subject to:

$$\sum_{i=0}^{n} \sum_{v=1}^{m} x_{ij}^v \geq 1; \; j = 0, \ldots, n \tag{5.2}$$

$$\sum_{i=0}^{n} x_{ip}^v - \sum_{j=0}^{n} x_{pj}^v = 0; \; v = 1, \ldots, m; \; p \in \mathcal{V} \tag{5.3}$$

$$\sum_{i \in V} \sum_{j \in V} x_{ij}^v \leq |V| - 1; \; v = 1, \ldots, m; \; V \subseteq \mathcal{V}^+ \tag{5.4}$$

$$\sum_{i=0}^{n} x_{ip}^{v} = \sum_{j=1}^{n} x_{pj}^{v}; \ v = 1, \dots, m; \ p \in \mathcal{V} \tag{5.5}$$

$$y_{iv} = h_{i_1}^{v} dm_i^1 +, \dots, + h_{i_k}^{v} dm_i^k; \ i = 1, \dots, n; \ v = 1, \dots, m \tag{5.6}$$

$$\sum_{v=1}^{m} y_{iv} = dm_i; \ i = 1, \dots, n \tag{5.7}$$

$$\sum_{i=1}^{n} y_{iv} \leq Q_v; \ v = 1, \dots, m \tag{5.8}$$

$$dt_v = \sum_{i=0}^{n} se_i + t_{ij}, \forall j \in V; \ v = 1, \dots, m \tag{5.9}$$

$$dt_v \leq T; \ v = 1, \dots, m \tag{5.10}$$

$$ct_i + t_{ij} \leq \hat{l_{ij}}, \ ct_i + t_{ij} \leq dl_{ij}^{v}; \ i = 0, \dots, |\mathcal{V}|; \ j = 1, \dots, |\mathcal{V}^{+}|; \ v = 1, \dots, m \tag{5.11}$$

Constraints (5.2) to (5.4) are routing constraints. Constraint (5.2) requires each customer location denoted as i to be visited at least once by vehicle v until the customer demand is satisfied. Constraint (5.3) is a comparison of routing similarity, where the result is equal to 0. The routes are then considered as the same route. This constraint aims to prevent simultaneously recommending identical routes with inverted directions for vehicle v, whereas Constraint (5.4) eliminates sub-tours of other recommended routes by comparing the current route sequence. It should not match the previously recommended route except the depot location.

However, Constraints (5.5) to (5.8) are customer demand constraints. Constraint (5.5) indicates that a customer demand can be delivered to customer i only if vehicle v has passed through route i. Moreover, the delivery by vehicle v is less than or equal to the demand at the customer's location i. This constraint aims to avoid situations where the vehicle is far away from the customer location, but is selected for recommendation. Constraint (5.6) allows the demand of customer i can be split, but each order is not detachable. Constraint (5.7) ensures that all demands for each customer i are satisfied with the delivery of cargo k from vehicle v. The summation of delivered items from all vehicles v that go to customer i should be equal to the customer's demand i. This constraint aims to avoid situations where the customer demand is not satisfied with the solution provided by the model. Constraint (5.8) indicates that the delivery taken by each vehicle v does not exceed vehicle capacity (Q_v), i.e., the delivery should be less or equal to the vehicle capacity. This constraint tries to avoid situations where a vehicle that cannot match the delivery size is chosen. In this case, the delivery should be split and shared among vehicles. Notably, the logistics fleet vehicles considered in this study are similar, with no differences in vehicle capacity. For this study, the capacity is set to 2,700 cubic feet.

Constraints (5.9) to (5.11) are the time constraints. Constraint (5.9) refers to dt_v, and is the accumulated travel time when vehicle v arrives at customer j. Constraint (5.10) ensures that no incidents have occurred and drivers are not exceeding the limit of working hours per trip by limiting consecutive hours of vehicle v to eight hours, preventing it from being unnecessarily overused. Constraint (5.11) is the time window limit for each customer j, requiring the vehicle to arrive before a given deadline (dl^v_{ij}). Therefore, the accumulative travel time should be less than or equal to the time window limit; otherwise, the trip is not recommended.

Up to this point, the general SDVRP problem formulation already took up to 10 constraints. Suppose one wants to include more elements from the real-world problem, the model will get more complicated. Therefore, we omitted this part and included it into the RL problem formulation instead, which much easier to interpret.

To perform the model construction in RL, the combinatorial optimization problem formulated in Equations (5.1)–(5.11) is transformed via the RL problem formulation [72]. The RL problem formulation consists of tuple $< S, I, A, T, G, R, C >$ and their definitions are defined as follows:

- **States (S)**: S is a finite set of the state (s) of the environment.The elements in the environment state are defined as $s = [latitude, longitude, demand, vehicle_{avail.}, driver_{avail.}, vehicle_{maintenance}]$. For simplicity, states are the current environment that the RL interacts with and provides a set of partial solutions to the problem (e.g., a partially constructed route for the VRP problem).

- **Initial state (I)**: Current vehicle location with current fleet availability status specified by the company. At this stage, the element consists of 6 dimensions: $latitude$, $longitude$, $demand$, $vehicle_{avail.}$, $driver_{avail.}$, $vehicle_{maintenance}$.

- **Actions (A)**: A is the possible actions performed by 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.

- **Transition model (T)**: T is the state transition probability from $S \times A \times S \to [0, 1]$. This statement is defined in the "Formulation of Reinforcement Neural Vehicle Route Optimization" section in Equation (5.29). Therefore, in each state $s \in S$, the best possible action to chose computed from optimal policy $\pi^*(s) \in A$.

- **Goal test (G)**: Do all deliveries satisfy customer requirements, and is the traversal cost of each trip minimized as defined in Equation (5.1)? Do the actions also satisfy the evaluation from Equations (5.2) to (5.11)?

- **Reward (R)**: For each state transition, the author defines the reward function as $S \times A \times S \to G(Z)$. The $G(Z)$ defined as Equation (5.16) in the "Reward Processing for the RL Agent" section.

- **Path cost (C)**: C is a traversal cost from $S \times A \times S$ defined in Equation (5.12).

Note that the RL problem formulation did not represent the mechanism for the SDVRP at the moment. Therefore, the mentioned mechanism will be carried on to present in the "The Proposed Model" section. In addition, the dynamic elements of the real-world problems, which are taken into consideration for the RL formulation of the SDVRP, consist of customer demand, road-network traffic, operation status (e.g., fleet availability, resources' utility, and productivity). These elements are also included incidents (e.g., order cancellation or postponement).

The behavior analysis models in Chapters 3 – 4 were used to analyze these elements for any abnormalities. This event is done when the RL agent's behavior interacts with the environment. The information obtained is then used to train the RL agent in the next episode. Therefore, the model trained the RL agent to perform the appropriate actions and dynamically adapted to the current transportation environment while performing daily vehicle route optimization tasks. At the end, when the terminated condition is satisfied, the optimal result is returned. For the path cost, the author calculates the traversal cost and profit per day as in Equations (5.12) and (5.13), respectively.

$$\begin{aligned} traversal_{cost}^{dow} = \sum_{vr=1}^{N} &((d^{vr} \times mt) + ins^{vr} \\ &+ (d^{vr} \times de) + sc^{vr} + (d^{vr} \times f) + \left(\left(\frac{td^{vr} \times 20}{1,000} \right) \times f_{price} \right) \\ &+ (tr^{vr} \times l) + (tr^{vr} \times al) + (ovt^{vr} \times wt)) \end{aligned} \tag{5.12}$$

where dow denotes a calendar day, vr denotes a vehicle used in the current route, N denotes the total number of daily use vehicles, d denotes the distance (Km.), wt denotes the waiting time needed to complete the delivery (Hrs.), tr denotes the traveling time (Hrs.), and td denotes traffic congestion time of the road link (Mins.). Note that this equation has the same role as parameter c_{ij} in Equation (5.1).

For this chapter, the author uses Thai Baht (THB) as a base currency. However, when applied this model formulation to optimizing vehicle routes in other regions, the currency must only be adjusted. The remaining parameters remain the same. These parameters are general and used to compute costs of the vehicle by various agencies, as described below.

Further, mt denotes maintenance cost (THB/Km.); ins denotes insurance cost (THB); de denotes the depreciation cost of the vehicle (THB/Km.); sc denotes the service cost of handling goods from the containers (THB); f denotes fuel cost (THB/Km.); $fuel_{price}$ denotes fuel price (THB/Liter); l denotes labor cost (THB/Hrs.); al denotes the additional labor cost for driver assistance (THB/Hrs.); and ovt denotes overtime payment (THB/Hrs.), in cases where the vehicles are required to wait at the customer site (e.g., waiting for more than regular working hours). If the time is less than 60 Mins, it is count as 1 Hr.

Equation (5.12) calculates the total traversal cost of the routing in each day. It is denoted as dow. This cost is the sum of the traversal cost incurred from vehicles. These vehicles are used for delivery on the day dow. Therefore, the cost of each vehicle vr is calculated and summed as the total traversal cost.

$$profit^{dow} = incomes^{dow} - traversal_{cost}^{dow} \tag{5.13}$$

where $incomes^{dow}$ denotes a payment retrieved from the customer, the customer paid the delivery fee to the logistics agency and dow denotes a calendar day.

For instance, in the case-study conducted, these values given in Equation (5.12) were obtained from the company's actual operational report. They are described in Table 5.4 below.

After the problem definition was already formulated, it is now ready to present the proposed model in the following section.

Table 5.4: Notation and values that used to calculate the profit and traversal cost.

Notation	Value	Description
d	From the routing in Km.	Distance of travel
wt	From the routing in Hrs.	Waiting time required until the delivery is completed
tr	From the routing in Hrs.	Travel time aggregated from GPS data
td	From the routing in mins	Traffic congestion time aggregated from GPS data
mt	1.35 THB/Km.	Maintenance cost
ins	105 THB	Insurance cost
de	2.8 THB/Km.	Depreciation of vehicle cost
sc	100 THB	Service cost of handling the containers
f	7.29 THB/Km.	Fuel cost of each trip
f_{price}	19.89 THB/Litre	Fuel cost per litre
l	300 THB/Day or 12.5 THB/Hrs.	Labor cost of the driver
al	200 THB/Day or 8.33 THB/Hrs	Additional labor cost for driver assistance
ovt	1,000 THB/Day or 41 THB/Hrs.	Overtime cost

5.3 The Proposed Model

In this chapter, the proposed model's implementation consists of four crucial steps. It consists of travel time estimation for vehicle routing, reward processing unit for the selected agent, and receiving the agent's action value, reinforcement neural model formulation for vehicle route optimization. Additionally, the behavior analysis model described in Chapters 3 – 4 are applied to the proposed methodological framework. The overall procedures for formulating the model are illustrated in Figure 5.1.

At this point, the author gave a brief introduction to the proposed model. Therefore, in the following sections, the processes shown in Figure 5.1 are presented in detail.

5.3.1 Travel Time Estimation Model for Vehicle Routing

In this section, the author computes the road-network traveling time using the framework's traveling time estimation model. The model architecture is the deep neural network model that has seven input neurons, five hidden layers, and one of the output layer. The model was constructed with a batch size of 64, a learning rate of 10^{-3}, and 1,500 epochs. It also pre-trained using the GPS probe data. This model architecture was obtained from the hyperparameter tuning process. To estimate the traveling time on the road network, the author collected data over 1 year from multiple trucks registered at a logistic agency fleet. The traveling time estimation model was developed because, generally, the route of a delivery truck is not similar to that of a regular vehicle. Suppose that the travel times presented by Google Maps were used, then they would reflect the passenger-vehicle travel times rather than the truck travel times. Doing so would cause significant differences between the traveling speed and duration gaps.

For this study, the origin and destination were input into the model, and the outcome is the travel time of the vehicle route from those origins and destinations. The travel times were computed from the speed on a specific road network on the current day of the week at the current time. The agent used this road network information to support their selected customer to make deliveries in the route optimization module. This information is communicated to the agent via the reward processing unit.

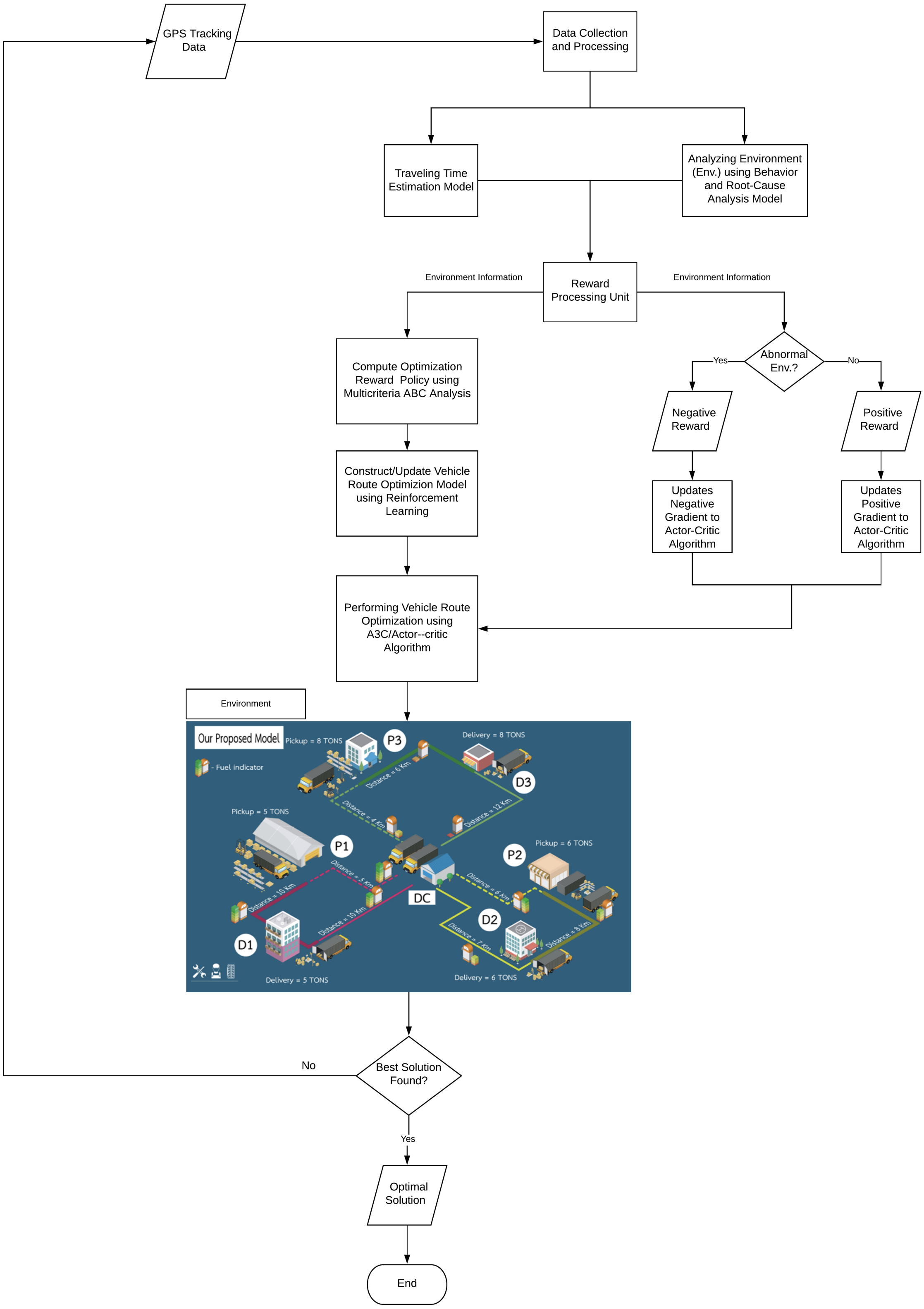


Figure 5.1: Methodology to perform the behavior analysis and new reinforcement neural model for vehicle route optimization.

5.3.2 Reward Processing Unit for the RL Agent

This section presents the reward processing unit. This unit helps in calculating rewards for the evaluated agent when optimizing the vehicle routes. For example, the positive reward is given to the agent when it returns the optimal solution. Otherwise, a negative reward is given when the solution has violated any problem constraints. It also communicated with the agent by updating the actor–critic gradient with these values. This process is presented in Equations (5.30)—(5.31).

To construct the reward processing unit, the information collected from the multi-sensory, operation reports, and model of Chapters 3 – 4 are taken as inputs for this unit. This information is then used to compute RL agent reward using the new reward function, as expressed in Equation (5.16).

The author designs the reward function to consider the operational feasibility and fairness for the goods deliveries and road-link factors of task assignment. The stated factors are given as inputs to the multiplicative utility function. It is used to express the behavior of an agent when it interacts with an uncertain environment. The corresponding action's outcomes involve multiple data attributes. These attributes represent as $X = \langle x_1, \ldots, x_n \rangle$, where each x_i represents a data attribute for each involving factor from the environment. For instance, traffic condition, customer demand, fleet availability, incidents, and more.

The uncertainty environment problem can be solved by the multi-attribute utility theory [72]. The author assumes that the agent returns better results when it has a higher utility value. Therefore, the author evaluates the properties of each utility function's result. This evaluation is to ensure that the RL agent understands the action selection's outcomes. Thus, the expected preferences were determined from each property in the utility functions. The objective is to maximize the value of the utility (u) from the agent's behavior. This objective is described using Equation (5.14):

$$u = argmax\,(F\,[\,f_1\,(\,x_1\,),\,\ldots,\,f_n\,(\,x_n\,)\,]\,) \tag{5.14}$$

where F calculates the utility function of each data attribute, as mention earlier. The multiplicative utility function is a part of the reward function. Normally, the multiplicative utility function (MU), is formulated as Equation (5.15):

$$MU = k_1 u_1 \; + k_2 u_2 +, \ldots, + \, k_n u_n \tag{5.15}$$

where k_i is the weight of the utility (u_i), it is the weight of each data attribute. Hence, the higher utility values indicate that the agent is doing in the appropriate direction.

The proposed reward function in this section was adapted from [73]. It also includes the theory that was adapted from [72], as shown earlier in Equation (5.15). These changes to the function were to be applied to the VRP. The reward function was also modified to consider the logistics management strategies. Therefore, each function focuses on solving a particular aspect using RL.

The new reward function $G(Z)$, is formulated in Equation (5.16). This function consists of four main components. For instance: the traversal cost of the vehicle routing ($traversal_{cost}$), the function that evaluates the number of vehicles used ($U(Z)$), the number of delayed and undelayed deliveries ($T(Z)$), and the fairness ($I(Z)$). These last three components are multiplicative utility functions. They are transformed into a mathematical equation, represents as Equation (5.15).

$$G(Z) = traversal_{cost} \; + \; \beta \, U(Z) \; + \; \delta \, T(Z) \; + \; \gamma \, I(Z) \tag{5.16}$$

where Z is the transportation logistics system under evaluation. $U(Z)$ and $T(Z)$ indicate the usage and delivery performance, respectively. $I(Z)$ indicates the daily tasks distribution fairness. This value is due to the actions taken. Moreover, β, δ, and γ are the weights to match

100%. They are used to indicate how each utility evaluation function is important. Thus, these utility evaluation functions are expressed in Equations (5.17)–(5.23).

Operation evaluation function: to compute efficiency $p(x)$ in the work assignment for each period x, as represented in Equation (5.17):

$$p(x) = \Theta\,(\,c - W(x)\,)^{(\,c - W(x)\,)} \tag{5.17}$$

where c is the maximum vehicle's capacity to manage the deliveries and $W(x)$ is the number of tasks assigned to the vehicle (v) in period x. x is a period under analysis. Θ is a function that returns zero, in the case that the input is less than zero. It denotes that the vehicle capacity was exceeded.

The capacity was proportionally exponential to the capability of handling goods deliveries. Therefore, the larger $p(x)$ value denotes that the vehicle (v) can handle more goods deliveries. Thus, the total usage $(U(Z))$, as indicated in Equation (5.18):

$$U(Z) = \sum_{x \in Z} p(x) \tag{5.18}$$

Delay evaluation function: to evaluate the delay in operation processing $(d_{dt}(v))$, as indicated in Equation (5.19):

$$d_{dt}(v) = \Theta\,(1\, +\, (\,t - \propto(\,t_{ad},\, t_{aed}\,)\,)\,) \tag{5.19}$$

where t denotes the current time, t_{ad} denotes the actual departure time, and t_{aed} denotes the estimated departure time. Furthermore, $\propto$ is the function used to calculate the road network travel times by inputting the departure and arrival times together with the origin and destination to the traffic estimation model. Θ is a function that returns zero, in the case that the input is less than zero. It denotes that the deliveries were delayed. The summation of the delivery performance $(T(Z))$ with delay and no delay is indicated by Equation (5.20):

$$T(Z) = \sum_{x \in Z} \sum_{v \in x} d_{dt}(v) \tag{5.20}$$

Operation feasibility evaluation function: to evaluate the operation feasibility and fairness $I(Z)$, as indicated in Equation (5.21):

$$I(Z) = \sum_{x \in Z} (\,d_{ad}(x)\, +\, O_{at}(x)\,) \tag{5.21}$$

where $d_{ad}(x)$ is the impact on the distribution of delivery delay or cancellation, and $O_{at}(x)$ is the operation loss due to the delivery delay or cancellation in all vehicles.

Task distribution evaluation function: to evaluate the fairness $(d_{ad}(x))$ as indicated in Equation (5.22):

$$d_{ad}(x) = \sum_{v \in x} (\frac{d_{at}(v)}{size(delivery)}) \times 100 \tag{5.22}$$

where $d_{at}(v)$ is the number of vehicles used to deliver the customer demand, and $size(delivery)$ represents the number of deliveries that flow into the transportation logistics system. This function is affected by the imbalance of deliveries' distribution among vehicles.

Feasibility evaluation function: to determine the operation feasibility, the author discovered that the unsuccessful deliveries were a proportionally exponential impact on the logistics agency income. Therefore, the $O_{at}(x)$ function is quantified based on the delay time and the number of incidents, as indicated in Equation (5.23):

$$O_{at}(x) = \sum_{v \in x} \Theta\,(\,O(v)\; -\; \hat{d_{at}}(v)\,)^{(\,\Theta\,(\,O(v) - \hat{d_{at}}(v)\,)\,)} \tag{5.23}$$

where $O(v)$ indicates the expected delivery cargo that a vehicle should be load, and $\hat{d_{at}}(v)$ indicates the total delay or delivery cancellations in the vehicle (v). Θ returns 0 in the case that the delivery of vehicle (v) has been delayed and not successfully delivered. Therefore, the larger $O_{at}(x)$ value denotes that the logistics agency can earn more income from goods deliveries. Note that x for all equations presented earlier indicates the period for delivering goods to customers.

After the reward for an agent is obtained, as defined in Equation (5.16), the reward is given according to the agent's behavior when optimizing the vehicle routing. Suppose that the behavior analysis model detects any abnormalities in the behavior of the agent. This condition also includes unusual changes in the fleet capacity from the LSTM model's forecast result; then, the negative reward is returned. Otherwise, the positive reward is returned.

The reward mentioned earlier is sent to update the actor–critic gradient, as defined by Equations (5.30)–(5.31). Lastly, to make the programmable algorithm, all mathematics equations presented in this section are transformed into the reward-function algorithm, presented in Algorithm 5.1.

Algorithm 5.1: New reward function to evaluate the RL agent.

```
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
    $d^n \leftarrow calculate_{dist}\,(\,sample_{solution},\, sample_{solution_{titled}}\,)$
    $travel^n \leftarrow d^n / model_{traff}\,(\,F_1,\, F_2,\, \ldots\, F_n\,)$
    $behavior_{stat} \leftarrow model_{behav}\,(\,Vehicle_{used},\, F_1,\, F_2,\, \ldots\, F_n\,)$
    $evaluate_{feasibility}\,(\beta,\, \delta,\, \gamma\,) \leftarrow \sum_{i=0}^{n}(\,\beta \times evaluate_{util}\,(work)\,)\, +$
        $\hookrightarrow$ $(\,\delta\, \times evaluate_{util}\,(delay)\,)\, +$
        $\hookrightarrow$ $(\,\gamma \times (\,evaluate_{util}\,(operation,\, fairness)\,)\,)$
    $\beta,\, \delta,\, \gamma \leftarrow model_{mdcm}\,(\,Vehicle_{used},\, F_1,\, F_2,\, \ldots\, F_n\,)$
    if $behavior_{stat}$ is true then
        reward $= -(d^n + waiting_{time} + evaluate_{feasibility}\,(\,\beta,\, \delta,\, \gamma\,)\,)$
        return reward
    else
        reward $= d^n + evaluate_{feasibility}\,(\,\beta,\, \delta,\, \gamma\,)$
        return reward
    end if
end for
```

$sample_{solution}$ denotes the route selected by the agent, $sample_{solution_{titled}}$ denotes the inverse order of $sample_{solution}$, d^n denotes the traveling distance, $model_{traff}$ denotes the travel time calculation in Chapter 5.3.1, $model_{behav}$ denotes the behavior analysis model described in Chapters 3 – 4.

$evaluate_{feasibility}$ is the function that transforms from Equation (5.16), and $model_{mdcm}$ is the multi-criteria analysis model used to select an appropriate logistics management strategy, as discussed in the next section.

5.3.3 Criteria and Method of Selecting Appropriate Logistics Strategy for Vehicle Route Optimization

In this section, the mechanism to activate the utility functions of the reward processing unit is presented. The utility functions are activated concordant to the current environment and based on the given logistics management strategy. For example, suppose that the optimization task is focused on the efficiency of the delivery plan. The $U(Z)$ and $T(Z)$ utility functions in Equation (5.16) should be activated with β, δ, and γ set to [1,1,0]. To make it adaptive, the author used the multi-criteria ABC analysis method to perform this task.

In the study of decision making, multi-criteria ABC analysis is a well-known method of selecting strategies. It is also used to recommend a cost-effective solution during logistics management. The multi-criteria ABC analysis methods are based on the Pareto principle [74]. This principle is initiated by Vilfredo Pareto.

The principle of multi-criteria ABC analysis was developed based on the assumption that the majority of economic productivities (80%) are attributable to only a small part (20%) of the economy. The outcome shows that the relationship between input and output is unbalanced. In this chapter, the principle is applied to the ABC class as follows.

- Class A: The solution selected by the agent represents approximately 15–20% of the set of solutions presented. However, these solutions represent 80% of the value of the company cost. The solutions in this class represent the actions responsible for the highest value of the annual cost.

- Class B: The solution represents 30–35% of the set of solutions provided by the agent and approximately 15% of the profit of the company. The solutions in this class denote the actions representing the medium cost returned from the optimization process.

- Class C: The solution represents 50% of the set of solutions provided by the agent, but only 5% of the cost returned to the company. The solutions in this class denote the vehicle route optimization solutions with the lowest cost.

To select an appropriate reward function and logistics management strategy, the author used multi-criteria ABC analysis written in the python programming language to perform this task. The essential steps included in [75] were modified. Thus, the procedure for performing multi-criteria ABC analysis is comprised of six steps as follows:

1. Calculate the operational cost for transportation logistics by evaluating the summation of demands, multiplied by the transportation cost per unit.

2. Concatenate the obtained operation cost with fleet utilization and productivity ratio per date reflected from the logistics operation.

3. Sort the transactions in descending order of the operational cost of transportation.

4. Calculate the cumulative operational cost and the percentage.

5. Divide the operational cost of each transaction into classes.

6. Perform the analysis and decision making.

In this section, the traversal cost and profit per day were calculated similar to equations in section 5.2.

To select appropriate logistics strategies from the profit analysis of various strategies, the methodology of [76, 77] were applied. The author used ML models to classify each situation of the environment into classes. This classification is considering the dynamic information from the environment. ML was performed because the model is dynamic optimization. Therefore, it is not practical to manually evaluate a class for each task. Consequently, it also increases the computation time. However, an accurate classification has also reflected the efficiency of the multi-criteria model when selecting the evaluation factors.

Furthermore, the author used five-fold cross-validation to evaluate the model performance. Accordingly, the following classification criteria were considered: the value of the profit per date after deducing the traversal cost and fleet utilization with the productivity ratio per date, as reflected from the logistics operation. This ratio was discussed earlier in Chapter 4. A summary of multi-criteria ABC analysis and ML strategies is presented in Algorithm 5.1 and illustrated in Figure 5.2.

While reward function was computed using Equation (5.16), the multi-criteria ABC analysis set the learning parameters, β, δ, and γ. These parameters were set for selecting the strategies of rewards that should be given to the agent according to the current fleet situation. In the next section, the author will present mechanisms for performing vehicle route optimization using RL.

5.3.4 Formulation of RL Neural Vehicle Route Optimization

After the reward function was defined in the prior section, RL's vehicle route optimization model is now ready to be formulated. Note that this section represents as "Construct/Update Vehicle Route Optimization Model using the Reinforcement Learning" component of Figure 5.1.

In this section, RL is used to formulate the vehicle optimization problem. The reason behind this decision is that the author aims to reduce the model's complexity while maintaining a high-efficiency solution. Every time the new customer requirement and route information are taken as inputs, the conventional models must be reconstructed for adapting to the new environment changes. These inputs are presented in the form of constraints. If there are numerous constraints, then it is not feasible to alter the model's structure. This problem motivated us to replace the mixed-integer linear programming part with ML and RL. The model utilized the data to adapt rather than input fix constraints.

In the model, the set of inputs are defined as $X = \{x^i\}$, $i = \{1, \ldots, M\}$. The inputs indicate the customer information (e.g., coordinates and demand) flow into the transportation logistics system.

The model updates elements in X during the decoding stage, where goods (d) are delivered to the customer at time t and location in state s. Each input of X is represented as x^i and denoted by a sequence of nodes expressed as $x^t \doteq (\, s^i, d^i \,)$, where $t = 0, 1, \ldots, n$. The author follows the same concept as that of the model presented in [78].

To make the model support the split-delivery VRP, the author uses the masking scheme introduced by [78] to label the nodes as follows:

1. Customer nodes with no goods delivery requests will not be visited.

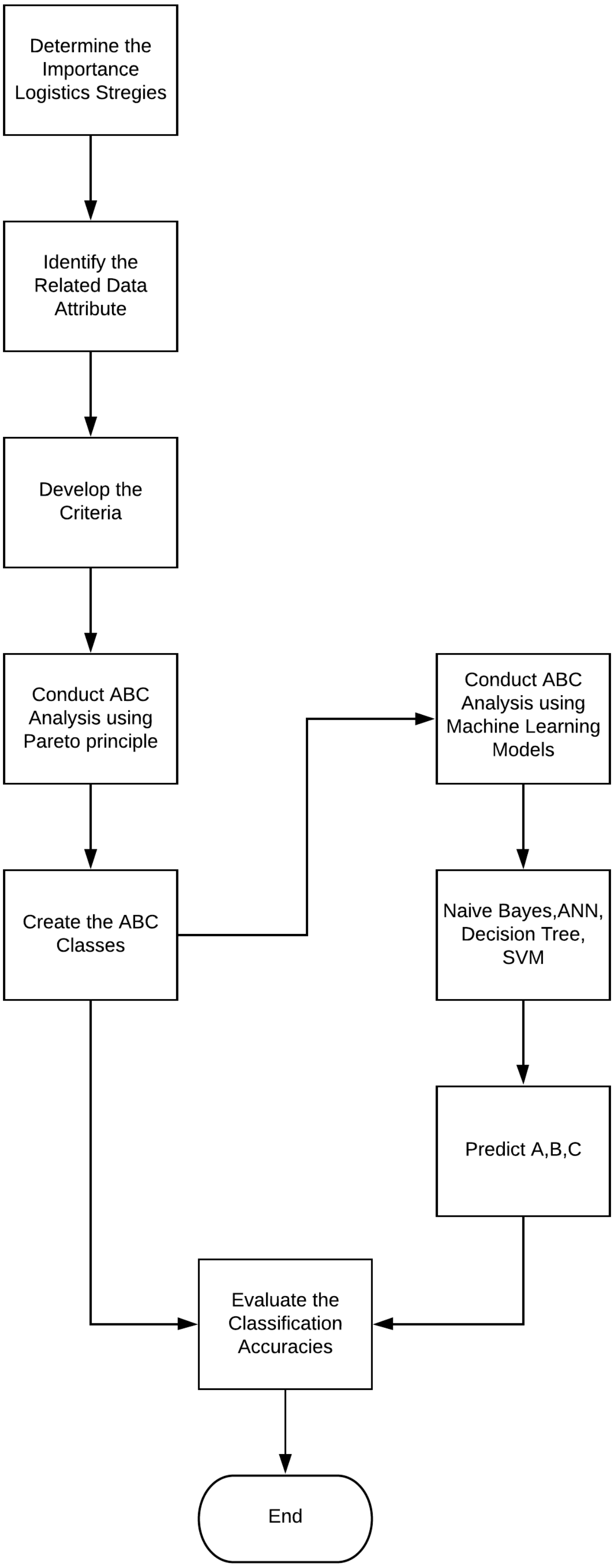

Determine the Importance Logistics Stregies
Identify the Related Data Attribute
Develop the Criteria
Conduct ABC Analysis using Pareto principle
Create the ABC Classes
Conduct ABC Analysis using Machine Learning Models
Naive Bayes,ANN, Decision Tree, SVM
Predict A,B,C
Evaluate the Classification Accuracies
End

Figure 5.2: Methodology to perform the logistics management strategies selection for vehicle route optimization.

2. Customer nodes with goods delivery requests higher than the vehicle capacity are masked.
3. All customer nodes will be masked if the remaining vehicle capacity is 0.

These conditions are related to classical VRP constraints. Suppose that the problem is formulated as a split-delivery VRP, then condition (2) should be relaxed to allow a vehicle to visit a customer more than once. Therefore, all customer delivery requests are satisfied when the traveling plan is set under these masking scheme conditions.

In [78]'s study, they set the log-probabilities of infeasible solutions to $-\infty$. In other words, a solution is forced in the case that a particular condition is satisfied. However, for this study, the author uses the reward function and Nazari et al. defined conditions together in the masking scheme. The author aims to make the masking scheme label the customer's node according to the logistics management strategies and the general VRP conditions.

After a brief introduction about the mechanism to perform the vehicle route optimization, the author starts from the input, X_0, where a pointer is used. The pointer is denoted by y_0, to refer to the input. At each decoding step at time t, where $t = 0, 1, \ldots, n$. y_{t+1} are the pointers to the remaining inputs, X_t. The topmost adjacent node is then selected as the next decoder step. The process evaluates the current obtained solution with the goal test condition and continues until a termination condition is satisfied (e.g., reach the maximum process iteration or a feasible solution is obtained with optimal iteration).

In the RL model, the policy (π) is determined by the policy-gradient method. The policy-gradient is calculated and used by the actor and critic network. The author uses the actor-network to predict the probability of the following action to be chosen at a given decision step. On the other hand, the critic network is used to calculate the value of the action performed in the current state. It is generated from the sequences of Y to minimize the loss of the objective function.

The optimal policy (π^*) generates the optimal value with a probability of 1. Therefore, to determine the optimal solution, π and π^* must be as nearly identical as possible. Similar to the process in [78], the author uses the probability chain rule to determine the state transition probability of generating sequence's (Y) from the next given sequence (y_{t+1}), and decoded node (X_t) based on Equation (5.24):

$$\Pr(Y \mid X_0) = \prod_{t=0}^{T} \Pr(y_{t+1}, X_t) \tag{5.24}$$

The state is repeatedly updated with the state transition function (f), as indicated in Equations (5.25)–(5.26). While performing this step, the current node's sequence (e.g., $location_1$, $location_3$, $location_2$, and vice versa) was send to the reward function unit to evaluated in terms of its utility, productivity, and feasibility.

$$X_{t+1} = f(y_{t+1}, X_t) \tag{5.25}$$

$$\Pr(Y_{t+1} \mid Y_t, X_t) = softmax(g(h_t, X_t)) \tag{5.26}$$

where g is the function that calculates the distance between the input vectors and h_t is the state of the RNN. The outcome indicates the probability that the previously given decoded step will transition into the next decoder node. The probability is obtained from the *softmax* function. Thus, this part denotes as transition model of the RL problem.

After the model performs these steps and the termination condition is reached, the output is the sequence of routing (e.g., $location_1$, $location_3$, $location_2$, and vice versa). It is a tour that each vehicle in the fleet is assigned to make goods delivery to customers. Furthermore, the model returns the minimum traversal cost and feasible for recommendations into the operation. Therefore, the objective function is satisfied, as presented in Equation (5.1).

In conclusion, at this step, the minimum traversal cost is obtained concerning the vehicle route's distance, time, and operational attributes (e.g., fleet capability, efficiency, and fairness). This output is also recommended and chosen by the RL agent in solving the VRP. Additionally, when new information is input into the model, the model is dynamically updated follows the new information.

In this section, the author demonstrated the mechanism to create the optimal solution. Generally, VRP was modeled as a graph consists of nodes and edges. However, using the machine learning approach to do in a similar way required further modifications. Therefore, in the next section, the author will present how the RNN is modeled as a graphical network.

Proposed Network for Managing a Sequence of Inputs

Generally, the neural network neurons do not have any references showing which neuron is adjacent and connects to them. Therefore, using the neural network to model the vehicle route required a mechanism that connects each network's node together as a graph. This section presented how a graphical network is constructed and how each network node is connected.

To begin, the input is embedded as described previously instead of using the RNN hidden states because the author wants the network node to represent the customer location and the demand information. This approach differs from those of previous works because the inputs are non-arranged customer locations, and the demands are dynamically updated. Unfortunately, when the RNN hidden states technique is used, each location's index is not essential when a random change is applied to the locations. Thus, it has the same form as the original input sequences. Also, it is not easy to update the customer demand to the hidden states directly.

To solve the issue mentioned above, the proposed model consists of two components. The first is a set of embedding processes that connect the input's attributes and project them into a D-dimensional vector space.

On the other hand, the second model component is a decoder layer that points to an input. It is used to determine the next visited node at every decoding step. This procedure is also similar to the process presented in [78]'s study. The author uses an RNN to constructing the decoder network. The decoder network is used to determine the next visiting node from the sequence of remaining nodes using Equation (5.26). After identifying the next node to be visited by each vehicle, it is necessary to update the delivery information. It is because the number of deliveries changes over time while goods are continuously delivered to the customers. The question raised at this step is, "where shall we stored the demand and coordinate information?" It is because it is not possible to store these pieces of information into the network node directly. To answer this question, the author adopts the attention mechanism of [79, 78] to assist in updating and storing the delivery and customer coordinate information.

In this section, the author has presented how the RNN is modeled as a graphical network. In the next section, the attention mechanism is described in detail on how each node's demand and coordinate are updated when the vehicle is successfully delivered goods to the customer or when a new customer request is inputted into the system.

Attention Mechanism

In this section, the author presents the attention mechanism. This attention mechanism is used to store the demand information of each network node and connect independent network nodes together as a graph. Therefore, every time the vehicle comes to make goods delivery at each customer location (e.g., [latitude, longitude]), the remain demand information is updated on the "attention layer" instead of the network itself. Therefore, the model is not necessary to be rebuilt again by doing so. The process described earlier is shown in Figure 5.3.

Typically, an attention mechanism is a component used to reference inputs at the decoder step. It is denoted by i and used to calculate the probability that the adjacent node is the next node of the current network node. This mechanism is vital because it connected the independent nodes as a graph. The nodes are also updated when the goods delivery requests from customers change due to goods being successfully delivered. As mentioned earlier, the pointer network must be reconstructed, and the node sequences must be rearranged. However, Suppose that the attention mechanism is used, the network structure remains the same. The amount of remaining goods delivery requests is updated through the attention layers rather than moving the node location. Therefore, the reference pointer is used to point to the node.

To perform this process, the encoded input is denoted by $\bar{x}_t^i = (\,\bar{s}^i, \bar{d}_t^i\,)$. t and $h_t \in \Re^D$ indicate the RNN state memory at the decoding step t. For example, x_0 represents the beginning of the tour, and the attention mechanism is used to determine the remaining nodes. They are nodes that are suitable to be the next node of the current decoded node. The context value is computed to support this decision. The attention layer is denoted as $attent_t$. It is defined in Equation (5.27):

$$attent_t = attent_t\,(\,\bar{x}_t^i,\, h_t\,) = softmax\,(\,u_t\,) \tag{5.27}$$

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 denotes the set of nodes in sequence Y. Moreover, v_a and W_a are the training variables.

After obtaining the attention value, the author determines the node context ($context_t$). The context value is used to select the possible next-visited node. It is determined by Equation (5.28):

$$context_t = \sum_{i=1}^{M} attent_t^i \bar{x}_t^i \tag{5.28}$$

With the encoded input, the values are normalized using the $softmax$ function, where $Pr\,(\,y_{t+1}\,)$ denotes the next sequence in Y, given the current sequence Y_t and decoded node X_t, as indicated in Equation (5.29):

$$\Pr\,(\,y_{t+1}\,|\,Y_t,\, X_t\,) = softmax\,(\,\tilde{u}_t^i\,) \tag{5.29}$$

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

A single Long Short-Term Memory (LSTM) layer for a decoder is used to implement this RL model. Based on the hyperparameters tuning, the author discovered that an appropriate decoder and vector size that can store all vital information is 128. It is because the model accuracy is unchanged after increase size to 129 and beyond. In this study, the author stores customers' coordinates and demand in each node of the network. Therefore, suppose that more information is required, then the decoder's and vector's size must be adjusted.

Furthermore, the customer location is embedded in a vector size of 128. The demand d_t^i is updated at step time t as $delivered_i - d_t^i$. After that, it is mapped to the attention layer. To train the actor–critic networks, the Asynchronous Advantage Actor–Critic (A3C) algorithm is used for the SDVRP [78] and the Adam optimizer with a learning rate of 10^{-4}. The dropout

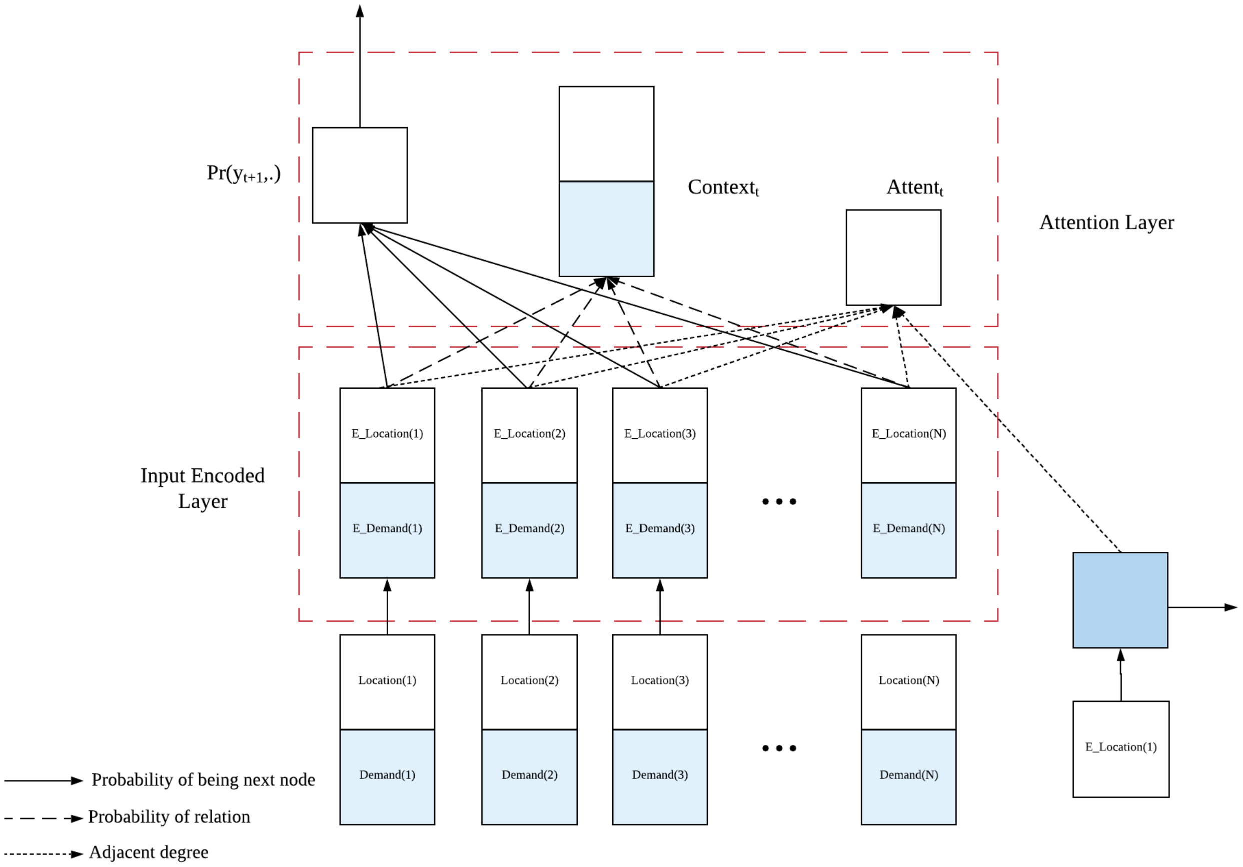


Figure 5.3: RL model architecture for vehicle route optimization.

with a probability of 0.1 is set at the LSTM decoder. After obtaining and setting up all necessary parameters, the author then trains the model on the NVIDIA Tesla P100 GPU. The model architecture is illustrated in Figure 5.3.

From this section, the author presented how the network is dynamically updated when the environment changes. In the next section, the training algorithms for training this network are described in detail.

5.3.5 Training Algorithm

In this chapter, two algorithms are presented for training the RL agent, i.e., the actor–critic algorithm and the A3C algorithm. The author trained the actor–critic and A3C algorithms with 120,000 and 100,000 steps for testing. The algorithms are trained with three years of historical data (from 2017 – 2019) and test with three-month route optimization data during May – July 2018. Note that during May – July 2018 data period was reserved for testing and was not included in the training stage. This period was reserved because the site survey is conducted at the logistics agency. Also, validation data were obtained from both drivers and staffers from the agency during this period. In the following sections, each training algorithm is described in detail.

Actor–Critic Algorithm

According to the model architecture presented in Figure 5.3, the author used the actor–critic algorithm to train the agent. The model integrates the benefits of the policy- and value-based approaches for evaluating the agent's action. For updates, the actor gradient is defined by Equation (5.30), and the critic gradient is defined by Equation (5.31).

$$d\theta \leftarrow \frac{1}{N} \sum_{n=1}^{N} (\, R^n - V\,(X_o^n;\, \phi\,)\,)\, \nabla_\theta \log \Pr\,(\, Y^n \,|\, X_o^n \,) \tag{5.30}$$

where $d\theta$ represents the actor gradient and R^n represents the reward from solution n. Reward R^n was obtained from the reward processing unit presented in Chapter 5.3.2. Figure 5.4 illustrates how the reward was used to update the agent behavior.

$$d\phi \leftarrow \frac{1}{N} \sum_{n=1}^{N} \nabla_\phi\, (\, R^n - V\,(X_o^n;\, \phi\,)\,)^2 \tag{5.31}$$

where $d\phi$ represents the critic gradient and R represents the reward from solution n. Reward R^n was obtained from the reward processing unit presented in Chapter 5.3.2.

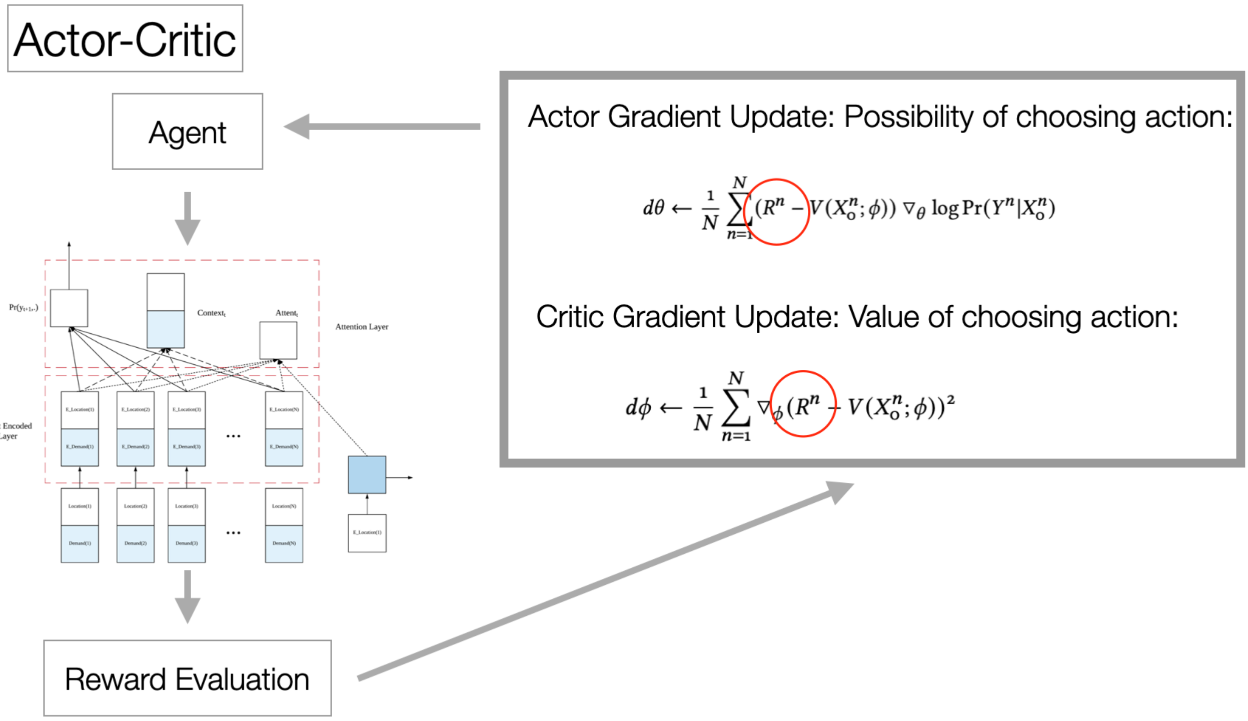


Figure 5.4: Actor–Critic algorithm architecture

A3C Algorithm

In this section, the author applies A3C to train multiple agents. Each agent is initialized and trained in parallel on a different copy of the environment. The policy and value estimations are similar to those of the general actor–critic algorithm. However, the policy is communicated to the "global agent" of the upper level. Thus, all experiences are collected from each agent.

Each agent also learns other agents' experience from the information provided by the global agent at each time step [80].

Moreover, A3C increases the opportunities for agents to obtain experiences shared by local agents. When agents obtain experiences in this manner, their action-selecting behavior is updated. This phenomenon increases the efficiency in comparison to the learning from only one agent in a broad environment. Figure 5.5 illustrates how the experience is shared among agents. The experience is also inherent to the other agent's behavior.

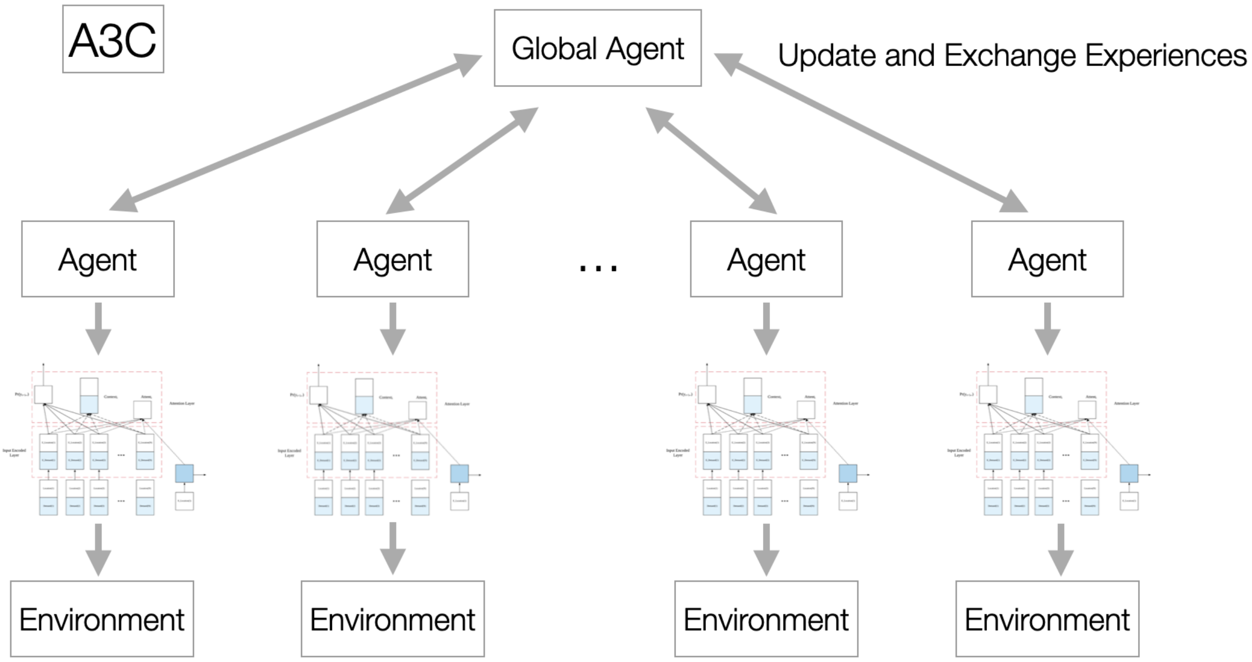


Figure 5.5: Asynchronous actor–critic algorithm architecture

In summary, the model is updated dynamically. The customer delivery request changes over time. Therefore, the attention mechanism is used to determine the node that should be decoded for the next visit. Unlike a traditional pointer network, it is unnecessary to rebuild the model for each change occurring during each decoding step. The model is transformed into a programming algorithm, as presented in Algorithm 5.2. In [78]'s study, the authors defined a constraint using the tour's negative rewards to guide the agent's actions. However, it is necessary to consider the external factors regarding the logistics agency's capability and environmental information in real applications. Therefore, using only defined constraints and a negative tour length for the agent reward might not feasible for addressing this problem. Numerous constraints cause problems in achieving the optimal solution. In some cases, the agent returns the same solution for all different transportation logistics environments.

Consequently, the author modified the algorithm of [78] and the new reward function was equipped into it as operation state information. This step is described in the previous section. The modification aims to overcome the issues, as mentioned earlier. In addition to the development of the reward-function unit, the behavior analysis defined in Chapters 3 – 4 is used to assist in selecting the direction of the given reward. It determines whether a positive or negative reward should be given to the agent. Furthermore, it was determined with respect to the agent's behavior when it performed an action. In this context, the agent was performing the vehicle route optimization. To train the RL agent, three years of historical data were used. The RL agent learns how to perform the vehicle route optimization and adapt to changes in the environment. The author evaluates the trained model in terms of action values and routing results while vehicle route optimization tasks were performed.

This process is similar to a human (e.g., trainee and trainer). Every time an action is chosen, the trainer evaluates the result and provides feedback. The trainee then tries to avoid any actions that reduce his/her performance.

Additionally, the author applied search strategies to search for nodes. They are nodes that should be visited from the current nodes. The search strategies consist of the greedy best-first search (GD) and beam search (BS). Stochastic beam search (SBS) was proposed to solve the limitation of heuristic and local search strategies. This issue was mentioned in [72].

Algorithm 5.2: RL with model-based integration algorithm.

```
Input  : random weight $\theta$, $\theta^n$, $\phi$, $\phi^n$
Output: Route sequence ($Y^n$) for vehicle $v_n$
initialize the actor network with random weight $\theta$
initialize the critic network with random weight $\phi$
initialize $N$ thread-specific actor and critic networks with weights $\theta^n$ and $\phi^n$
    ↪ 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 $\Phi_M$
        for $n = 1, \ldots, N$ do
            initialize step counter $t \leftarrow 0$ and select vehicle $v_n$
            repeat
                select $y^n_{t+1}$ refer to $\Pr(y^n_{t+1} \mid Y^n_t, X^n_t)$
                $X^n_t \leftarrow X^n_{t+1}$
                $t \leftarrow t + 1$
            until terminated condition is matched
            reward $R^n \leftarrow model_{behav}(Y^n, X^n_0)$
        end for
        $d\theta \leftarrow \frac{1}{N}\sum_{n=1}^{N}(R^n - V(X^n_0; \phi))\nabla_\theta \log \Pr(Y^n \mid X^n_0)$
        $d\phi \leftarrow \frac{1}{N}\sum_{n=1}^{N}\nabla_\phi(R^n - V(X^n_0; \phi))^2$
        update $\theta, \phi$
    end for
    return $Y^n$
end for
```

Finally, using the formulated model in Chapters 5.2–5.3 and the methodologies outlined in Chapters 3–4, vehicle route optimization was ready to perform and test using case studies.

5.4 Case Studies

In this section, the author presents the model benchmarks and validation. The model was evaluated in terms of effectiveness and ability. The author also compared the proposed model optimization result against the real vehicle route optimization result. It is a result that a logistics agency performs.

The relationship between problem formulation and problem-solving in case studies is that each customer demand may contain multiple containers. In this case, it cannot be fulfilled by one vehicle. Therefore, the demand should be split and taken care of by other vehicles or the same vehicle if it is available until demand satisfies customer requirements (e.g., quantity and delivery time). Thus, the problem that the author solved and presented in this section is SDVRP.

According to the operation dataset for validation, it consisted of real vehicle task assignments from January 1–December 29, 2018. This dataset was used to evaluate how well the RL agent performed the vehicle route optimization tasks compared to the ground truth.

Furthermore, the agent selected appropriate actions for vehicle route optimization based on the newly inherited reward processing unit within the model. This unit also updates the reward to the actor and critic gradient function. The summary of the criteria to enable the reward function by multi-criteria ABC analysis is described in Table 5.5. This function was used by the agent.

Table 5.5: List of the reward function and parameter settings used by the RL agent.

Mode(s)	Function(s)	β	δ	γ
Feasibility Reward	$G(Z) = traversal_{cost} + \beta U(Z) + \delta T(Z) + \gamma I(Z)$	1.0	0.0	0.0
Efficiency Reward	$G(Z) = traversal_{cost} + \beta U(Z) + \delta T(Z) + \gamma I(Z)$	0.0	1.0	0.0
Feasibility and Effic.	$G(Z) = traversal_{cost} + \beta U(Z) + \delta T(Z) + \gamma I(Z)$	1.0	1.0	0.0
Fairness Reward	$G(Z) = traversal_{cost} + \beta U(Z) + \delta T(Z) + \gamma I(Z)$	0.0	0.0	1.0
Balance Reward	$G(Z) = traversal_{cost} + \beta U(Z) + \delta T(Z) + \gamma I(Z)$	0.33	0.33	0.33

In Table 5.5, the definitions of reward modes are described as follows. First, the feasibility function indicates the feasibility's degree of the solution being performed in the current environment. Therefore, a weight equal to 1 was assigned to β. Second, the efficiency function indicates that the optimized solution is considered to be efficient. This result answers whether or not the solution returns the maximum profit. However, it did not consider the feasibility of applying the solution. Third, the feasibility and efficiency function indicates that a feasible solution is returned. They were also maintaining the maximum profit. Forth, the fairness function indicates that all delivery requests are distributed equally amongst all drivers. Lastly, the balance function denotes that all factors are equally considered.

This gives rise to the question, "How does the agent select the mode to be performed?" The answer is that the author uses the multi-criteria ABC analysis to select the logistics strategy mode for the optimization process. It is when the optimization process dealing with daily optimization requirements. The methodology of [76, 77] were applied, as discussed in Chapter 5.3.3.

To demonstrate the practicality of the model, the author provided case studies to evaluate the task assignments of container trucks. These trucks are assigned to making deliveries across all regions in Thailand. An example of task assignment is shown in Figure 5.7, and there were two case studies listed in the following sections.

In addition, to demonstrate how the experimental results cause impact to the transportation logistics business, Figure 5.6 shown the basic statistics of deliveries from the year 2017 until 2019. From Figure 5.6, the transportation planning done by the company had the success rate of only half of the goods' total amount that it should make delivered to the customer.

5.4.1 Case Study with no Uncertain Changes

In this case study, vehicle route optimization was performed based on reinforcement neural model. This model deals with situations wherein no sudden uncertain changes and incidents

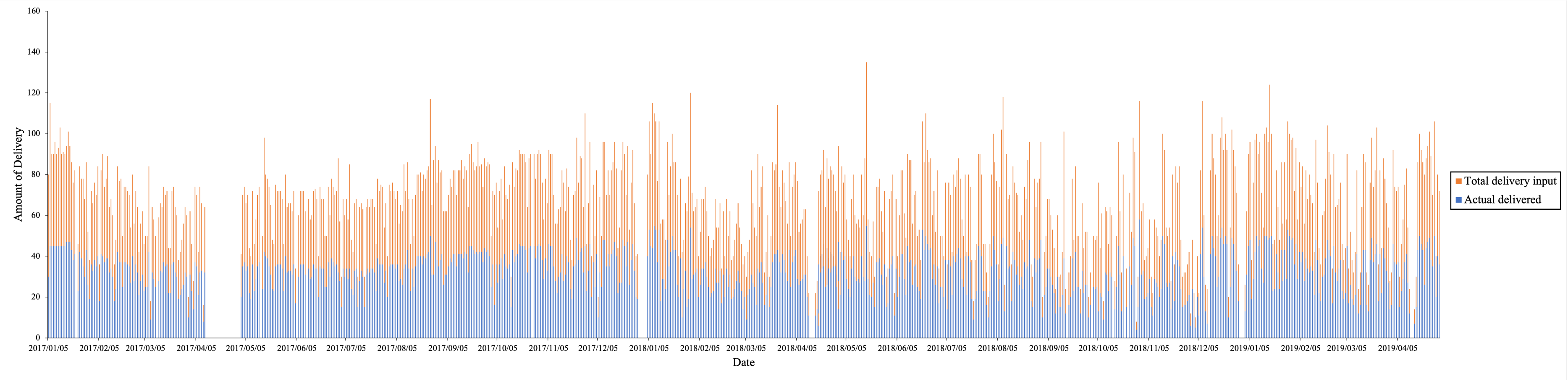


Figure 5.6: Demonstration of the goods deliveries statistics from the year 2017 until 2019. The orange bars represent the total intended deliveries, and the blue bars are the successful deliveries. Note that the blank areas denote Thailand holidays.

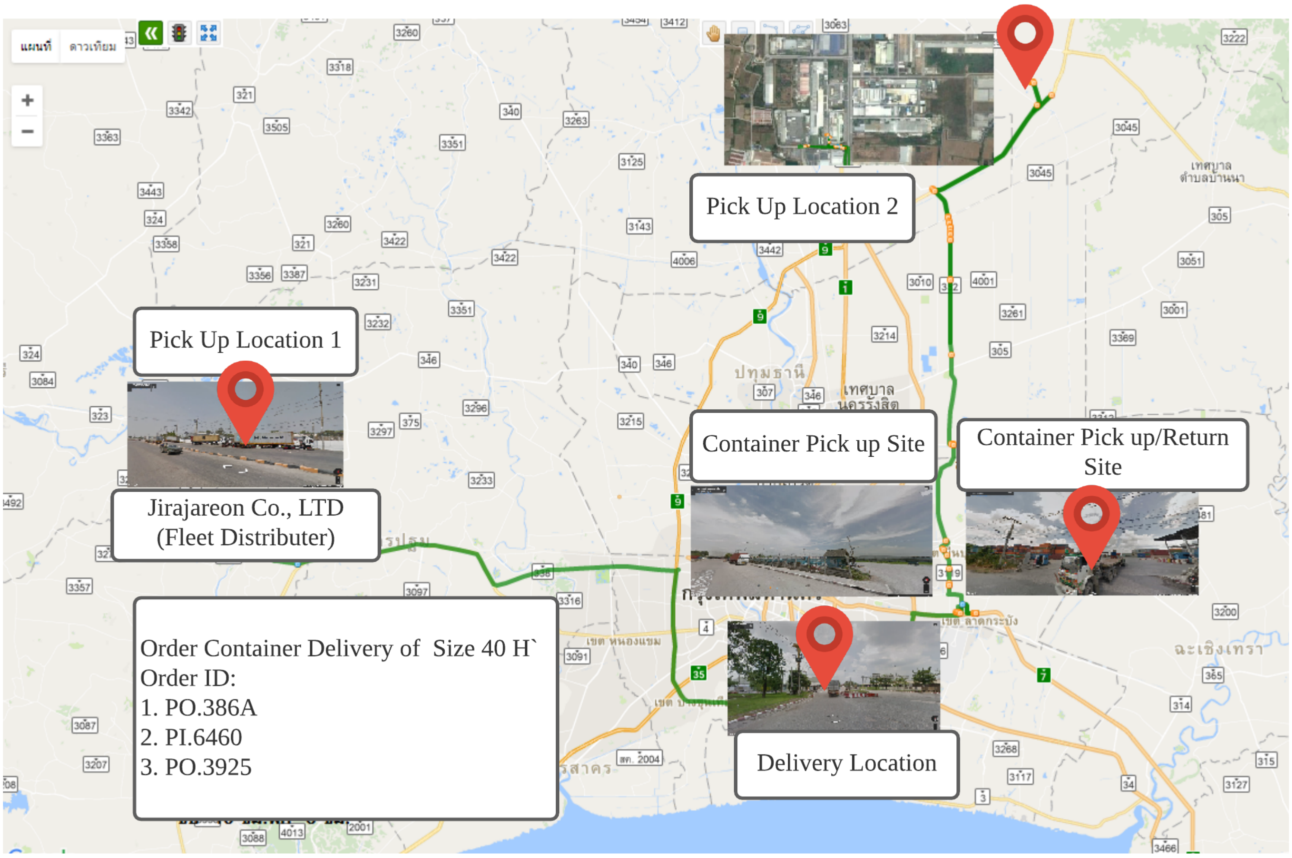


Figure 5.7: The example of routine route assignment performs by the logistics agency.

occur during daily operations. For instance: no delivery request postponement, cancellation, vehicle shortages, or human resources issues.

Generally, the logistics agency's fleet that handles the delivery is considered to be in the medium and long-haul range container delivery. Therefore, for this case study, the test data has a number of container deliveries starting from 21 deliveries with 37 vehicles available (per day) from May to July 2018. This amount of delivery refers to the actual deliveries performed by the company, as presented in Figure 5.6. Furthermore, the author obtained incident information from the report of ordering and fleet status, as shown in Table 4.2.

In the model, a new reward processing unit was used to estimate the reward for the agent that concordant to the current environment. While the reward processing unit computes the reward, it also takes the dynamic information from the environment and considers the logistics strategies. All reward settings were considered with the traversal costs, as presented in Table 5.5. The reward activation settings were selected by performing the multi-criteria ABC analysis. The author aimed to determine the suitable optimization strategy for the current industry situation by taking the dynamic information from the environment into account.

To perform this case study, the author employed the A3C and the general actor–critic algorithms with new reward processing units to perform the vehicle route optimization tasks. The author also evaluated the results of each model in terms of the profit compared with the framework in [78, 79], a well-known optimization model from the operation research that uses the local search algorithm with ML [81], and OR-Tools OSS [82]. Therefore, the efficiency of the proposed model is demonstrated. Note that the author reproduced these models and also enabled the capability for supporting split delivery tasks.

In this section, the routine vehicle route optimization tasks were performed by RL. In the next section, these tasks become more complicated because it involving uncertainties from the environment.

5.4.2 Case Study with Uncertain Changes

In this case study, uncertain changes disturbed the operations (e.g., vehicle shortage, sudden change in demand, accident, and more). Therefore, the effectiveness of the model was tested in managing the situation. Therefore, in this case, the amount of deliveries is adjusted to 60. Like the first case study, the incidents' information is obtained from the report of ordering and fleet status, as shown in Table 4.2. The incident in this scenario involved the number of vehicles that could not feasibly manage all deliveries. Therefore, the model was evaluated on how it can solve the situation. All reward settings were considered with the traversal costs, as presented in Table 5.5. The multi-criteria ABC analysis was performed for this task as well. The goal of this case study also the same as the first case study.

To perform optimization tasks using RL, the author used the training algorithm for the RL agent. It consisted of multiple agents using A3C and a single agent using the general actor–critic algorithm. The A3C method used tree-based regression and the new reward processing unit is compared. The author also evaluated the results of each model in terms of the profit improvement compared with the baselines. These baselines are also the same set as presented in the first case study. Thus, the evaluation result is to demonstrates the efficiency of the proposed model. This case study also included an incident that had been solved. In the next section, the performance evaluation metrics are presented to compared these models' solutions.

5.4.3 Performance Evaluation Metrics

In this section, the performance of the proposed model is evaluated in two stages, as follows.

First, the author evaluated the RL agent by the cumulative reward and the agent's action value, as shown in [83, 84, 85]. These outcomes indicate that a better RL model can be obtained when these metrics tend to the maximum value. The cumulative action value (CAV) can be computed using Equation (5.32):

$$CAV = \sum_{i=0}^{N} d\phi \tag{5.32}$$

where i indicates the training step, N indicates the total training step, and $d\phi$ indicates the action value of the RL agent defined in Equation (5.31).

Second, the author used the performances of humans and state-of-the-art models as baselines for evaluation. The author aims to determine the improvements when performing the same optimization tasks against the proposed RL agent. In previous studies, an improvement was called an "optimal gap." This optimal gap can be calculated using Equation (5.33):

$$Optimal_{Gap}(\%) = \frac{\sum_{i=1}^{N} Baseline_i/N - current_{solution}}{\sum_{i=1}^{N} Baseline_i/N} \times 100 \tag{5.33}$$

where i represents the baseline index, $Baseline_i$ represents the result of the state-of-the-art model, and $current_{solution}$ represents the proposed model's result.

Therefore, in this chapter, the author evaluated the RL model using the CAV of the agent at each training step. It is defined as in Equation (5.31). Furthermore, the optimization results were compared with those of the state-of-the-art models using Equation (5.33). The author also used the CI to determine the boundaries of the optimization result. CI can be computed using Equation (5.34):

$$CI = \bar{X} \pm z \frac{sd}{\sqrt{n}} \tag{5.34}$$

where $\bar{X}$ represents the mean of the optimization result, z represents the CI values selected from Table 5.6, sd represents the result's standard deviation, and n is the number of test samples. Note that Equation (5.34) and Z-values (z) are referred from [86, 87].

Table 5.6: 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

Finally, the author evaluated the multi-criteria ABC analysis task using the AUC, precision, and recall. The AUC measured the whole 2D area under the whole ROC. The AUC applies a different threshold for a comparison between the TPR and the FPR. The TPR and FPR can be calculated as Equation (5.35) and Equation (5.36), respectively.

$$TPR = \frac{TP}{TP + FN} \tag{5.35}$$

where TP denotes the number of true positives and FN denotes the number of false negatives.

$$FPR = \frac{FP}{FP + TN} \tag{5.36}$$

where FP denotes the number of false positives and TN denotes the number of true negatives. The classification result was also validated using the ground-truth data. It was evaluated in terms of its accuracy, precision, and recall. The precision and recall evaluation metrics are calculated as Equations (5.37) and (5.38), respectively.

$$Precision = \frac{TP}{TP + FP} \tag{5.37}$$

where TP denotes the number of true positives and FP denotes the number of false positives.

$$Recall = \frac{TP}{TP + FN} \tag{5.38}$$

where TP denotes the number of true positives and FN denotes the number of false negatives.

Additionally, the author used the accuracy, F_1-Score, and Cohen's kappa (k) to evaluate the proposed model. The metrics are defined in Equations (5.39), (5.40), and (5.41), respectively. The higher the evaluation value returned from these metrics, the better the model.

$$Accuracy = \frac{TP + TN}{Total} \tag{5.39}$$

$$F_1\text{-score} = 2 \times \frac{Precision \times recall}{Precision + Recall} \tag{5.40}$$

$$k = \frac{p_o - p_e}{1 - p_e} \tag{5.41}$$

where p_o represents the relative observed classification among the data points and p_e represents the hypothesis probability of the correct classification.

Correspondingly, the proposed hybrid model's efficiency was evaluated using these evaluation metrics. It was evaluated when dealing with routine and uncertain changes from the environment. Therefore, results are presented in section 5.5.

5.5 Results

After the new reward processing unit was integrated into the reinforcement neural model for the vehicle route optimization, the author executed the experiment. The author presents the results here in terms of the profit returned and compared them with the baselines.

The first baseline comprised the vehicle route optimization framework proposed by Nazari et al. [78], denoted as $Baseline_1$. The second baseline was the conventional optimization model (i.e., the vehicle route optimization model that used the local search algorithm with ML [81]). The model performed optimization using the same parameter settings, denoted as $Baseline_2$. The third baseline was the open-source software suite for routing optimization (e.g., OR-Tools), denoted as $Baseline_3$. The fourth baseline was the vehicle route optimization result performs by the company, denoted as $Baseline_4$. Finally, the vehicle route optimization framework proposed by Kool et al. [79] was used, denoted as $Baseline_5$. These baselines and the solutions obtained from the model were given as input to Equation (5.33). This equation was used to calculate the optimal gap in percentage.

The author used the optimal gap to analyze the results of the proposed model. The gap was used to determine whether it improved or degraded the overall optimization performance in comparison to both the state-of-the-art operational results from the company and software suite models (e.g., OR-Tools OSS).

Consequently, the average baseline for this experiment was computed. The average baseline is 20,575.70 THB considering the average of $Baseline_1$–$Baseline_5$. The author tested results against 1,000 samples. Therefore, the results are presented as the average of the recommended profit (THB) ± 95% of CI and the standard deviation shown in the $Result_{mean}$ and SD columns, respectively.

Furthermore, the third column illustrates the training time when the model was trained with 120,000 steps. The fourth column presents the prediction time when a set of customer coordinates was input into the model. Finally, the fifth and sixth columns present the optimal gap between the baseline and solution suggested from the model. It also includes the analysis results.

In addition to the vehicle route optimization experiment, the author also experimented to determine the appropriate classification method for the multi-criteria ABC analysis method. These results are presented in the following sections.

5.5.1 The Experimental Results from the Multi-criteria ABC Analysis method

The author used Multi-criteria ABC analysis to select the logistics management strategies. These strategies were used for the vehicle route optimization process when dealing with daily optimization requirements. The multi-criteria ABC analysis was used instead of other approaches. It is because it is simple and easy to interpret. It is also an active research area of logistics management that deals with inventory and cost analysis problems. This research area also utilizes the data to determine the strategies to be applied. Unlike the analytic hierarchy process (AHP), AHP highly depends on a system expert for the weights' assignment of data attributes and share experiences. Therefore, the methodology presented in section 5.3.3 was applied. The results are presented in Table 5.9.

After the experiment of multi-criteria ABC analysis was performed and already presented the result. Now, it is ready to presented the result for the vehicle route optimization experiments, as following sections.

5.5.2 Experimental Result of New Hybrid Reinforcement Neural Vehicle Route Optimization

Experimental Result

In this section, the author presents the experimental results of vehicle route optimization. These results are shown when the proposed vehicle route optimization model integrated with multi-criteria ABC analysis method is trained with three years of data from January 2017 – April 2019. Furthermore, the model is also tested to perform route optimization with three months of goods delivery tasks.

For the model comparison, the author compared the proposed model with the state-of-the-art models such as models from Nazari et al., Kool et al., and Musolino et al. The author reproduced these models and also enabled the capability for supporting split delivery tasks. Therefore, these models were compared in terms of efficiency and profit improvement provided to the company. Thus, the experimental results are presented in Table 5.10. Note that the model with bold color is the model that applied this study methodology. Also, "+" denotes an improvement from the baseline, otherwise "-" is given. $Baseline_1$ denotes an actual profit that the company earned when setting up the transportation plan by the staffer in the company. Later on, in the next section, these models are tested on dealing with routine and uncertainties from the environment.

Case Study with No Uncertain Changes During Vehicle Route Optimization

In this section, the experiments were performed according to the previously described reinforcement neural optimization model. The model was discussed in section 5.3.4. The weights, β, δ, and γ, of the modified reward function, presented in Table 5.5, were set to 0.33 each, with a total sum of 1. It is because each partition was equally important recommended by the multi-criteria ABC analysis method. Therefore, the author established a balance between the reward function's efficiency and fairness strategies due to the experimental results presented in Table 5.8.

After the methodology presented in Chapter 5.3.3 was applied, it demonstrated that the logistics management operations could be segmented into three classes. These processes were done by using multi-criteria decision-making. Multi-criteria decision analysis demonstrated that

Table 5.7: Reward function evaluation in vehicle route optimization.

Function	Result	Result by Company	Expected Result	Suggest Gap	Current Gap
Feasibility	36,798.95	38,909.54	57,574.11	36.08%	32.42%
Efficiency	50,052.92	38,909.54	57,574.11	13.06%	32.42%
Fairness	48,759.50	38,909.54	57,574.11	15.31%	32.42%
Balance	50,051.55	38,909.54	57,574.11	13.06%	32.42%

Table 5.8: Experiment results of Multi-criteria ABC analysis for assisting RL neural model in reward selection.

Class	Cum. delivery	Cum. request	Cum. maintenance	Cum. incident	Quantity (%)	Profit (%)	Logistics strategy
A	40,888	43,467	2,065	2,579	15.39%	17.15%	Feasibility and Efficiency
B	78,359	80,756	15	2,397	24.89%	32.96%	Feasibility and Efficiency
C	119,584	128,748	0	9,164	59.72%	49.89%	Balance

Table 5.9: The experiment results of classification performance when ML classifiers were used to perform multi-criteria decision analysis.

Model	Analysis	Kappa	Accuracy	Class	Precision	Recall	F_1-score	ROC_{Area}
Naive Bayes	ABC Analysis	0.488	0.752	A	0.731	0.979	0.837	0.984
				B	0.702	0.110	0.191	0.966
				C	0.870	0.884	0.877	0.994
Decision Tree	ABC Analysis	0.981	0.989	A	0.991	1.00	0.995	0.999
				B	0.991	0.966	0.978	0.999
				C	0.989	0.986	0.989	0.998
ANN	ABC Analysis	0.948	0.971	A	0.986	0.991	0.988	0.999
				B	0.933	0.948	0.941	0.989
				C	0.970	0.930	0.949	0.996
SVM	ABC Analysis	0.979	0.988	A	0.989	1.00	0.995	0.999
				B	0.993	0.963	0.978	0.998
				C	0.979	0.984	0.982	0.999

Table 5.10: Experiment results of trained RL agent with three years of vehicle route optimization data and testing with three months delivery tasks.

Model	$Result_{mean} \pm CI$	SD	$Compute_{Time}$	$Baseline_1$	$Optimal_{Gap}$	Analysis
Nazari et al.	17,178.48±1.24	20.972	8.01 h	24,778.85	30.67%	-
Kool et al.	17,178.48±1.25	21.11	7.33 h	24,778.85	30.67%	-
Musolino et al.	24,188.16±0.00	0.00	6.50 h	24,778.85	2.42 %	+
A3C-feas(SBS)	**24,642.90**±0.28	**4.793**	**6.25 h**	**24,778.85**	**0.548%**	+

the current information obtained from the current transportation logistics environment did not have any vehicle maintenance records and the situation also provided the highest values of profit. It also returned the lowest operational cost. In addition, due to the imbalance between the number of actual and successful goods delivered requests. Therefore, the situation of this case study was classified into the balance strategy. It is a segment of class "C."

As discussed previously, this case study's experiment follows the guideline of section 5.4.1. The results of each model are presented in terms of the number of profits improved compared to the baselines.

The results are presented in Table 5.11 and Figure 5.10. The results in Table 5.11 denote the returned profit (THB). Table 5.11 comprising six columns. The "$Result_{mean} \pm CI$" column represents the model optimization result from a given set of customers. The CI is computed to present the reliability of the returned solution. A lower CI indicates that the model is stable.

Next, the SD column presents the standard deviation of the optimization result; a lower SD denotes a stable model. The third and fourth columns present the computational time when the model was trained and tested. Finally, the fifth and sixth columns show the efficiency of the optimization model. The "$Optimal_{Gap}$" column shows how optimal the solution in the form of a percentage. It was compared against the optimal baseline (e.g., "+" denotes positive increase and "−" denotes negative decrease). The bold text in the first column indicates the proposed models that applied the proposed methodology in this chapter.

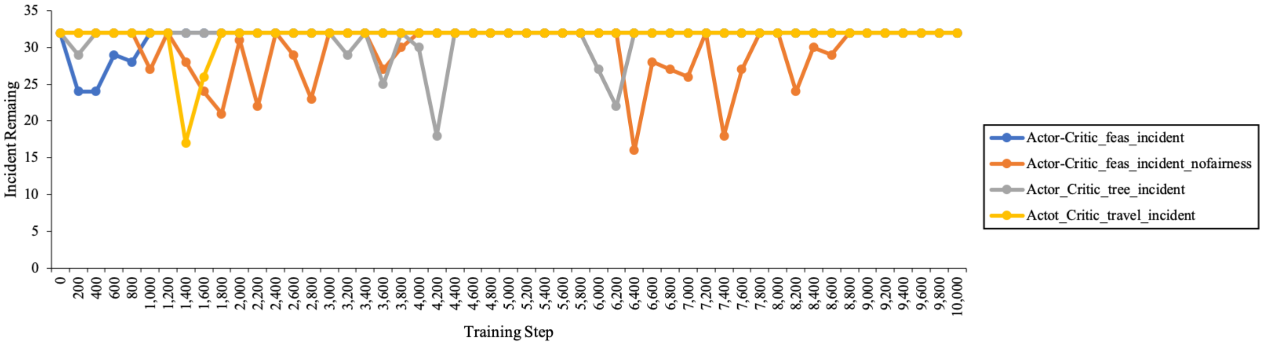


Figure 5.8: Number of remaining incidents using the actor–critic algorithm.

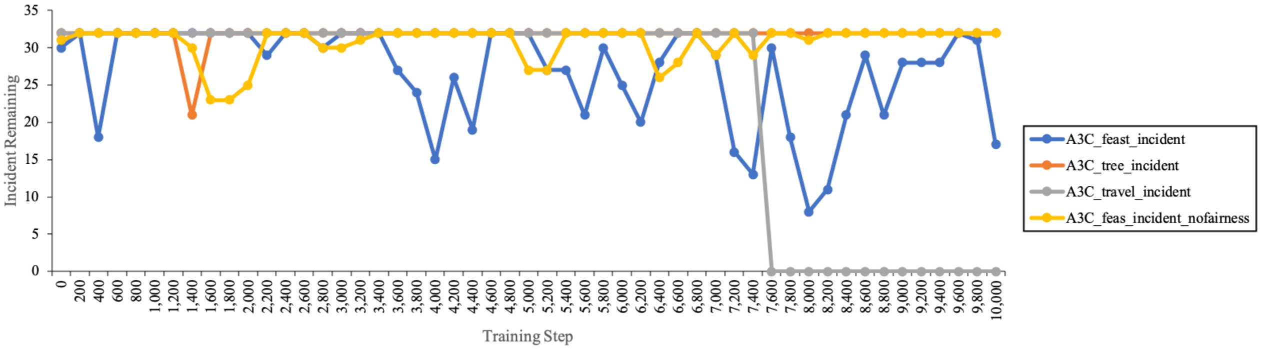
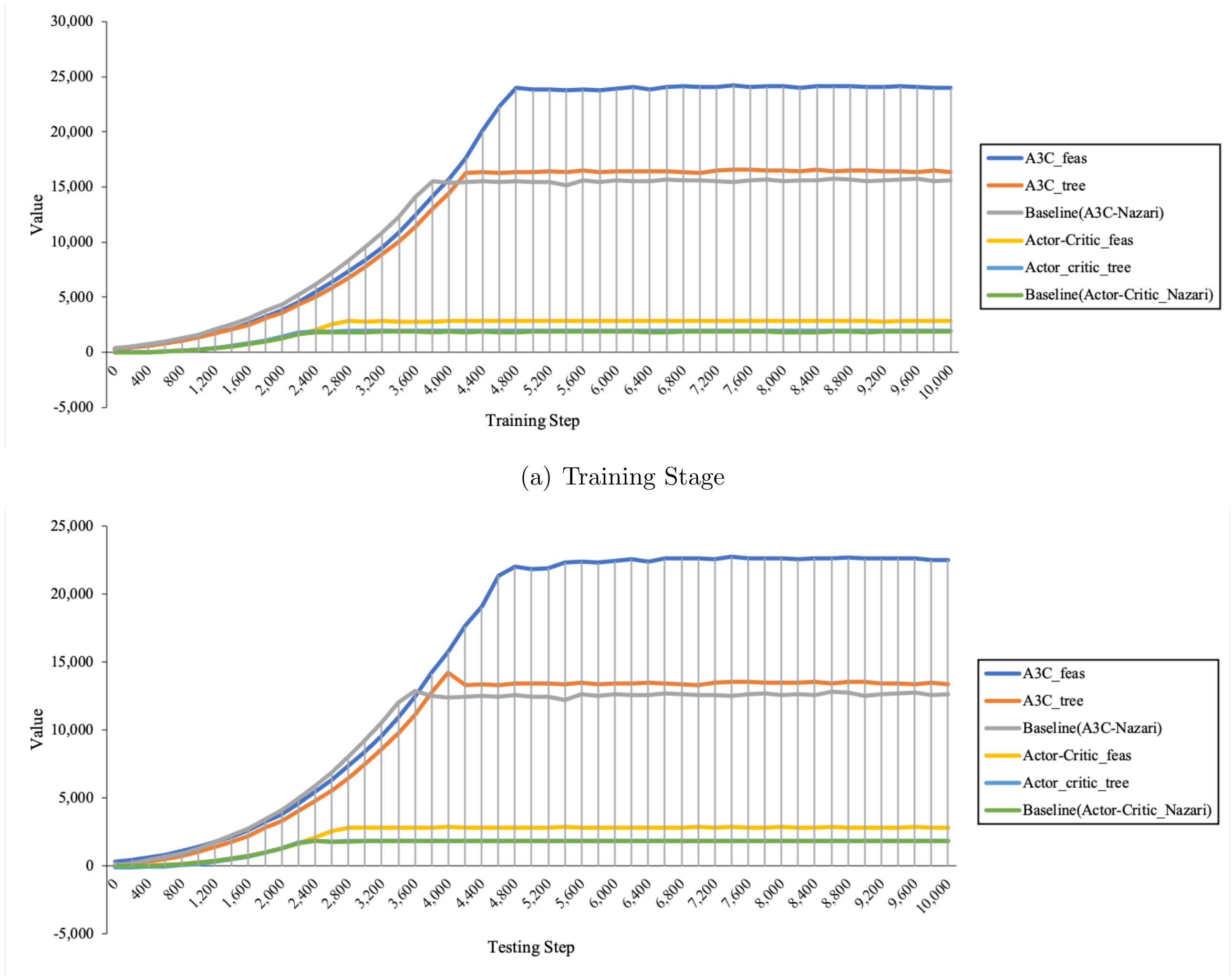


Figure 5.9: Number of remaining incidents using the A3C algorithm.

Figure 5.11 illustrated how the RL agent was used to solves the VRP problem in each step using this chapter's proposed methodology.

Table 5.11: Case study experiment when uncertain changes did not occur when the vehicle route optimization is performed.

Model	$Result_{mean} \pm CI$	SD	$Training_{Time}$	$Prediction_{Time}$	$Optimal_{Baseline}$	$Optimal_{Gap}$	Analysis
Nazari et al.	15,616.80±1.75	28.19	6.31 h	289 s	20,575.70	24.10%	−
Kool et al.	15,616.80±1.75	28.23	5.25 h	288 s	20,575.70	24.10%	−
Giovanni et al.	24,100.96±0.00	0.00	7 h	-	20,575.70	14.63%	+
Musolino et al.	27,246.74±0.00	0.00	6.26 h	-	20,575.70	24.48%	+
OR-Tools	20,156.80±0.00	0.00	7 h	-	20,575.70	2.04%	−
Company	27,387.15±0.00	0.00	-	-	20,575.70	24.87%	+
Actor-tree(GD)	4,917.13±1.3	20.96	5.35 h	282 s	20,575.70	76.10%	−
Actor-tree(BS)	4,942.90±0.44	7.07	5.35 h	283 s	20,575.70	75.98%	−
Actor-feas(GD)	5,806.53±1.03	16.57	5.50 h	270 s	20,575.70	71.78%	−
Actor-feas(BS)	5,842.41±0.33	5.34	5.55 h	272 s	20,575.70	71.61%	−
A3C-tree(GD)	16,472.12±0.58	9.38	6.44 h	390 s	20,575.70	19.94%	−
A3C-tree(BS)	16,494.89±0.06	0.98	6.45 h	391 s	20,575.70	19.83%	−
A3C-feas(GD)	27,246.74±1.33	21.38	6.49 h	379 s	20,575.70	24.48%	+
A3C-feas(BS)	27,278.42±0.47	7.52	6.50 h	380 s	20,575.70	24.57%	+
A3C-feas(SBS)	**27,381.35**±0.43	**6.98**	**6.50 h**	**384 s**	**20,575.70**	**24.86%**	+



(a) Training Stage

(b) Testing Stage

Figure 5.10: The training (a) and testing (b) stage regarding to the situation in which no uncertain environmental changes occur using the hybrid RL model.

After the case study for the routine environment was presented, it is time to presented the case study that includes uncertainties from the environment when the vehicle optimization task is performed, as shown in the next section.

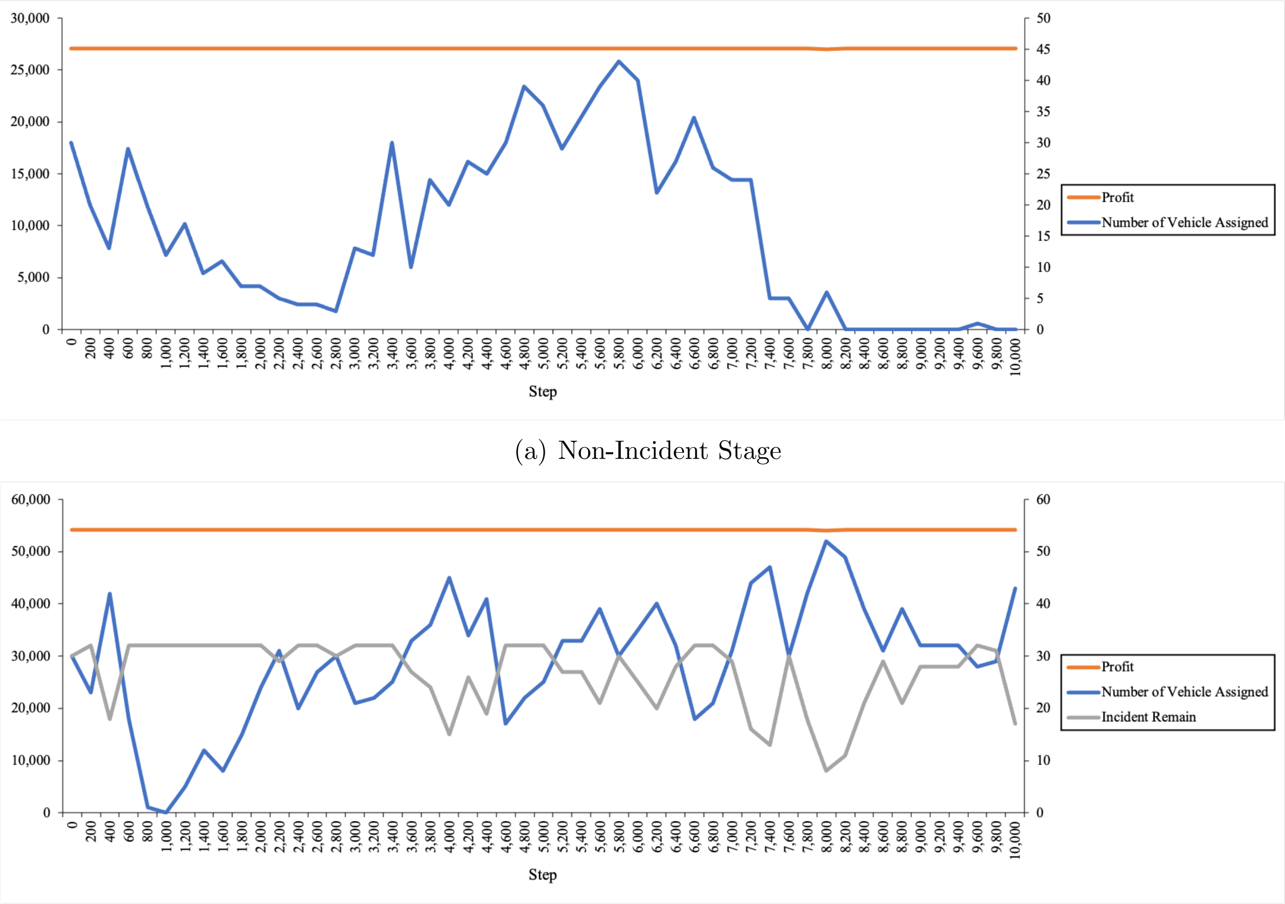
Case Study with Uncertain Changes During Vehicle Route Optimization

In this section, the experiment is discussed further. It is conducted following the guideline in section 5.4.2. It considers the case where uncertain changes occurred during the daily schedule of a vehicle. The learning parameters, β, δ, and γ of the proposed reward function, were set to 0.33 each, with a total sum of 1. The author set this setting for the same reason, as described in the previous section.

Additionally, the results of each model are presented in terms of the number of profits improved compared to the baselines. The results are presented in Table 5.12, and the training values are illustrated in Figure 5.12. The results in Table 5.12 represent the returned profit in THB. In Table 5.12, the same procedure was used for analyzing the result of Table 5.11. The bold text in the first column denotes the proposed models that applied the methodology

Table 5.12: Case study experiment when uncertain changes occur when the vehicle route optimization is performed.

Model	$Result_{mean} \pm CI$	SD	$Training_{Time}$	$Prediction_{Time}$	$Optimal_{Baseline}$	$Optimal_{Gap}$	Analysis
Nazari et al.	15,616.80±1.75	28.19	6.31 h	289 s	20,575.70	24.10%	−
Kool et al.	15,616.80±1.75	28.23	5.25 h	288 s	20,575.70	24.10%	−
Giovanni et al.	24,100.96±0.00	0.00	7 h	-	20,575.70	14.63%	+
Musolino et al.	27,246.74±0.00	0.00	6.26 h	-	20,575.70	24.48%	+
OR-Tools	20,156.80±0.00	0.00	7 h	-	20,575.70	2.04%	−
Company	27,387.15±0.00	0.00	-	-	20,575.70	24.87%	+
Actor-tree-inc.(GD)	6,869.44±1.49	24.11	5.35 h	282 s	20,575.70	66.61%	−
Actor-tree-inc.(BS)	6,892.21±0.66	10.67	5.35 h	283 s	20,575.70	66.50%	−
Actor-feas-inc.(GD)	8,651.05±0.86	13.82	5.30 h	271 s	20,575.70	57.96%	−
Actor-feas-inc.(BS)	8,680.21±0.13	2.06	5.30 h	271 s	20,575.70	57.81%	−
A3C-tree-inc.(GD)	32,959.49±0.68	10.94	6.44 h	390 s	20,575.70	37.57%	+
A3C-tree-inc.(BS)	32,990.26±0.08	1.22	6.45 h	391 s	20,575.70	37.63%	+
A3C-feas-inc.(GD)	48,171.27±1.42	22.92	6.52 h	380 s	20,575.70	57.28%	+
A3C-feas-inc.(BS)	48,201.73±0.58	9.35	6.52 h	380 s	20,575.70	57.31%	+
A3C-feas-inc.(SBS)	**48,895.68**±0.57	**9.19**	**6.55 h**	**385 s**	**20,575.70**	**57.91%**	+



(a) Non-Incident Stage

(b) Incident Stage

Figure 5.11: Non-Incident case (a) and Incident case (b) stage regarding to solving VRP.

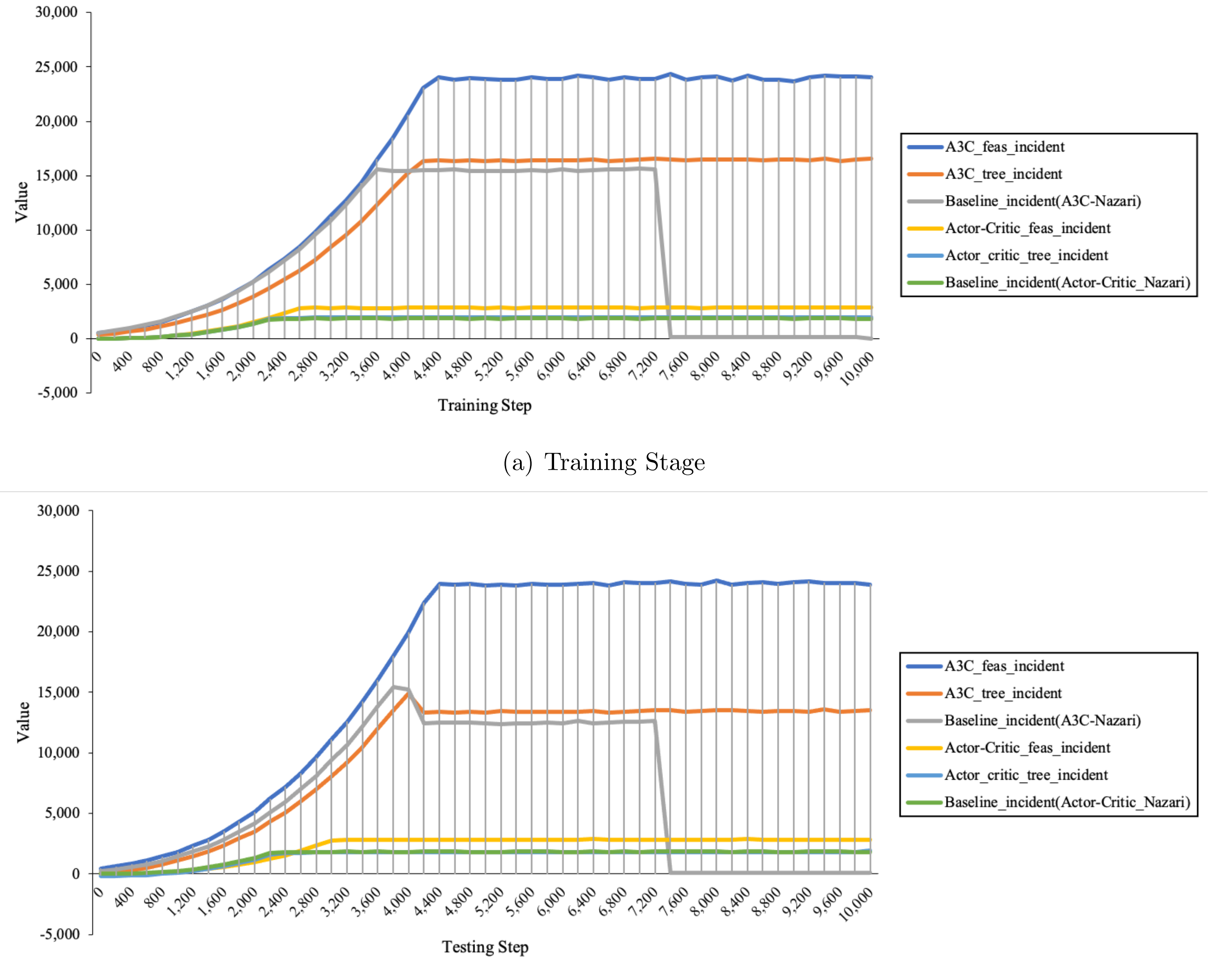
introduced in this study. Additionally, the incidents solved by the models are displayed in Figure 5.9.

Demonstration of Practicality of New Hybrid Model when Vehicle Route Optimization is Performed in Advance

In the previous section, the author compared the models' performance when the vehicle route optimization was done on day to day basis. In this section, the author uses the model to manage a situation when the goods delivery schedule is assigned in advance.

Unlike in the previous experiments, that the deliveries were planned at least 3 days in advance. This time the vehicle route optimization is performed 1 or 2 weeks in advance. The company also used this strategy in its routine operation. Thus, the author aims to demonstrate the model's efficiency in managing such complex situations. They are situations when the available information for future route planning is only the number of deliveries and the current fleet capacity. Therefore, the results are shown in Figure 5.13.

After all experiment results were presented, in section 5.6, the discussion of those experiments is presented.



(a) Training Stage

(b) Testing Stage

Figure 5.12: The training (a) and testing (b) stage regarding the situation in which uncertain environmental changes occurred using the hybrid RL model.

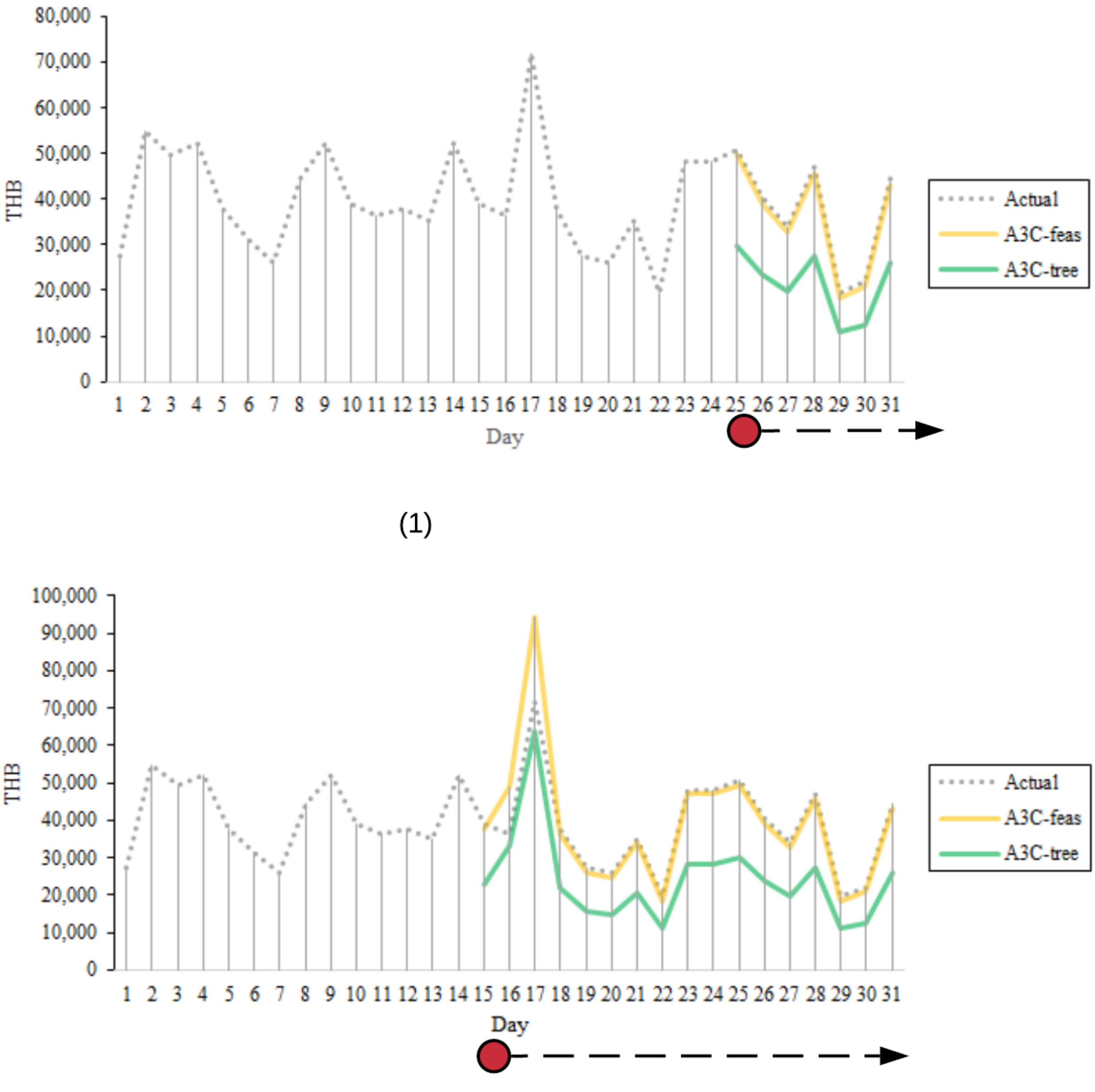
5.6 Discussion

5.6.1 The Multi-criteria ABC Analysis

From Table 5.9, results indicate that the decision tree is recommended for practical use. It is because it returns the best predictions. Each class can be described as follows.

First, for class “A”, the result showed that the number of vehicles in maintenance increases significantly over time. Therefore, this increases the consumption cost of the fleet, and the profit percentage holds at 17.19%. Suppose that this situation was to occur, then the vehicle route optimization process should consider efficiency as a top priority factor. This action is to increase the profit. Thus, the multi-criteria analysis model applied the feasibility and efficiency strategies.

Second, for class “B”, the author discovered that there are some vehicles in maintenance cycles. However, it does not affect the fleet capacity. The result shows that the fleet maintained a profit of 32.96%. It reflects that the fleet has a medium consumption cost, and 24.89% of this situation occurred during the vehicle route optimization process. Therefore, the multi-criteria



(1)

(2)

Figure 5.13: Demonstration of vehicle route optimization when plan is executed in advance. In (1), the result when vehicle route optimization is performed 1 week ahead of the schedule is shown. The red circle denotes the current dates and the arrow denotes the direction of the schedule. In (2), the result when vehicle route optimization is performed 2 weeks in advance is shown.

analysis model applied the feasibility and efficiency strategies.

Finally, for class “C”, the result showed that the number of actual vehicles requested is more than the number of deliveries to manage. Additionally, there are no vehicles in maintenance. Hence, the fleet consumption cost is low, providing the highest value profit of 49.89%. This class had the most significant portion of task quantity up to 59.72% in comparison to other states. In this situation, the multi-criteria analysis model suggested that the logistics agency must consider the optimization efficiency. It is because the successfully delivered tasks and actual customer delivery requests tasks are imbalanced. Furthermore, the logistics agency must be fair to all drivers. The reason is that some drivers will not have a task, according to the high competition among drivers. Thus, the multi-criteria analysis model applied the balanced strategies between the efficiency and fairness strategies to the vehicle route optimization process.

From this point, it is evident that the main objective of vehicle route optimization is efficiency. The add-ons or the second and third essential factors are the fairness and feasibility strategies, respectively. Unfortunately, the fairness and feasibility strategies cannot be applied separately. This phenomenon is demonstrated by the result shown in Table 5.7. It is because they impact the profit returned from the model. Based on the supporting information, fairness should be applied to an upper level, such as the company and organization level.

With vehicle route optimization, the main objective is to focus on earning more profit rather than attempting to distribute task assignments equally among drivers. However, suppose that we move to the upper company level, it becomes evident how the distribution of tasks sent to company $A, \ldots, N$ is more critical than its efficiency. In addition to these strategies, the feasibility strategy takes action as a constraint, i.e., the solution should not violate the fleet's capability.

In conclusion, the case studies presented in this chapter are fit in Class "C" when performing the procedures presented in Chapter 5.3.3. Also, the decision tree classification model is suitable to classify the given information about the current environment. Therefore, the multi-criteria analysis model applied a balanced strategy between efficiency and fairness in the reward function. The suitable weights (β, δ, and γ) were also returned. These weights were used to tune the optimization agent in different sets of environments discussed in section 5.6.2.

5.6.2 Experimental Result of New Hybrid Reinforcement Neural Vehicle Route Optimization

Experimental Results

In this chapter, the author proposes a new hybrid model for solving vehicle routing optimization. The model is not only performing the optimization tasks. However, it also reduces the model complexity. Unlike conventional optimization models, these models require numerous constraints. Thus, it is difficult to modify the model architecture to support a new customer requirement and route information. To handle this issue, the model using ML and RL agent used historical data to trained and performed the vehicle route optimization instead. Therefore, the agent who has the same role as humans can learn how to handle the situation when a new requirement from a customer is inputted into the model.

Table 5.10 shows that the trained model that used this study's methodology returned the near-optimal result compared to the baseline. However, the solution returned from Nazari et al. and Kool et al. are still far more behind. This phenomenon happened because these models optimized the route by considering only minimizing cost. Suppose that some uncertain events occurred at the current time, the recommended solution that did not take any uncertainties factors into account might not be practical and suffer from penalties fee (e.g., waiting and delay costs shown in Equation (5.12)). It is the reason why the profits returned from these two models are less than the proposed model. Note that the data inputted in Nazari et al. and Kool et al. models are the same as those inputted into the proposed model.

Similar to the proposed model, their models also used RL-based for solving the route optimization problem. The critical difference is that their models are based on a model-free RL strategy. The limitation of the model-free RL principle is that the model does not include an external environment as the model-based RL does, so the model cannot fully adapt to the current situation changes. The statement presented earlier is similar to when we were running in the gym and at the park. So the activities that we do in the gym (e.g., closed or internal environment) might not be available when we were in the public park (e.g., open or external environment). However, the proposed model combined both benefits from model-free and model-based principles together. It is the reason why the proposed model provided a significant improvement and is so-called "hybrid." For more detail about model-free and model-based RL strategies, please refer to [80].

The rationale behind this process can be explained considering the initial steps of RL. The RL agent search for the solution using SBS search strategies. It then evaluates the solution provides from the behavior analysis component. This component is part of the new hybrid model. Therefore, the behavior analysis component's results provide information to the reward processing unit, i.e., whether an agent should be rewarded or penalized.

Additionally, the author designs the reward processing unit follow the guideline of logistics management strategy (e.g., utilities, productivities, and feasibility). It also considers the current information from the fleet. For instance: the availability of the vehicle or driver and the number of incidents. These strategies were used for evaluating the reward values. This reward is given to the agent when it was performing the vehicle route optimization task. After the reward is processed, it updates to the agent using the actor–critic gradient. It is a channel to informing the agent about the direction of action to be performed. It also includes the actions that should be avoided. The process is to ensure that the cumulative reward and action values are not reduced.

The processes mentioned above are different from the conventional RL models. The conventional models using only the trial–error strategy and constraints. Also, the reward is given based on the random location selection. Suppose that the chosen location minimizes the overall distance, then the positive reward is given. Otherwise, the penalty is given and the total reward is reduced. The conventional model also does not consider the environmental information. Therefore, the model only optimizing the route based on minimum distance. The provided solution seems to be acceptable. However, it is unlikely to be usable in the real-world scenario.

Furthermore, the model from Musolino et al. has shown that the multimodal network information is crucial for solving VRP. It has reduced the gap of the suggested solution to 2.42% when compare to the baseline and model that did not use external environment information (e.g., Model of Nazari et al. and Kool et al.). Unfortunately, [88]'s model is developed on mixed-integer linear programming. Therefore, it is not easy to include all environment elements as the RL does. Thus, it is the reason why the author developed the model combined benefits from [78, 88]. This decision also contributed to an improvement of the route optimization result. The proposed model is simple and can also be used for any other route optimization problems with minimal changes.

In the next section, the practical usage of the trained hybrid model is presented in terms of case studies.

Case Study with no Uncertain Changes During Vehicle Route Optimization

Figure 5.10 shows that the agent stopped increasing the CAVs when the values started to converge. This phenomenon started in the 2,400th step of the actor–critic model. The model denoted as "Actor" in Table 5.11. Furthermore, in the 3,600th step of the remaining model in the A3C algorithm. The values were calculated using Equation (5.32). The experiment results showing that the "A3C-feas model" provided the highest values according to the performed actions. This indicates that the RL agent presented good performance while optimization was performed.

Furthermore, a comparison in terms of the profit suggested from the model presented in Table 5.11 demonstrated that all actor–critic models failed for these tasks. It is because they required more time steps than the A3C algorithm to gain experience from the environment. As a result, the obtained profit and agent's CAV were also reduced in most cases.

Moreover, the use of tree-based regression did not provide better results. From the finding, the author suggests that the use of prior experience for training an agent's behavior did not guarantee the adaptation of the agent to future environmental changes. Therefore, it is evident that when the proposed hybrid model combined the RL's strategies with the dynamic environment information, the proposed model can enhance the optimization result. These strategies consist of trial and error accompanied by dynamic information. Therefore, this combination made the behavior analysis model can capture the new environmental patterns. After the information of agent behavior is processed in the behavior analysis model, it then sent a signal to communicate the RL agent via Equations (5.30)–(5.31). Therefore, the agent's behavior is adjusted with regard to the new information. Thus, the vehicle optimization task performed by the RL can adapt to new environmental changes by doing so.

Consequently, the result obtained from the model provided a profit improvement of 42.97% over the models used in [78] and [79]. Furthermore, the author discovered that there are profit improvements of 11.98% and 26.38% for the models of [81] and [82], respectively. The models utilized by [78, 79, 82] were optimized the route with respect to minimize the traversal cost and assume that the logistics agency resources and environment are feasible for making goods deliveries. Unfortunately, there is a pitfall of doing so. In the real world situation, it is not guaranteed that the logistics agency resources and environment will always be feasible for delivered goods to customers. Therefore, when this solution was in use, the penalty fee (e.g., waiting and delay costs shown in Equation (5.12)) applied and reduced the profit because the vehicle failed to deliver goods to the customer. This is the rationale behind the reason why the obtained profits from those models were less than that of the proposed model.

The results further indicate that the recommended profit by the proposed model was close to the company's actual profit by up to 0.02%. This finding supports the proposition that this model can replace human resources in a company. The replacement can be done without any modification or monitoring. It is because the agent can be trained with human experiences based on historical data patterns and dynamic environment information.

The rationale behind this process can be explained as follows. First, the initial steps of RL were considered. Then, it searches for the solution using search strategies such as GD, BS, and SBS. Second, the environment is analyzed using the behavior analysis component's solution in the new hybrid model. Third, the behavior analysis component results then provide information to the reward processing unit (i.e., whether an agent should be rewarded or penalized when interacting with the environment).

As discussed in the previous section, the reward processing unit also considers the fleet's current information. After the reward is processed, it communicates to the agent using the actor–critic gradient. It is a channel to informing the agent about the direction of action to be performed and also includes the actions that should be avoided to ensure that the cumulative reward is not reduced.

The experiment results show that the SBS provided the highest result of up to 27,381.35±0.43 THB. It also has an improvement of 24.86%, on average. Thus, the RL neural model using SBS is suggested for practical use. The results from Table 5.11 also demonstrate that the actor–critic algorithm could not outperform the A3C algorithm in terms of profit and action values. The reason is similar to those already discussed in the previous section.

Additionally, the A3C algorithm was trained in different environments and it also consists of multiple agents. Therefore, it can explore various environments and gain more experience. The result also showed that when the author was applying this technique, it can reduce the learning time.

In the next section, the result of the case study when the uncertainties disturb the vehicle route optimization task is discussed.

Case Study with Uncertain Changes During Vehicle Route Optimization

Table 5.12 presents that the proposed hybrid model with the new reward processing unit was effective. The model provided a profit improvement of up to 68.06%. It also outperformed the models of [78] and [79]. On the other hand, the author compared the proposed model with the vehicle route optimization model using the local search algorithm with ML [81]. The result showed that the proposed model provided an improvement to the current result of approximately 50.71% of profit improvement. Furthermore, it provided an improvement of up to 58.78% of profit improvement when comparing with a solution obtained from OR-Tools OSS.

The experimental results show that the state-of-the-art models returned identical optimization results similar to the first case study. The results were optimized toward the minimum distance and assume that the resource and environment are feasible for making goods deliveries. The environmental information was also discarded from the computation of the optimal solution. Therefore, these models can not adapt to the environmental changes and return the exact solution for every iteration. This outcome resulting in penalty fees (e.g., waiting and delay costs shown in Equation (5.12)) applied and reduced profit. These results were also less than the result of the proposed hybrid model when optimization was performed.

The rationale behind this improvement is that when the proposed model performs the optimization tasks using different case studies, it also considers the environmental information and adapt concordant to the new environment and management policies. This change is made by the new dynamic reward function.

To demonstrated the phenomenon presented above, Figure 5.9 shown that when the proposed model was used, the number of the incident was significantly reduced compared with other model designs. Figure 5.11 supports the prior statement by illustrated that the proposed model is efficient to learn the current environment and solve VRP. It is because the model can maintain a high profit across the performing iterations. For the incident case, the number of the incident is significantly reduced from 32 to 10 remaining incidents.

Therefore, it can be concluded that this model effectively manages the uncertain environmental changes to some extent. The experiment result also demonstrated that the reward function was vital when the model is applied to the real-world scenario. It was clear that the agent's rewards impacted how an agent's action was selected during the training step.

Figure 5.12 demonstrated the RL agent's behavior in each training step. The use of the A3C algorithm for training the agent with the proposed methodologies and reward function returned the highest action values. This training algorithm provided the action value up to 25,000 and more than the A3C algorithm using tree-based regression methods. However, the tree-based regression methods predicted the rewards according to prior experience but discarded the current environment information. Therefore, the RL agent cannot fully adapt to the new environment, resulting in a reduced cumulative reward.

Furthermore, the proposed model provided the highest profit recommend to the company, up to 48,895.68±0.57 THB with an improvement of 57.91%, on average. This solution was obtained using the SBS strategy for RL agent. Thus, the new reward function is significant for the RL neural optimization model and provides the best solution for two reasons.

First, the information required to evaluate the solution was based on logistics strategies. It consisted of the feasibility, efficiency, and fairness of the optimization solution. Furthermore, prior experiences were used to determine whether a reward or penalty should be given to an RL agent.

Second, the agent actions were not selected for the network nodes to decode based on the traveled distance. For the proposed model, the author designs the agent to select nodes from the availability of fleet capacity, number of orders flowing into the system, available drivers, traffic, and the vehicle's current position in addition to the traveled distance. The RL agent also used prior actions as experiences for dealing with incidents. These experiences influenced the decisions while it is limited to the conventional optimization model.

At this point, this gives rise to the question, "How does the agent know the current status of the fleet?" The answer to this question is that the agent uses trial and error to explore and interact with the environment in the first step. After that, the environment is analyzed using the behavior and root-cause analysis model, as shown in Figure 5.1. The behavior analysis model used obtained data from multisources to model the current transport environment. This model is also used to inform the RL agent on how well it handles an unusual incident (e.g., accident, vehicle shortage, absence of drivers, or order postponements and cancellations from customers). It also includes the newly assigned task in the current environment. The behavior analysis model informed the agent whether its action is suitable for optimizing the vehicle route in the current environment or adjustment was required. The way to inform the agent of its behavior is discussed as follows. First, this information from the behavior analysis model was transmitted through the reward processing unit to evaluate the vehicle route suggested by RL in terms of feasibility, efficiency, and fairness with respect to the current environment. Second, the reward processing unit used multi-criteria ABC analysis with a decision tree to determine the appropriate logistics strategy that could be implemented in the reward function and for adjusting the optimization strategies used by the RL agent. This selection was made with respect to logistics management criteria. Finally, the reward processing unit sent a signal to update the actor–critic gradient, computed as Equations (5.30)–(5.31). Hence, the RL agent's action for optimizing the vehicle route was adjusted concordant to the new strategy.

The RL agent performed the vehicle route optimization task until a feasible solution was returned or the terminate condition was satisfied. The question posed above is the rationale behind the process that makes the proposed model more efficient when dealing with dynamic information and uncertain environmental changes.

From the findings, the author discovered that in real-world applications, the vehicle route optimization process must consider the utility and productivity of the deliveries together with the minimum traversal cost. It should be noted that the minimum traversal cost is not always guaranteed to be the optimal solution. To strengthen the findings, the results were validated against the company's actual profits. The experimental results show that the proposed model improved up to 43.99% of profit improvement when the real company operations were compared. Thus, it was clear that the RL agent in the model can seamlessly replace the human resources. This outcome does not require any further modifications. The author also discovered that using previous vehicle route optimization solutions as RL agent experiences could help the agent determines optimal nodes. They are nodes that should be visited in the current environment. Unfortunately, in some cases, the situation is significantly different from the past; only prior experience was not always practical. Therefore, past actions cannot be applied in a similar manner to different environmental situations. It can be concluded that essential information that is updated dynamically and management policies is required to manage different environments. This presents gaps that are fulfilled by the new hybrid model.

Thus, this is the motivation behind the hybrid model proposed in this chapter. The experimental results also indicate that a hybrid A3C algorithm model using the SBS strategy is suitable for practical use. It is because it exhibits excellent performance compared to other approaches for the optimization of delivery tasks in most cases.

In this section, the author presented the experimental results and case studies for test the RL agent when performing vehicle route optimization tasks in different situations. In the next section, more complicated tasks are presented.

New Hybrid Model when Vehicle Route Optimization is Performed in Advance

From the previous section, the hybrid A3C algorithm model using the SBS strategy was chosen for performing the vehicle route optimization tasks. In this section, the same model was used to perform the advanced planning of the vehicle route optimization. The planning is about 1 – 2 weeks in advance.

Figure 5.13 showed that the new hybrid model could manage the situation when the plan is appropriately set in advance. This advanced planning was also dealing with the deliveries that have incidents. The results indicated that the differences between the proposed model's and actual planning's results are 2.90% and 6.46%, on average. These results are 1 and 2 weeks of advanced planning, respectively.

This experiment results can be explained by the fact that the agent is trained based on the experience of the staff. It also includes the information required for forecasting the fleet capacity. This forecasting was done in the behavior analysis component. Therefore, when dealing with routine situations, the agent behaved similarly to humans.

Unfortunately, the staff obtained the information of the incident on the day that the tasks had to be executed. Therefore, the staff could only utilize past experiences to support their decisions. This is supported by the fact that the staff and the A3C-tree model results are similar. This phenomenon is shown in (2) of Figure 5.13. The ML model used by the A3C-tree was trained based on prior experiences. Therefore, the agent used the ML model to predict the outcome of the selected action.

However, the strategy was different for the proposed model (the "A3C-feas" model), shown in (2) of Figure 5.13. The model managed the situation more efficiently than the A3C-tree. It is because it used the behavior analysis component results to forecast the situation in the next time step. For example, the model considered the trends of customer demands, such as "Will it increases or reduces?", "Will it be normal or abnormal?", "Where are the fleet's vehicles in the maintenance cycle?", and "Is the number of vehicles sufficient to manage the customer demands?" Thus, the "A3C-feas" model maximized utility from the solution using the logistics management strategies according to the forecasting's result of the observed time step.

In conclusion, this section's findings supported the proposition as follows. If the company planned the vehicle route optimization in advance, then the forecasting result from the fleet's capacity would be crucial. It is because it significantly improves the result of vehicle route optimization than those models that were only used the prior knowledge. This knowledge was extracted from past actions. Thus, companies can understand a situation beforehand and accordingly prepare the resources.

After the experimental results were discussed, it is now ready to conclude this chapter's contents, as shown in the conclusion section.

5.7 Conclusion

In this chapter, the author presented a hybrid model based on RL for optimizing vehicle routing. This routing is a route of logistics agency container delivery in Thailand. The author aims to solve the model complexity and efficiency issues with the proposed hybrid model. These issues occur when route optimization is performed in the real-world scenario. After the methodology of RL was applied, in the initial step, the RL agent performed a solution search using a trial-and-error strategy. The action of the agent made to the environment was then evaluated through the behavior analysis model in the next step. These results were inherited to compute the reward for the agent using MUI. Lastly, to select the type of reward to give, a multi-criteria ABC analysis was utilized. It then updated the actor and critic gradient to prepare the solution search in the next time step. This process was known as a channel for communicating with the RL agent.

The experiment results showed that the proposed hybrid model with the proposed methodology was superior. This phenomenon happened when performing vehicle route optimization tasks in comparison to previous approaches. The proposed hybrid model provided the optimization result close to optimal up to 0.548% gap.

The findings from this chapter show that the use of dynamic information from the environment with prior experience and trial-and-error strategies increased the ability of the agent. The agent was able to return improvements on vehicle route optimization's solution. These improvements were shown for both routine and non-regular cases.

The essential part of improving the experimental results is the behavior analysis component. It is a deep neural network that was constructed and executed to reveal the RL agent's and operation's behavior. This behavior information was used to assist and adjust the RL agent when the vehicle route optimization process was performed. After numerous time steps had passed, the knowledge was discovered by the agent. Thus, the route optimization process was performed using the obtained knowledge. This knowledge was further used to train the RL agent in the current time step. The agent then continues in the optimization tasks until the terminate condition was reached.

This study revealed that the insights from the data and the current information from the environment are crucial for training the RL agent and compulsive RL agent behaviors in the route optimization process. Therefore, the agent is then adapted to avoid selecting the actions that reduced its value and reward.

To demonstrate the practical significance of the proposed model, the experimental results were validated using real operational data. These data were given by the company staff. They consist of reports on vehicle scheduling and disturbances in the vehicle route optimization process. In this chapter, the author also reproduces the state-of-the-art models. Finally, these models were used to compare with the proposed model.

The author performed and demonstrated the practical significance of using case studies. The experimental results indicated that a new hybrid model of the proposed methodology with A3C is suitable for practical use. It is because the model has significantly improved results over other approaches.

Furthermore, the hybrid model was efficient for optimizing vehicle routing, mainly when uncertain changes occurred. These cases are also compared to previous frameworks and well-known optimization algorithms. For instance: the local search with ML. The hybrid model recommends an optimal route with profit for the company up to 27,381.35±0.43 and 48,895.68±0.57 THB. It also includes a profit improvement of 24.86% and 57.91%, on average, for the non-incident and incident cases, respectively.

This study delimits only the transportation planning perspective. Nevertheless, it is a part of logistics management. In the future, the author will consider using the multi-criteria approach to select route optimization criteria that are not limited only to transportation. These criteria referred to users' preferences and manufacturing factors (e.g., production timeline and manufacturing requirement). This modification would make the proposed optimization model more robust for routing optimization problems. Furthermore, they consist of situations when dealing with the upscale uncertainty from the environment.

From the obtained findings, the potential for transportation does not only rely on the transportation level. However, manufacturing and production processes are also essential parts of being considered to increase operational efficiency.

Chapter 6

Dissertation Contribution

6.1 Practical Implication

(a) More accurate abnormality detection in the transportation planning process increases opportunities for logistics agencies. It enables logistics agencies to better understand operation behavior so that policies to manage transportation logistics are distributed effectively. This benefits the agency because the more feasible the policies are to handle goods delivery to potential customers with the optimal route, the lower the operational costs to the logistics agency performing the deliveries.

(b) This dissertation proposes methodologies that can support logistics management decision making involving optimizing vehicle routing (e.g., fleet utilization, productivity, and difficulties).

(c) Logistics agencies can use the proposed methodologies to make suitable policies and take appropriate actions to assist or substitute expert staff in eliminating any disturbance before it causes difficulties in the transportation planning process.

6.2 Theoretical Implication

1. The first academic contribution of this dissertation is the development of a new behavior analysis model for detecting temporal anomalies from time-series data using LSTM.

   (a) Recent research proposed an LSTM to analyze time-series data for abnormalities [3, 4, 24]. LSTM has a dynamic threshold for detecting anomalies in time-series data [18], uses AE to enhance LSTM results as LSTM-AE [19] and the concept drift adaptive method for improving detection results [22]. However, the proposed model suffers from false alarms or increases in the false positive rate. Additionally, detecting anomalies that are conditional on spatial-temporal context and behavior attributes is still limited. Therefore, this study proposes thresholds that include conditional and behavioral attributes. It enables the detection model to detect more complicated anomalies that are correlated to spatial and temporal conditions.

   (b) In most studies of temporal anomaly detection, the models are efficient in detecting point anomalies. However, these same models cannot be used to detect contextual and collective anomalies. Therefore, the LSTM with the contextualized dynamic threshold proposed in this dissertation enhances the detection result and reduces the false positive rate. It also expands the ability to detect various types of anomalies in the same model.

2. The second academic contribution of this dissertation is the development of a new hybrid model for behavior analysis jointly using LSTM and LSTM-AE with clustering. This methodology is essential in assisting the RL agent described in Chapter 5 in tuning its behavior while optimizing its fleet vehicle routing.

   (a) Recent research proposed a model that used LSTM-AE to perform data dimension reduction and input the low-dimensional data to clustering (e.g., K-means and GMM [34, 5]). However, this would be more accurate if the model was to combine the result with another model to reduce bias and false detection rates. This process also added the capability to deal with high-dimensional and nonlinearly separable data. Therefore, this dissertation proposes an ensemble method.

   (b) In most studies of hybrid models for detecting abnormalities, the models combine different detection strategies. This study proposes a hybrid model that takes the unique advantages of distance-based, cluster-based, and statistical-based ideas to combine the detection results, making them more accurate than when using a single model.

3. The third academic contribution of this dissertation is the development of root-cause analysis. In previous studies, this process is also called anomaly explanation. The development of the root-cause analysis in this study consists of forward and inverse problem-solving techniques and the ML approach.

   (a) In recent research, it was shown that processes that describe or provide some hint of an abnormality in the system are still limited and currently under development. Therefore, sometimes there is a bias in defining data points regardless of whether they are abnormalities. In the worst case, the data point is mistakenly defined as abnormalities. Therefore, the false positive rate is increasing significantly. These issues also impact the information returned to support agency decision making. It contains a high risk of making an incorrect decision from inaccurate information. In addition, performing a root-cause analysis adds the capability to trace back to the origin of the anomalous events. Therefore, anomalies are detected and eliminated until their origin. This action prevents any disruption and failure of the transportation planning processes.

   (b) The effectiveness of the proposed methodology was presented through case studies. The findings indicate when the root cause of the abnormalities is effectively revealed. The proposed solutions for the transportation planning process performed by the hybrid RL model can overcome the general reinforcement neural optimization model proposed by Nazari et al. and Kool et al. It also includes the vehicle route optimization model using local search algorithm and ML by Giovani et al., and the OR-Tools on handling an uncertain change in the logistics transportation environment. Furthermore, when compared to the case in which vehicle route optimization is performed by staff, the information regarding the abnormalities and root causes can alert the model to avoid choosing an action that reduces the cumulative profit. Therefore, the solutions returned from the hybrid RL optimization model have a higher percentage of optimal and overcome the solution provided by the company.

6.3 Contribution to Knowledge Science

(a) The methodologies in this study enhance the knowledge in the data mining and artificial intelligence research area. They are enhanced by the new novel methodologies.

(b) Novel methodologies for detecting anomalies and their root causes contribute to knowledge discovery from multisource data, such as real-time data streaming from multiple sensors and the IoT.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In Chapter 2, the author presented and defined a fundamental challenge in detecting an anomalous event in a vehicle route optimization process for transport logistics. It leverages the benefits of innovative anomaly detection approaches.

In Chapter 3, the LSTM approach was proposed and is suitable for detecting temporal anomalies in time-series data. Simultaneously, the author addressed the challenge involved and the remaining research question associated with unlabeled multidimensional datasets. The process also includes their interpretability. It was shown that the problem is related to detecting anomalies from nonstationary data series with numerous anomaly scenarios in urban transportation logistics. To address this problem, a novel contextualized dynamic threshold approach was proposed. It does not depend on any labels and entirely unsupervised. This approach's capability was also extended to detect complex anomalies involving multidimensional factors. The proposed approach also identifies the causes of abnormal events.

The empirical results showed that the data do not have any dependencies between time steps; furthermore, the anomaly detection model required a specific context to detect anomalies, as the AUC, F-Score (F_1-score), precision, and recall metrics increased significantly to 0.870, 0.839, 0.836, and 0.842, respectively. The obtained result significantly improves the detection result from the model proposed by [18].

Nevertheless, the proposed model is simple and can be applied for any other anomaly detection problem with minimal changes. For instance, minimal changes in location-time contexts (e.g., events at a location and specific time) and thresholds of behavior attributes (e.g., resource usage, number of workflows, and transactions) should be considered.

In Chapter 4, the hybrid unsupervised model was proposed to detect and identify the cause of anomalies in the route optimization process of urban freight transportation. From the study of behavior and root-cause analysis, it was discovered that each model has a different capability when detecting abnormalities from the dataset. Moreover, the anomaly explanation is still under development. Therefore, the author proposed LSTM-AE to reveal the significance of the data's feature representation (e.g., latent features). After that, the data's significant feature representations are input to the distance- and density-based detectors. This procedure solved the problem of high-dimensional and nonlinearly separable data. The problem previously mentioned is limited to the conventional ML and threshold-based approach (e.g., reconstruction error does not easily find boundaries for separation between normal data points and anomalous data points).

The author combined the results of multiple detectors from different properties to significantly increase the effects of the detection rate. Additionally, models for explaining the cause of anomalies were proposed to reduce bias in detecting datapoints. The root-cause analysis models were proposed. The approach consists of forward and inverse problem analysis, including root-cause analysis using ML. These three models were executed and compared with the baselines. The author selected the model that had the highest performance (e.g., AUC, F_1-score and the computational time) in detecting anomalies and defining their root causes. Note that the GPS probe data presented in Chapter 3 were used as input into these detection models.

From the empirical results, the detection rate increased and was more accurate than using a single model. The author discovered that the optimal number of classifiers or detectors is five. This number had the best performance when detecting abnormalities in both temporal and static types. Therefore, temporal LSTM-AE with four detectors (i.e., GMMs EM/PCA-EM-DATA/Kmean/IS) was suggested when deploying the model because it obtained the highest AUC, F-Score (F_1-score), precision, and recall. This information was crucial for later chapters because it was used to predict the trend from the data. For example, if vehicle route optimization was performed in advance, the behavior analysis model predicted the outcomes from the given solution of the RL agent, determining whether there would be abnormalities in customer demand or fleet capability, for instance, vehicles in maintenance or a high level of demand from the customer. Therefore, the agent adapted to these changes and alerted the company to revise its fleet planning to match the demand if the current number of vehicles was insufficient.

After the anomaly detection results were obtained, they were used as an input in the root-cause analysis model. The result shows that the model using ML outperformed the exact and approximate-based model, such as forward and inverse problem analysis, proposed by [54, 52].

The reason is that the PNN is a well-known ML model for performing pattern recognition. Therefore, the author adapted the benefit of pattern recognition to the root-cause analysis problem. Instead of searching the relationship between data attributes such as inverse and forward problem analysis, the PNN stored the data pattern in the pattern and summation layers. It further classifies the type of root cause using a probability distribution. Thus, a more accurate classification result is returned.

The experimental results are validated with real data from reports of disturbances in the route optimization process by company staff, showing an accuracy of up to 0.83 (AUC of 0.888) with less processing time than that required by other existing methods. The cause of the anomaly is perfectly revealed. As the detection decision is explained, it makes the user clearly understand the relation as to why the model reached this such decision.

The anomaly is also detected and eliminated until its origin. This procedure is performed before the anomaly causes any disruption to the transportation planning processes. The experimental results also showed that the proposed hybrid model is typically general and can be applied to other applications with some modifications. The model still maintains a high performance up to 0.88 AUC in numerous anomaly detection applications.

In Chapter 5, case studies applying behavior and root-cause analysis were presented. The author demonstrated how behavior and root-cause analysis played an important role in vehicle route optimization in real company operation. The first case study was when the optimization was performed based on the neural reinforcement optimization, which handles situations wherein no sudden and uncertain changes occur during the daily operation. The case study used a normal environment (no traffic congestion) with no incidents occurring in the transport container fleet (e.g., no order postponement, cancellation, vehicle shortages, or human resources issues). The author also tested the model with an environment where uncertain changes occurred in the second case study.

This study assumes that the RL agent has a role similar to humans for performing daily vehicle route optimization. Therefore, training is required to achieve the goal. The model equipped with the proposed behavior and root-cause analysis was compared with the model that does not have behavior and root-cause analysis. In addition, the experimental results were also compared to state-of-the-art approaches.

In the experiment, the result implied that the interconnection between RL, behavior analysis, and reward processing of the proposed model increased the ability of the agent to perform route optimizations in a similar way as humans for routine daily scheduling. When uncertain changes occurred in the environment, the agent outperformed the humans when making rescheduling decisions. This is because the route optimization model not only considered the prior experience and search for the solution but also factored in dynamic information from the environment when evaluating which customer should be visited and which delivery should be performed.

Furthermore, A3C provided an impressive result when training the agent compared with the general actor-critic algorithm, which measures using baselines [78, 79], well-known local-search optimization algorithms with ML [81], and an open-source software suite for optimization (e.g., OR-Tools). The improvement in the route optimization result implies that there are benefits to combining the RL trial-and-error strategy, behavior analysis, and a reward processing unit when fine-tuning the agent's behavior while selecting actions to perform. Furthermore, the model also tested the use of vehicle route optimization to plan schedules. The results showed that the proposed hybrid model is robust in deciding both the next time step and daily routine planning.

7.2 Future Work

According to the limitations of the experiments presented in this dissertation, there are opportunities for further studies to improve this methodology for behavior and root-cause analysis with application to transportation logistics. They are listed as follows:

1. The proposed model in Chapter 3 only used an LSTM-based model. However, there are many time-series prediction models that can be explored further.

2. The essential factors in Chapter 3 can still be improved and further evaluated. There are more features in logistics management strategies (e.g., human resources, production and supply, and manufacturing timelines) that can be further explored. The author aims to expand the proposed approach's capabilities. This expansion also includes implementing the framework, which takes other operational areas into account. This area also involves the streaming of data. This improved methodology will enable more reliable and efficient decision-making.

3. In Chapter 4, the noise that influences the false positive rate remains. It also causes anomaly misclassification, including determining its cause. Therefore, methods to address this issue are required for further investigation in future work.

4. The scenario of determining root-cause analysis in Chapter 4 was considered when the disturbance occurred in transportation planning. However, it is worth exploring and improving the model. The improvement will increase the model's ability to deal with a scenario in which the disturbance occurs in other components of logistics management

(e.g., production, demand, and resources planning). The explanations of the causes are also included and more precise than before.

5. In Chapter 5, the author tested the behavior and root-cause analysis model with real case studies. These case studies were conducted with the logistics agency in Thailand. The case studies also covered only vehicle route optimization tasks. However, it is worth exploring and investigating more in future work when vehicle route optimization tasks consider the manufacturing and production timelines. The supply chain's efficiency relies not only on manufacturing and production but also on the effectiveness and potential of transport logistics. Therefore, the proposed model can expand the capability to handle these tasks. As a result, the potential in-vehicle route optimization can be improved.

Bibliography

[1] DHL, "8 TRENDS THAT ARE DISRUPTING LOGISTICS TRANSPORTATION," 2020. [Online]. Available: https://www.dhl.com/content/dam/dhl/global/dhl-supply-chain/documents/pdf/glo-dsc-eight-trends-disrupting-logistics-transportation.PDF

[2] J. Macaulay, L. Buckalew, and G. Chung, "Internet of Thing in Logistics," DHL Customer Solutions & Innovation, Troisdorf, Germany, Tech. Rep., 2015.

[3] R. Chalapathy and S. Chawla, "Deep Learning for Anomaly Detection: A Survey," *ArVix*, 2019. [Online]. Available: http://arxiv.org/abs/1901.03407

[4] G. Pang, C. Shen, L. Cao, and A. v. d. Hengel, "Deep Learning for Anomaly Detection: A Review," *ACM Computing Surveys*, vol. 1, no. 1, pp. 1–36, 7 2020. [Online]. Available: http://arxiv.org/abs/2007.02500http://dx.doi.org/10.1145/3439950

[5] Bo Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection," in *Sixth International Conference on Learning Representations*. Vancouver, Canada: OpenReview.net, 2018, pp. 1–13.

[6] A. Gulenko, O. Kao, and F. Schmidt, "Anomaly Detection and Levels of Automation for AI-Supported System Administration," in *Communications in Computer and Information Science*, vol. 1070 CCIS, 2020, pp. 1–7.

[7] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," ACM, Tech. Rep. September, 2009.

[8] L. Rosa, T. Cruz, M. B. d. Freitas, P. Quitério, J. Henriques, F. Caldeira, E. Monteiro, and P. Simões, "Intrusion and anomaly detection for the next-generation of industrial automation and control systems," *Future Generation Computer Systems*, vol. 119, pp. 50–67, 2021.

[9] Y. Wang, M. Perry, D. Whitlock, and J. W. Sutherland, "Detecting anomalies in time series data from a manufacturing system using recurrent neural networks," *Journal of Manufacturing Systems*, 2020.

[10] N. Carneiro, G. Figueira, and M. Costa, "A data mining based system for credit-card fraud detection in e-tail," *Decision Support Systems*, vol. 95, pp. 91–101, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.dss.2017.01.002

[11] J. Jurgovsky, M. Granitzer, K. Ziegler, S. Calabretto, P. E. Portier, L. He-Guelton, and O. Caelen, "Sequence classification for credit-card fraud detection," *Expert Systems with Applications*, vol. 100, pp. 234–245, 2018. [Online]. Available: https://doi.org/10.1016/j.eswa.2018.01.037

[12] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, “Data mining for credit card fraud: A comparative study,” *Decision Support Systems*, vol. 50, no. 3, pp. 602–613, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.dss.2010.08.008

[13] 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 *UbiComp/ISWC 2019- - Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers.* London, UK: ACM, 2019, pp. 996–1007.

[14] L. Liu, C. f. Li, X. k. Sun, and J. j. Zhao, “Event alert and detection in smart cities using anomaly information from remote sensing earthquake data,” *Computer Communications*, vol. 153, pp. 397–405, 2020.

[15] F. Cauteruccio, L. Cinelli, E. Corradini, G. Terracina, D. Ursino, L. Virgili, C. Savaglio, A. Liotta, and G. Fortino, “A framework for anomaly detection and classification in Multiple IoT scenarios,” *Future Generation Computer Systems*, vol. 114, pp. 322–335, 2021.

[16] C. Yin, B. Li, and Z. Yin, “A distributed sensing data anomaly detection scheme,” *Computers and Security*, vol. 97, 2020.

[17] J. Ko and M. Comuzzi, “Detecting anomalies in business process event logs using statistical leverage,” *Information Sciences*, vol. 549, pp. 53–67, 2021.

[18] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, “Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding,” in *KDD 2018, August 19-23, 2018, London, United Kingdom.* London, UK: ACM, 2018, pp. 387–395. [Online]. Available: http://arxiv.org/abs/1802.04431%0Ahttp://dx.doi.org/10.1145/3219819.3219845

[19] M. S. Elsayed, N. A. Le-Khac, S. Dev, and A. D. Jurcut, “Detecting abnormal traffic in large-scale networks,” in *International Symposium on Networks, Computers and Communications (ISNCC).* Montreal, QC, Canada: IEEE, 2020, pp. 1–7.

[20] M. Said Elsayed, N.-A. Le-Khac, S. Dev, and A. D. Jurcut, “Network Anomaly Detection Using LSTM Based Autoencoder,” in *Q2SWinet ’20.* Alicante, Spain: ACM, 2020, pp. 37–45.

[21] H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng, J. Chen, Z. Wang, and H. Qiao, “Unsupervised Anomaly Detection via Variational Auto-Encoder for Seasonal KPIs in Web Applications,” in *The Web Conference 2018 - Proceedings of the World Wide Web Conference, WWW 2018.* Lyon, France: ACM, 2018, pp. 187–196.

[22] R. Xu, Y. Cheng, Z. Liu, Y. Xie, and Y. Yang, “Improved Long Short-Term Memory based anomaly detection with concept drift adaptive method for supporting IoT services,” *Future Generation Computer Systems*, vol. 112, pp. 228–242, 2020.

[23] N. Ding, H. X. Ma, H. Gao, Y. H. Ma, and G. Z. Tan, “Real-time anomaly detection based on long short-Term memory and Gaussian Mixture Model,” *Computers and Electrical Engineering*, vol. 79, 2019.

[24] M. Braei and S. Wagner, “Anomaly Detection in Univariate Time-series: A Survey on the State-of-the-Art,” pp. 1–39, 4 2020. [Online]. Available: http://arxiv.org/abs/2004.00433

[25] H. Ren, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong, and Q. Zhang, “Time-series anomaly detection service at microsoft,” in *KDD ’19: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage, AK, USA, 2019, pp. 3009–3017.

[26] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, “A review on outlier/anomaly detection in time series data,” *arXiv*, pp. 1–32, 2020.

[27] Z. Hasani, “Robust anomaly detection algorithms for real-time big data: Comparison of algorithms,” in *2017 6th Mediterranean Conference on Embedded Computing (MECO)*. Bar, Montenegro: IEEE, 2017, pp. 1–6.

[28] T. Y. Kim and S. B. Cho, “Web traffic anomaly detection using C-LSTM neural networks,” *Expert Systems with Applications*, vol. 106, pp. 66–76, 2018.

[29] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, “Unsupervised real-time anomaly detection for streaming data,” *Neurocomputing*, vol. 262, pp. 134–147, 2017.

[30] J. Wu, W. Zeng, and F. Yan, “Hierarchical Temporal Memory method for time-series-based anomaly detection,” *Neurocomputing*, vol. 273, pp. 535–546, 2018.

[31] S. Zhai, Y. Cheng, W. Lu, and Z. Zhang, “Deep Structured Energy Based Models for Anomaly Detection,” in *Proceedings of the 33nd International Conference on Machine Learning*, New York City, USA, 2016, pp. 1110–1119. [Online]. Available: http://arxiv.org/abs/1605.07717

[32] A. Borghesi, A. Bartolini, M. Lombardi, M. Milano, and L. Benini, “Anomaly Detection using Autoencoders in High Performance Computing Systems,” in *Proceedings of the AAAI Conference on Artificial Intelligence*. California, USA: AAAI Press, 2019, pp. 9428–9433. [Online]. Available: http://arxiv.org/abs/1811.05269

[33] D. Lee, “Anomaly detection in multivariate non-stationary time series for automatic DBMS diagnosis,” in *Proceedings - 16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017*, vol. 2018-Janua, 2018, pp. 412–419.

[34] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, “Towards K-means-friendly Spaces: Simultaneous Deep Learning and Clustering,” in *ICML’17: Proceedings of the 34th International Conference on Machine Learning*. Sydney Australia: OpenReview.net, 2017, p. 3861–3870. [Online]. Available: http://arxiv.org/abs/1610.04794

[35] T. Amarbayasgalan, B. Jargalsaikhan, and K. Ryu, “Unsupervised Novelty Detection Using Deep Autoencoders with Density Based Clustering,” *Applied Sciences*, vol. 8, no. 1468, pp. 1–18, 2018.

[36] Y. Zhou, R. Arghandeh, and C. J. Spanos, “Online learning of Contextual Hidden Markov Models for temporal-spatial data analysis,” in *2016 IEEE 55th Conference on Decision and Control, CDC 2016*. Las Vegas, USA: IEEE, 2016, pp. 6335–6341.

[37] S. Selim, M. Hashem, and T. M. Nazmy, "Hybrid Multi-level Intrusion Detection System," *International Journal of Computer Science and Information Security*, vol. 9, no. 5, pp. 23–29, 2011.

[38] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Computer Networks*, vol. 174, 2020.

[39] S. W. Yahaya, A. Lotfi, and M. Mahmud, "A Consensus Novelty Detection Ensemble Approach for Anomaly Detection in Activities of Daily Living," *Applied Soft Computing Journal*, vol. 83, 2019.

[40] F. T. Chan, Z. X. Wang, S. Patnaik, M. K. Tiwari, X. P. Wang, and J. H. Ruan, "Ensemble-learning based neural networks for novelty detection in multi-class systems," *Applied Soft Computing Journal*, vol. 93, 2020.

[41] D.-i. Curiac and C. Volosencu, "Expert Systems with Applications Ensemble based sensing anomaly detection in wireless sensor networks," *Expert Systems With Applications*, vol. 39, no. 10, pp. 9087–9096, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2012.02.036

[42] B. Krawczyk, M. Woźniak, and B. Cyganek, "Clustering-based ensembles for one-class classification," *Information Sciences*, vol. 264, pp. 182–195, 2014.

[43] D. Chakraborty, V. Narayanan, and A. Ghosh, "Integration of deep feature extraction and ensemble learning for outlier detection," *Pattern Recognition*, vol. 89, pp. 161–171, 2019.

[44] L. Sivaprakasam, "a Study on Anomaly Detection Algorithms," *Journal of Applied Logic*, vol. 4, no. 9, pp. 1–13, 2015.

[45] J. Chen, S. Sathe, C. Aggarwal, and D. Turaga, "Outlier detection with autoencoder ensembles," in *Proceedings of the 17th SIAM International Conference on Data Mining, SDM 2017*. Houston, Texas, USA: SIAM, 2017, pp. 90–98.

[46] T. Kieu, B. Yang, C. Guo, and C. S. Jensen, "Outlier detection for time series with recurrent autoencoder ensembles," in *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2019-Augus, 2019, pp. 2725–2732.

[47] W. Zhang, D. Yang, S. Zhang, J. H. Ablanedo-Rosas, X. Wu, and Y. Lou, "A novel multi-stage ensemble model with enhanced outlier adaptation for credit scoring," *Expert Systems with Applications*, vol. 165, p. 113872, 2021. [Online]. Available: https://doi.org/10.1016/j.eswa.2020.113872

[48] J. Zhang, Z. Li, K. Nai, Y. Gu, and A. Sallam, "DELR: A double-level ensemble learning method for unsupervised anomaly detection," *Knowledge-Based Systems*, vol. 181, 2019.

[49] B. Wang and Z. Mao, "A dynamic ensemble outlier detection model based on an adaptive k-nearest neighbor rule," *Information Fusion*, vol. 63, pp. 30–40, 2020.

[50] B. Wang, Z. Mao, and K. Huang, "Detecting outliers for complex nonlinear systems with dynamic ensemble learning," *Chaos, Solitons and Fractals*, vol. 121, pp. 98–107, 2019.

[51] B. Krawczyk and M. Woźniak, "Dynamic classifier selection for one-class classification," *Knowledge-Based Systems*, vol. 107, pp. 43–53, 2016.

[52] S. Lu, X. Wei, B. Rao, B. Tak, L. Wang, and L. Wang, “LADRA: Log-based abnormal task detection and root-cause analysis in big data processing with Spark,” *Future Generation Computer Systems*, vol. 95, pp. 392–403, 2019.

[53] J. Wang, Z. Yang, J. Su, Y. Zhao, S. Gao, X. Pang, and D. Zhou, “Root-cause analysis of occurring alarms in thermal power plants based on Bayesian networks,” *International Journal of Electrical Power and Energy Systems*, vol. 103, pp. 67–74, 2018.

[54] D. V. Groenewald and C. Aldrich, “Root cause analysis of process fault conditions on an industrial concentrator circuit by use of causality maps and extreme learning machines,” *Minerals Engineering*, vol. 74, pp. 30–40, 2015.

[55] H. Zhou, Y. Li, H. Yang, J. Jia, and W. Li, “BigRoots: An Effective Approach for Root-Cause Analysis of Stragglers in Big Data System,” *IEEE Access*, vol. 6, pp. 41 966–41 977, 2018.

[56] R. C. Aster, B. Borchers, and C. H. Thurber, *Parameter estimation and inverse problems.* Elsevier B.V., 2018.

[57] C. Liu and J. Xiong, “Chapter 1 Forward and inverse problems in geophysics,” in *Methods in Geochemistry and Geophysics*, 2002, vol. 36, ch. 1, pp. 3–28.

[58] D. F. Andrews, “Plots of High-Dimensional Data,” *Biometrics*, vol. 28, no. 1, p. 125, 3 1972. [Online]. Available: https://www.jstor.org/stable/2528964?origin=crossref

[59] R. E. Moustafa, “Andrews curves,” *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 3, no. 4, pp. 373–382, 2011.

[60] S. Mascaro, A. E. Nicholso, and K. B. Korb, “Anomaly detection in vessel tracks using Bayesian networks,” *International Journal of Approximate Reasoning*, vol. 55, no. 1, pp. 84–98, 1 2014. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0888613X13000728

[61] G. Zaccone, *Getting Started with TensorFlow.* Birmingham, UK: Packt Publishing, 2016.

[62] A. Ganti, “Weighted Average,” 2019. [Online]. Available: https://www.investopedia.com/terms/w/weightedaverage.asp

[63] A. Singh, “Anomaly detection,” 2018. [Online]. Available: https://www.kaggle.com/arshahuja/anomaly-detection

[64] Anush, “Network Anomaly Detection,” 2018. [Online]. Available: https://www.kaggle.com/anushonkar/network-anamoly-detection

[65] Ask9, “Detecting Anomalies in Wafer Manufacturing,” 2020. [Online]. Available: https://www.kaggle.com/arbazkhan971/anomaly-detection

[66] E. Frazelle, *Supply Chain: The Logistics of Supply Chain Management.* the United States of America: McGraw-Hill, 2002. [Online]. Available: http://www.lavoisier.fr/livre/notice.asp?id=O3LW2LA2RR3OWA

[67] B. Wang and Z. Mao, “Outlier detection based on a dynamic ensemble model: Applied to process monitoring,” *Information Fusion*, vol. 51, pp. 244–258, 2019.

[68] M. Christopher, *Logistics and Supply Chain Management*, 3rd ed. New Jersey, US: Pearson Education,Inc., 2005, vol. 41.

[69] H. Akaike, "Information Theory and an Extension of the Maximum Likelihood Principle," in *2nd International Symposium on Information Theory*, 1998, pp. 199–213.

[70] G. Schwarz, "Estimating the Dimension of a Model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[71] A. Lokrantz, E. Gustavsson, and M. Jirstrand, "Root cause analysis of failures and quality deviations in manufacturing using machine learning," *Procedia CIRP*, vol. 72, pp. 1057–1062, 2018.

[72] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. New Jersey, US: Pearson Education,Inc., 2010.

[73] L. L. Cruciol, A. C. De Arruda, L. Weigang, L. Li, and A. M. Crespo, "Reward functions for learning to control in air traffic flow management," *Transportation Research Part C: Emerging Technologies*, vol. 35, pp. 141–155, 2013.

[74] Richard Koch, *The 80/20 Principle.* London, UK: NICHOLAS BREALEY PUBLISHING, 1998.

[75] D. Avgerinou, "ABC Analysis of Active Inventory," 2018. [Online]. Available: https://www.kaggle.com/danavg/abc-analysis-of-active-inventory

[76] H. Kartal, A. Oztekin, A. Gunasekaran, and F. Cebi, "An integrated decision analytic framework of machine learning with multi-criteria decision making for multi-attribute inventory classification," *Computers and Industrial Engineering*, vol. 101, pp. 599–613, 2016.

[77] F. Lolli, E. Balugani, A. Ishizaka, R. Gamberini, B. Rimini, and A. Regattieri, "Machine learning for multi-criteria inventory classification applied to intermittent demand," *Production Planning and Control*, vol. 30, no. 1, pp. 76–89, 2019.

[78] 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 31*, vol. 1. Montréal, Canada: Curran Associates, Inc., 2018, pp. 9839–9849.

[79] W. Kool, H. Van Hoof, and M. Welling, "Attention, learn to solve routing problems!" in *7th International Conference on Learning Representations, ICLR 2019*, 2019, pp. 1–25. [Online]. Available: http://arxiv.org/abs/1803.08475

[80] R. S. Sutton and Andrew G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. London, UK: MIT Press., 2017.

[81] L. D. Giovanni, N. Gastaldon, M. Losego, and F. Sottovia, "Algorithms for a Vehicle Routing Tool Supporting Express Freight Delivery in Small Trucking Companies," in *Transportation Research Procedia*, vol. 30, 2018, pp. 197–206.

[82] Google Developers, "Google OR-Tools," 2019. [Online]. Available: https://developers.google.com/optimization

[83] D. L. Poole and A. K. Mackworth, *Artificial Intelligence: Foundations of Computational Agents*, 2nd ed. Cambridge, UK: Cambridge University Press., 2017.

[84] M. E. Taylor, B. Kulis, and F. Sha, “Metric learning for reinforcement learning agents,” in *10th International Conference on Autonomous Agents and Multiagent Systems 2011, AAMAS 2011*, vol. 2, 2011, pp. 729–736.

[85] J. Insa-Cabrera, D. L. Dowe, and J. Hernández-Orallo, “Evaluating a reinforcement learning algorithm with a general intelligence test,” *Lecture Notes in Computer Science*, vol. 7023 LNAI, pp. 1–11, 2011.

[86] S. Kotz, N. L. Johnson, and C. B. Read, *Encyclopedia of Statistical Sciences*, 1st ed. Hoboken, New Jersey, United States: Wiley, 2006.

[87] B. S. Everitt and A. Skrondal, *The Cambridge Dictionary of Statistics*, 4th ed. Cambridge, UK: Cambridge University Press., 2010.

[88] G. Musolino, A. Polimeni, and A. Vitetta, “Freight vehicle routing with reliable link travel times: a method based on network fundamental diagram,” *Transportation Letters*, vol. 10, no. 3, pp. 159–171, 2018.

PUBLICATIONS

Main Publications

International Journals

1. Thananut Phiboonbanakit, Van-Nam Huynh, Teerayut Horanont, and Thepchai Supnithi, “Unsupervised hybrid anomaly detection model for logistics fleet management systems,” *IET Intelligent Transport Systems*, vol. 13, no. 11, pp. 1636-1648, 2019.

2. Thananut Phiboonbanakit, Teerayut Horanont, Van-Nam Huynh, and Thepchai Supnithi, “A Hybrid Reinforcement Learning-based Model for the Vehicle Routing Problem in Transportation Logistics,” *IEEE Access*, Revise and resubmit, 24 pages.

3. Thananut Phiboonbanakit, Teerayut Horanont, Thepchai Supnithi, and Van-Nam Huynh, “A Novel Reinforcement Learning and Behavior Analysis Model for solving the Vehicle Routing Problem in the Uncertain Transportation Environment,”*Engineering Applications of Artificial Intelligence*, To be submitted, 46 pages.

International Conference Proceedings

1. Thananut Phiboonbanakit, Thepchai Supnithi, Teerayut Horanont, and Van-Nam Huynh, “Knowledge-based learning for solving vehicle routing problem,” in *UbiComp/ISWC 2018 - Adjunct Proceedings of the 2018 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2018 ACM International Symposium on Wearable Computers*, Singapore, pp. 1103-1111, 2018.

2. Thananut Phiboonbanakit, Teerayut Horanont, Van-Nam Huynh, and Thepchai Supnithi, “Using Bayesian Network to Model an Anomaly in Fleet Management Operation,” in *26th ITS World Congress, Singapore*, Singapore, pp. 21-25, 2019.

3. Thananut Phiboonbanakit, Van-Nam Huynh, Teerayut Horanont, and Thepchai Supnithi, “Detecting abnormal behavior in the transportation planning using long short term memories and a contextualized dynamic threshold,” in *UbiComp/ISWC 2019 - Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, London UK, pp. 996-1007, 2019.

Other Publications

1. Teerayut Horanont, Thananut Phiboonbanakit, and Santi Phithakkitnukoon, “Resembling Population Density Distribution with Massive Mobile Phone Data,” *Data Science Journal*, vol. 17, no. 9, 2018.

2. Thananut Phiboonbanakit and Teerayut Horanont, “Analyzing Bangkok City Taxi Ride: Reforming Fares for Profit Sustainability using Big Data Driven Model,” *Journal of Big Data*, vol. 8, no. 7, 2021.