AN INTEGRATED APPROACH FOR SUPPLY CHAIN OPTIMIZATION USING MACHINE LEARNING TECHNIQUES

A thesis submitted in partial fulfillment of the requirements for the degree of

MASTER of SCIENCE in Mechanical Engineering

By

S. M. Fahim Faisal

20MME020F

Supervised by

Dr. Sajal Chandra Banik

Professor, Dept. of ME, CUET

Department of Mechanical Engineering

CHITTAGONG UNIVERSITY OF ENGINEERING AND TECHNOLOGY

August, 2024

APPROVAL

The thesis titled "**An Integrated Approach for Supply Chain Optimization Using Machine Learning Techniques**" submitted by S. M. Fahim Faisal (ID:20MME020F, Session: 2020-2021) has been accepted as satisfactory by the Dissertation Committee for the partial fulfillment of the requirement for the degree of Master of Science (M.Sc.) in Mechanical Engineering, dated on 14th August,2024.

Dissertation Committee

_____	
Dr. Sajal Chandra Banik	Chairman
Professor,	(Supervisor)
Department of Mechanical Engineering, CUET.	
_____	
Dr. Mohammad Mizanur Rahman	Member
Head,	(Ex Officio)
Department of Mechanical Engineering, CUET.	
_____	
Dr. Jamal Uddin Ahamed	Member
Professor,	
Department of Mechanical Engineering, CUET.	
_____	
Dr. Kazi Afzalur Rahman	Member
Professor,	
Department of Mechanical Engineering, CUET.	
_____	
Dr. Md. Helal-An-Nahiyan	
Professor,	Member
Department of Mechanical Engineering, KUET.	External

Declaration

I hereby declare that the work in this thesis has not been previously submitted to meet the requirements for an award at this or any other higher education institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is cited. Furthermore, the Thesis compiles with PLAGIARISM and ACADEMIC INTEGRITY regulation of CUET.

S. M. Fahim Faisal
20MME020F
Department of Mechanical Engineering
Chittagong University of Engineering & Technology (CUET)

List of Publications

S. M. Fahim Faisal, Sajal Chandra Banik, Tasnuva Jahan Nuva and Nusrat Sultana, “An Analytics Model of Supply Chain Risk Optimization using Machine Learning” 2nd International Conference on Mechanical, Manufacturing and Process Engineering (ICMMPE – 2024), Paper ID: 78, 2024.

Approval by the Supervisor

This is to certify that S. M. Fahim Faisal (ID:20MME020F) has carried out this work under my supervision and that he has fulfilled the relevant Academic Ordinance of the Chittagong University of Engineering and Technology so that she is qualified to submit the following thesis in application for the degree of MASTER of SCIENCE in Mechanical Engineering. Furthermore, the Thesis compiles with PLAGIARISM and ACADEMIC INTEGRITY regulation of CUET.

Dr. Sajal Chandra Banik

Professor

Department of Mechanical Engineering

Chittagong University of Engineering & Technology

Abstract

In the modern world, supply chains completely rely on data to function properly under risk and uncertainty. Supply chain risk optimization is a process that involves identifying, assessing, and managing potential risks within a supply chain network to minimize disruptions. A machine learning analytics model of supply chain risk optimization uses data analytics and machine learning algorithms to understand and assess supply chain risks. Out of many types of risks involved in the supply chain, late delivery risk is the most common, and a lot of attention has been paid by researchers in this regard. The work presented in this thesis utilizes the DataCo Supply Chain dataset. Out of many risks, late delivery and fraud detection are considered in this research work to optimize the risks associated with the supply chain. In total, 15 different machine learning classification algorithms along with two hybrid algorithms are implemented and compared. The better performing hybridized classification algorithm is created in this paper combining the Multi-Layer Perceptron Classifier, Random Forest, and Extra Trees Classifier is put to the test. The hybrid algorithm outperforms all the algorithms and shows an accuracy of 99.45% and 99.15% for late delivery status prediction and fraud detection respectively. In the later part of the thesis, Deep Reinforcement Learning algorithms have been implemented for supply chain pricing policy optimization. The unique factor is that real-time data from an online marketplace in Bangladesh is used in this regard. Deep Q Network and State-Action-Reward-State-Action algorithm have been used, performance-wise Deep Q Network algorithm performed better and it achieved 19% more profit than constant price optimization. The overall work done in this thesis provides a solid foundation of integrated supply chain optimization by which supply chain managers can act proactively and can get benefit.

বিমূর্ত

আধুনিক বিশ্বে, সাপ্লাই চেইনগুলি ঝুঁকি এবং অনিশ্চয়তার মধ্যে ভালভাবে চালানোর জন্য সম্পূর্ণরূপে ডেটার উপর নির্ভর করে। সাপ্লাই চেইন রিস্ক অপ্টিমাইজেশান হল এমন একটি প্রক্রিয়া যাতে বিঘ্ন কমানোর জন্য সাপ্লাই চেইন নেটওয়ার্কের মধ্যে সম্ভাব্য ঝুঁকি চিহ্নিত করা, মূল্যায়ন করা এবং পরিচালনা করা হয়। সাপ্লাই চেইন রিস্ক অপ্টিমাইজেশানের একটি মেশিন লার্নিং অ্যানালিটিক্স মডেল সাপ্লাই চেইন ঝুঁকি বুঝতে ও মূল্যায়ন করতে ডেটা অ্যানালিটিক্স এবং মেশিন লার্নিং অ্যালগরিদম ব্যবহার করে। সাপ্লাই চেইন জড়িত অনেক ধরনের ঝুঁকির মধ্যে, বিলম্বে ডেলিভারি ঝুঁকি সবচেয়ে সাধারণ, এবং এই বিষয়ে গবেষকরা অনেক মনোযোগ দিয়েছেন। এই থিসিসে উপস্থাপিত কাজ ডেটাকো সাপ্লাই চেইন ডেটাসেট ব্যবহার করে করা হয়েছে। অনেক ঝুঁকির মধ্যে, দেরী ডেলিভারি এবং জালিয়াতি সনাক্তকরণ এই গবেষণা কাজে বিবেচনা করা হয় সাপ্লাই চেইন সাথে সম্পর্কিত ঝুঁকিগুলিকে অপ্টিমাইজ করার জন্য। মোট, দুটি হাইব্রিড অ্যালগরিদমের সাথে ১৫টি ভিন্ন মেশিন লার্নিং ক্লাসিফিকেশন অ্যালগরিদম প্রয়োগ করা হয় এবং তুলনা করা হয়। মাল্টি-লেয়ার পারসেপ্ট্রন ক্লাসিফায়ার, রান্ডম ফরেস্ট এবং এক্সট্রা ট্রিস ক্লাসিফায়ারের সমন্বয়ে এই পেপারে একটি সেরা পারফর্মিং অ্যালগরিদম হাইব্রিডাইজড ক্লাসিফিকেশন অ্যালগরিদম তৈরি করা হয়েছে। হাইব্রিড অ্যালগরিদম সমস্ত অ্যালগরিদমকে ছাড়িয়ে যায় এবং দেরী ডেলিভারির অবস্থার পূর্বাভাস এবং জালিয়াতি সনাক্তকরণের জন্য যথাক্রমে ৯৯.৪৫% এবং ৯৯.১৫% সঠিকতা দেখায়৷ থিসিসের পরবর্তী অংশে, সাপ্লাই চেইন প্রাইসিং পলিসি অপ্টিমাইজেশানের জন্য ডিপ রিইনফোর্সমেন্ট লার্নিং অ্যালগরিদম প্রয়োগ করা হয়েছে। অনন্য ফ্যাক্টর হল যে বাংলাদেশের একটি অনলাইন মার্কেটপ্লেস থেকে রিয়েল-টাইম ডেটা এই বিষয়ে ব্যবহার করা হয়। ডিপ কিউ নেটওয়ার্ক এবং স্টেট-অ্যাকশন-রিওয়ার্ড-স্টেট-অ্যাকশন অ্যালগরিদম ব্যবহার করা হয়েছে, পারফরম্যান্স অনুসারে ডিপ কিউ নেটওয়ার্ক অ্যালগরিদম ভাল পারফর্ম করেছে এবং এটি ধ্রুবক মূল্য অপ্টিমাইজেশানের তুলনায় ১৯% বেশি মুনাফা অর্জন করেছে । এই থিসিসে সম্পাদিত সামগ্রিক কাজ সমন্বিত সাপ্লাই চেইন অপ্টিমাইজেশানের একটি শক্ত ভিত্তি প্রদান করে যার দ্বারা সাপ্লাই চেইন ম্যানেজাররা সক্রিয়ভাবে কাজ করতে পারে এবং সুবিধা পেতে পারে।

Acknowledgment

Above all, I would like to express my gratitude to my supervisor, **Prof. Dr. Sajal Chandra Banik,** Professor, Dept. of ME, CUET for accepting me into his research group and also express my heartfelt thanks to him for his guidance, encouragement, and continuous support during my graduate studies. His enthusiasm for teaching and research offered me challenging opportunities to expand my scientific knowledge and my growing interest in the world of Industrial Engineering specifically in supply chain management.

TABLE OF CONTENTS

APPROVAL ii
Declaration iii
List of Publications iv
Approval by the Supervisor v
Abstract vi
বিমূর্ত vii
Acknowledgment viii
LIST OF FIGURES xii
LIST OF TABLES xiv
LIST OF SYMBOLS AND ABBREVIATIONS xv
Chapter 01 1
Introduction 1
1.1 Background and motivation 1
1.2 Thesis objectives 4
1.3 Context of supply chain risk optimization 4
1.4 Context of supply chain pricing policy optimization 4
1.5 Hybrid model 7
1.6 Conclusion 7
Chapter 02 9
Literature Review 9
2.1 Introduction 9
2.2 Previous study regarding ML 9
2.3 Previous study regarding DRL 13
2.4 Conclusion 21
Chapter 03 22
Methodology 22
3.1 Introduction 22
3.2 Supply chain risk optimization using machine learning 22
3.2.1 Data Collection 22
3.2.2 Structured data formation 23

3.2.3 Data analysis and preprocessing 24
3.2.4 Machine learning algorithms used 29
3.3 Supply chain pricing policy optimization using deep reinforcement learning . 30
3.3.1 Environment design 30
3.3.2 Real-time online based marketplace data 31
3.3.3 Implementation of price optimization algorithms 33
3.3.3.1 Constant price optimization 33
3.3.3.2 Greedy dynamic price optimization 34
3.3.3.3 Deep reinforcement learning approaches 34
3.3.3.3.1 Deep Q Network (DQN) implementation 34
3.3.3.3.2 Neural network architecture of DQN 35
3.3.3.3.2 SARSA implementation 41
3.4 Conclusion 45
Chapter 04 46
Results 46
4.1 Introduction 46
4.2 Results obtained for supply chain risk optimization problem 46
4.2.1 Confusion matrix 46
4.2.1.1 Logistic Regression 47
4.2.1.2 Gaussian Naive Bayes 48
4.2.1.3 Support Vector Machine 49
4.2.1.4 K-nearest Neighbors 49
4.2.1.5 Linear Discriminant Analysis 50
4.2.1.6 Random Forest Classifier 50
4.2.1.7 Extra Trees Classifier 51
4.2.1.8 Extreme Gradient Boosting (XGB) Classifier 51
4.2.1.9 Decision Tree Classifier 52
4.2.1.10 Ada Boost Classifier 52
4.2.1.11 Histogram Gradient Boosting Classifier 53
4.2.1.12 Light GBM Classifier 53
4.2.1.13 Multi-layer Perceptron (MLP) Classifier 54
4.2.1.14 Hybrid Model 2 54

4.2.2 Comparative Performance Analysis 55
4.3 Results obtained for supply chain pricing optimization problem 58
4.3.1 DQN algorithm results 58
4.3.2 SARSA algorithm results 60
4.4 Conclusion 62
Chapter 05 63
Conclusion 63
5.1 Introduction 63
5.2 Summary of findings 63
5.3 Future research directions 64
5.4 Concluding remarks 65
References 66
Appendix-A 75

LIST OF FIGURES

Figure 1.1	Phases of supply chain optimization	3
Figure 3.1	Framework of model (Analytics model of supply chain risk optimization)	25
Figure 3.2	Relation between product prize and sales per customer	26
Figure 3.3	Visualization of payment method used	27
Figure 3.4	Pie chart representing the regions with fraud	28
Figure 3.5	Pie chart representing customer's segmentation	29
Figure 3.6	Flow chart of the implemented DQN algorithm	40
Figure 3.7	Flow chart of the implemented SARSA algorithm	44
Figure 4.1	Confusion matrix of Logistic Regression model	48
Figure 4.2	Confusion matrix of Gaussian Naïve Bayes model	48
Figure 4.3	Confusion matrix of Support Vector Machine model	49
Figure 4.4	Confusion matrix of K-nearest Neighbors model	49
Figure 4.5	Confusion matrix of Linear Discriminant Analysis model	50
Figure 4.6	Confusion matrix of Random Forest Classifier model	50
Figure 4.7	Confusion matrix of Extra Trees Classifier model	51
Figure 4.8	Confusion matrix of Extreme Gradient Boosting Classifier model	51
Figure 4.9	Confusion matrix of Decision Tree Classifier model	52
Figure 4.10	Confusion matrix of Ada Boost Classifier model	52

Figure 4.11	Confusion matrix of Histogram Gradient Boosting Classifier model	53
Figure 4.12	Confusion matrix of Light GBM Classifier model	53
Figure 4.13	Confusion matrix of Multi-layer Perceptron (MLP) Classifier model	54
Figure 4.14	Confusion matrix of Hybrid model 2	54
Figure 4.15	DQN algorithm implementation	59
Figure 4.16	SARSA algorithm implementation	61

LIST OF TABLES

Table 4.1	Fraud detection and prediction	55
Table 4.2	Accuracy comparison of fraud detection with previous studies	56
Table 4.3	Late delivery status prediction	57
Table 4.4	Best profit results for DQN algorithm	59
Table 4.5	Best profit results for SARSA algorithm	62
Table 4.6	Comparison of best profit results	62
Appendix A	Information from ABC Company	75

LIST OF SYMBOLS AND ABBREVIATIONS

ABBREVIATIONS

ML	Machine Learning
DRL	Deep Reinforcement Learning
DQN	Deep Q network
SARSA	State-Action-Reward-State-Action
SCM	Supply Chain Management
GBM	Gradient Boosting Method
ReLU	Rectified Linear Unit

Chapter 01

Introduction

1.1 Background and motivation

In today's world, when it comes to effectively regulating their supply chains, organizations confront many obstacles. Factors such as globalization, complex logistics networks, demand variability, and uncertain market conditions have necessitated the development of advanced techniques to optimize supply chain operations. This research delves into the world of machine learning algorithms and their application in optimizing supply chain processes.

Supply chain management is the coordination and integration of various processes involved in the production, sourcing and distribution of goods and services [1]. A well-optimized supply chain minimizes costs, improves customer service, enhances operational efficiency, and ultimately contributes to the overall success of a company. However, achieving these objectives is no easy task, given the complexities and uncertainties involved. Traditional supply chain optimization approaches heavily rely on mathematical models and operations research techniques [2]. While these methods have proven effective, they struggle to manage the intricate and unpredictable structure of present supply chains. Herein lies the application of machine learning algorithms, offering new possibilities for enhanced decision-making and improved optimization outcomes [3].

As a branch of artificial intelligence, machine learning makes use of algorithms to evaluate and understand vast amounts of data, spot trends, forecast future events, or initiate action without explicit programming. In light of improvements in computing. capacity and the abundance of extensive datasets, machine learning algorithms have achieved significant popularity and proven their effectiveness across various domains. Supervised learning, such as linear regression, decision trees, and support vector

machines, can be used to model and predict demand, monitor inventory levels, and plan supply chain operations [4]. Unsupervised learning algorithms, such as dimensionality reduction techniques and clustering, allow the detection of hidden patterns and groupings within supply chain data. Reinforcement learning algorithms offer a way to optimize sequential decision-making processes by allowing an agent to acquire knowledge through trial and error [5].

Furthermore, the integration and coordination of demand planning, production, distribution, and purchasing constitute the supply chain. Making judgments at the strategic, tactical, and operational levels is necessary for all the action plans. Moreover, optimization models are being created to run these supply chain operations smoothly. Supply chain management (SCM) is hindered by an imbalance in knowledge and uncertainty. SCM decision-making relies on immutable criteria, which are now often impacted by information barriers [6]. However, with more data available than ever before, it is inefficient or impossible to analyze this data using conventional techniques, leading to the emergence of new techniques and applications. Machine learning (ML), which shades light on the creation and use of self-learning algorithms, is one of these approaches that can be used in analyzing supply chains [7], [8].

Additionally, society is entering a period known as "fourth industrial revolution" [9], characterized through the advancement of information technology, robotics, communication systems, and artificial intelligence (AI). AI has got the unique feature that machines acquire intelligence when making judgments rather than relying on the human brain. One of these methods is machine learning (ML), which is concerned with the creation and use of computer algorithms that "learn" from experience [10]. In several decision-making fields, such as cancer diagnosis and prognosis [11], drug discovery [12], and genetics and genomics [13], the literature showed that computers could provide more accurate findings and analysis than humans. As the need to make decisions under uncertainty is an important issue in supply chains [14], ML can be extensively used. For SCM, ML represents a true asset as ML is better than traditional approaches in describing non-linear relationships because its training model can illustrate how the result varies with the input with accuracy.

Supply chain optimization is the process of maximizing the efficiency and effectiveness of a manufacturing process and its distribution system. [15]. Optimization of a supply chain refers to the practice of modifying its operations to provide optimal efficiency. Supply chain optimization happens in three phases: the design phase, the planning phase, and the execution phase [16]. The three phases of optimization are now described below in Figure 1.1.

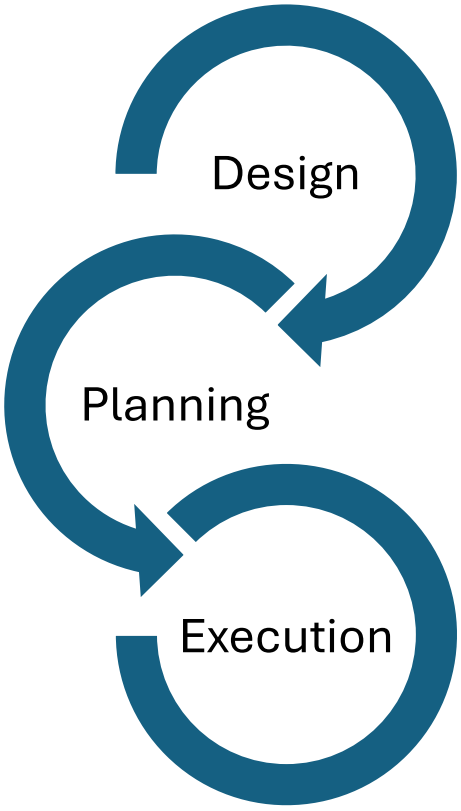

Figure 1.1: Phases of supply chain optimization

In the design phase, pricing policy optimization, supplier selection, and other related works are done. In the planning phase, the production plan, risk identification, and inventory planning are conducted. Finally, in the execution phase, all the strategies are implemented using the optimal strategy, and feedback is obtained for planning the next stage.

1.2 Thesis objectives

The main objectives of this thesis are as follows,

- To analyze the performance of various machine learning algorithms in supply chain risk optimization.
- To develop a hybrid model and compare it against traditional machine learning algorithms.
- To implement deep reinforcement learning algorithm in real-world supply chain pricing policy optimization.

1.3 Context of supply chain risk optimization

In this research, supply chain risk optimization is done by predicting late delivery risk and detecting fraud transactions. Late delivery prediction in the supply chain involves using data analytics and predictive modeling techniques to forecast the likelihood of a shipment being delivered late [17]. A variety of data points, including order quantities, lead times, transit times, carrier performance, and transportation schedules, may be taken into account by predictive models. Prediction accuracy can be increased over time by using machine learning algorithms to continuously examine and learn from new data. Utilizing machine learning approaches for supply chain fraud detection is crucial to minimizing financial losses and maintaining transaction integrity [18]. Machine learning algorithms are capable of analyzing massive volumes of data while identifying patterns and anomalies that may point to fraud.

1.4 Context of supply chain pricing policy optimization

One of the contributing factors of this research is its examination of the application of deep reinforcement learning (DRL) algorithms to optimize the pricing policy of a supply chain within an online marketplace in Bangladesh, with a particular emphasis on the pricing of T-shirts. Pricing is an important element in the supply chain management of any retail business as it directly influences consumer demand, inventory levels, and overall profitability. Traditional pricing strategies often rely on static models or heuristic approaches that may not fully capture the complexities and

dynamism of the market. In contrast, DRL offers a robust framework for engaging with the environment to discover the best pricing rules. and continuously adapting to new data [19].

Moreover, the unique aspect of this research is its reliance on real-time data rather than simulated environments or Markov Decision Processes (MDPs). This approach ensures that the findings are grounded in actual market conditions, enhancing the practical applicability of the results [20]. By leveraging real-time sales data from an online marketplace in Bangladesh, the study targets to provide practical advice that can be directly implemented to improve pricing strategies. The primary objective of this work is to develop and evaluate DRL-based algorithms for optimizing the pricing policy of a specific product—a T-shirt—in real time. The algorithms considered include Deep Q-Network (DQN) and State-Action-Reward-State-Action (SARSA). Each of these algorithms offers distinct advantages and complexities, making them suitable for different aspects of the pricing optimization problem [21] – [23]. By focusing on a single product, the study ensures a detailed and nuanced analysis of the pricing dynamics, while also providing a scalable framework that can be extended to other products in the future. The use of real-time data not only enhances the relevance of the study but also poses unique challenges in terms of data processing and algorithmic adaptation, which are addressed in the methodology. The significance of this research lies in its ability to bridge the gap between theoretical advancements in DRL and practical applications in retail pricing. While previous works have demonstrated the efficacy of DRL in various domains, there is a paucity of research that applies these techniques to real-time pricing in online marketplaces [24].

Furthermore, this study contributes to the literature by providing empirical evidence of the effectiveness of DRL in a real-world setting. Moreover, the focus on the Bangladeshi market adds a unique dimension to the research. As an emerging economy with a rapidly growing e-commerce sector, Bangladesh presents a fertile ground for innovative pricing strategies. The insights gained from this study can inform policymakers and business leaders in similar markets, facilitating the application of advanced AI techniques in the management of supply chains [25].

The methodology involves the collection of real-time sales data from an online marketplace, preprocessing the data, and implementing the DRL algorithms to optimize the pricing policy. The performance of DQN and SARSA. A detailed description of the data sources, preprocessing techniques, and algorithmic implementation is provided in the subsequent sections. One of the main challenges dealt with in the presented study is the real-world features of the data. Traditional DRL applications often rely on simulated environments where the data is static or semi-static [26]. In contrast, this study deals with continuous streams of data that require dynamic adaptation and real-time decision-making. The algorithms are trained and evaluated in a manner that reflects these real-world conditions, ensuring that the results are both robust and practically relevant.

While a comprehensive literature review is beyond the scope of this introduction, it is important to situate this research within the broader context of existing studies. Early work by Sutton and Barto [19] laid the foundation for reinforcement learning, providing the theoretical underpinnings that have been built upon by subsequent researchers. Mnih et al. [21] introduced the DQN algorithm, demonstrating its potential in complex decision-making tasks. Recent studies have explored the application of DRL in various domains, including finance, healthcare, and robotics [20], [24]. However, there is limited research on its application to real-world pricing in supply chains. This research aims to fulfill this gap by providing empirical evidence from a real-world online marketplace.

Overall, this research makes several key contributions like by using real-time data, the study provides empirical validation of DRL algorithms in a real-world setting, demonstrating their practical applicability and the comparative analysis of DQN and SARSA that provides details regarding both the benefits and drawbacks of every algorithm. This research focuses on the Bangladeshi market providing valuable insights that can inform pricing strategies in similar emerging economies. The results section of this research presents the findings, which are then discussed in the context of existing literature.

1.5 Hybrid model

In this research, a robust hybrid Model is developed combining Multilayer Perceptron (MLP), Random Forest, and Extra Trees Classifier to tackle a challenging classification task. One other hybrid model is also developed which didn't generate better results. The MLP, a type of artificial neural network, captures complex, non-linear patterns within the data through its deep, fully connected layers. Meanwhile, the Random Forest and Extra Trees Classifiers, both ensemble learning methods, enhance the model's performance by aggregating the predictions of multiple decision trees to reduce overfitting and improve generalization. In the approach presented here, the MLP first processes the input data, leveraging its powerful feature extraction capabilities. The output from the MLP, along with the original input features, is then fed into the Random Forest and Extra Trees Classifiers. The final predictions are obtained by aggregating the results of these classifiers, ensuring a comprehensive analysis of the data. This hybrid approach excels due to its ability to harness the strengths of both neural networks and ensemble methods. The MLP provides a sophisticated understanding of intricate data patterns, while the Random Forest and Extra Trees Classifiers add robustness and stability to the predictions. This combination results in superior performance, evidenced by high accuracy, precision, recall, and F1-score metrics, indicating the model's strong predictive power and reliability. The synergy between these diverse algorithms makes sure that the hybrid model captures wide variety of data characteristics, leading to exceptional classification results.

1.6 Conclusion

In conclusion, the optimization of pricing policy using DRL offers a noteworthy advancement in supply chain management space. By utilizing real-world data and cutting-edge algorithms, this study aims to provide actionable insights that can enhance the profitability and competitiveness of online marketplaces. The unique focus on a single product within the Bangladeshi market adds a practical dimension to the research, ensuring that the findings are both relevant and impactful. The work in

this paper aims to do a comparative analysis of various machine learning models to better predict late delivery and fraud transactions, which ultimately results in the optimization of supply chain risks. One hybridized ML algorithm was also implemented in the later part, which produced good results. Supply chain managers can act proactively with this information and raise overall supply chain performance by taking care of possible problems before they arise.

The organization of this thesis is structured in a total of five chapters. The next chapter illustrates all the previous studies and literature regarding supply chain risk optimization and pricing policy optimization. Chapter 3 explains the detailed methodology of how the work is done. Chapter 4 represents all the results obtained from implementing various machine learning algorithms and also the hybrid model implementation. Also, the results of DRL algorithms are discussed in detail. The last chapter of this thesis throws light on the conclusion of the work and future implementations and research directions.

Chapter 02

Literature Review

2.1 Introduction

The initial phase of this research endeavor involves a comprehensive analysis of studies about the optimization of supply chain pricing policies utilizing Deep Reinforcement Learning and supply chain risk optimization using ML. This preliminary investigation aims to offer a thorough understanding of the methodologies employed by previous researchers. Thus, the identified limitations and gaps in previous research will play a vital role in shaping the objectives fulfillment and methodologies of this thesis.

2.2 Previous study regarding ML

This section presents all the previous literature regarding supply chain risk optimization using various machine learning algorithms. First, some previous studies regarding supply chain optimization are presented and then the endeavor is given particularly on risk optimization in the supply chain.

Kalaitzi et al. [27] analyzed the implementation of machine learning in supply chain optimization, emphasizing the significant impact of ML techniques on improving supply chain efficiency and performance. This study highlights the importance of adaptive strategies that incorporate resource-efficient practices and stakeholder collaboration to manage natural resource scarcity. The authors employed a mixed-method approach, integrating quantitative data analysis with qualitative case studies. Data was collected through surveys and interviews with supply chain managers from various industries. One limitation is the potential bias in self-reported data, which might affect the generalizability of the findings. Additionally, the study is context-specific and may not fully capture the diverse strategies applicable across different regions and industries.

Building on the discussion of various machine learning algorithms, Wong et al. [28] explored the versatility of these algorithms when tackling multiple aspects of supply chain operation, including supervised, unsupervised, and reinforcement learning. Their comprehensive literature review analyzes numerous studies that applied ML techniques to supply chain problems. A major limitation noted is the complexity and computational demands of implementing advanced ML algorithms, which may require significant investment in technology and expertise. Despite these challenges, the study concludes that ML offers substantial potential for improving supply chain efficiency and performance.

Following the exploration of different ML algorithms, Yang et al. [29] has presented an AI-based model that focuses on financial risk prevention in supply chains. Utilizing data mining and machine learning techniques, the model helps enterprises make informed decisions by analyzing existing financial indices. The chaotic grasshopper optimization algorithm (CGOA) and Slime Mould Algorithm (SMA) enhance the Support Vector Machine (SVM) classification process. Empirical results demonstrate the model's efficiency in predicting financial risks, aiding in the proactive management of supply chain operations. The proposed approach leverages machine learning to transform traditional supply chains into intelligent, adaptive systems capable of mitigating financial risks. Moreover, Nguyen and Nghiem [30] focus on predicting supply chain risks using Bayesian networks. The proposed risk framework helps control and monitor supply chain processes, particularly in risk identification. The Bayesian network's optimization capabilities allow it to handle large datasets effectively, aiding in risk assessment, monitoring, and mitigation. The significance of machine learning algorithms is emphasized by this study in enhancing supply chain robustness and resilience, ensuring continuity and profitability in supply chain operations.

Sani et al. [31] developed a hybrid Bayesian-optimized Light Gradient-Boosting Machine (LightGBM) model to predict backorder risks in supply chains. The methodology integrates diverse machine learning algorithms, providing computational

efficiency and high accuracy. Findings show the model’s superiority in forecasting risks compared to traditional methods. The study emphasizes the model's potential to address disruptions and demand volatility. A limitation is the need for high-quality data to ensure model accuracy.

Vignesh et al. [32] combined quantum computing and machine learning for supply chain optimization. The hybrid framework leverages quantum annealers and classical machine learning techniques. The methodology includes rigorous experimentation on real-world scenarios, demonstrating significant efficiency gains and enhanced solution quality. Findings indicate that quantum machine learning can substantially reduce costs and improve sustainability. A limitation is the current infancy of quantum computing, which may restrict widespread adoption.

A multi-agent reinforcement learning (MaRL) technique was presented by Zhang et al. [33] for enhancing the efficiency of supply chains and traceability in inventory management. The methodology involves leveraging MaRL and topological information of supply chain networks. Simulation-based evaluations demonstrate superior performance compared to alternative optimization methods. Findings highlight the method's effectiveness in ensuring information security and cost reduction. A noted limitation is the complexity of implementing multi-agent systems in real-world supply chains.

Schroeder and Lodemann [34] carried out a systematic literature review analyzing the integration of ML in supply chain risk management (SCRM). This study identifies the early detection of production, transport, and supply risks as key areas where machine learning adds value. It suggests that integrating new data sources, such as social media and weather data, can significantly enhance SCRM. The paper proposes four research propositions to motivate further exploration of machine learning applications in supply chain risk management. In another study, Aljohani [35] proposed a strategy combining machine learning and predictive analytics to enhance supply chain agility and real-time risk mitigation. Traditional SCM often relies on post-event analysis, but this research advocates for a proactive approach using predictive models to foresee

disruptions. Machine learning models trained on historical and contextual data enable organizations to recognize risks as they emerge and implement preventive measures. The combination of predictive analytics into real-time monitoring improves risk visibility and response times, ensuring operational continuity. A limitation is the need for continuous model optimization to maintain accuracy in dynamic environments.

In another study, Li et al. [36] employed four distinct classifiers and the ML feature selection algorithm to determine the key influencing elements of 12,330 customers' online purchase intention data. Their findings provide valuable insights into consumer behavior and the factors that drive online purchasing decisions, which can be used to optimize supply chain strategies.

Reshehchi et al. [37] aimed to present a data-driven model that uses an analysis of network features to estimate the credit risks within a supply chain financing network. Their research unequivocally shows that considering the actor's network characteristics within prediction models can greatly improve the model's predictability, offering a new perspective on risk management in supply chain finance.

Additionally, by using the MATLAB platform, Han et al. [38] has build a supply chain risk management model based on the principle of neural networks and conduct model simulations. Their work demonstrates the potential of neural networks to provide robust risk management solutions and highlights the importance of simulation tools in developing and testing these models.Constante Nicolalde et al. [39] work with machine learning approaches, notably Random Forest and R part algorithms, used in forecasting smart supply chain fraud. This approach is useful for risk assessment, determining if a transaction is fraudulent or normal, and reducing potential risks. The dataset utilized in their study includes approximately 180k transactions from supply chains that DataCo Global used over a three-year period, demonstrating the scalability and effectiveness of ML techniques in fraud detection.

Lastly, Lahcen Tamym et al [40] also utilized two datasets for his research about good and activity monitoring in supply chain. One of the datasets was DataCo supply chain dataset and it is used for fraud detection. They also incorporated some predictive

analytics. The implemented algorithms are ANN, Support Vector Machine, Decision Tree and lastly logistic regression.

The supply chain risk optimization that has been done in this paper on the DataCO supply chain dataset has produced good results by implementing a hybrid machine learning algorithm which will be discussed later in the paper.

2.3 Previous study regarding DRL

Many researchers are doing their work regarding the implementation of DRL in supply chain optimization, especially pricing policy optimization. Reza Refaei Afshar et al. [41] introduced an automated deep reinforcement learning (DRL) pipeline designed to optimize dynamic pricing strategies. The framework simplifies the application of DRL for non-experts by automating three key design steps: Markov decision process modeling, algorithm selection, and hyperparameter optimization. The hyperparameter optimization stage uses a unique method that blends Bayesian optimization and genetic algorithm techniques to improve refinement. The proposed DRL pipeline considerably outperforms benchmark approaches for pricing policy optimization, as proved by reserve price optimization in online advertising, resulting in increased revenue generation in a real-time bidding scenario simulation. The limitation of the work is the reliance on simulation environments, which constrains real-world validation and necessitates further research to confirm applicability across diverse domains.

Saeed Abdol Hosseini et al. [42] presented a novel approach integrating reinforcement learning with agent-based modeling and simulation-optimization techniques for joint pricing and inventory management in competitive markets. The methodology on optimizing (R, Q) policies in the context of non-zero lead times and lost sales. The algorithm can manage price and inventory while taking into account a wide range of client preferences, as seen by the results, which indicate how successful it is at increasing profit creation when compared to traditional approaches. The study's dependency on simulated environments requires validation in real-world settings, while its computational complexity may constrain scalability in larger applications.

To find demand response within electricity markets, Jun Song et al. [43] presented a novel nonparametric constrained policy optimization approach. The methodology focuses on improving policy stability and optimality by eliminating limiting assumptions about policy representation and using an on-policy actor-critic algorithm. The results gathered from a pair of disaster recovery case studies exhibit exceptional efficiency in load shifting and pricing policy optimization, all while preserving system stability. The method addresses important power system stability issues while guaranteeing strong pricing strategies in DR applications. Nonetheless, there are still a number of important research gaps, including the need for more validation in a variety of demand response scenarios as well as concerns about computing efficiency and practical implementation.

Similar to the DQN algorithm presented in this paper, Inderpreet Singh [44] employed a Deep Q-Network (DQN) approach to dynamic pricing within the hospitality sector, specifically targeting hotel room pricing. In order to simulate demand distributions and optimize pricing strategies for maximum daily profits and lowest room vacancy, it uses a Random Forest model using real-world hotel booking data. The outcomes show a significant 15-20% increase in earnings and notable drop in the quantity of vacant rooms compared to traditional techniques. Nevertheless, the research's focus is restricted to one aspect of the hospitality sector, highlighting the necessity for more study in a variety of fields in order to properly confirm and generalize its conclusions.

David Chiumera et al. [45] introduced Proximal Policy Optimization (PPO) as a deep reinforcement learning (RL) framework for time series forecasting in quantitative finance. In order to improve important parameters like learning rates and discount factors, the study modifies an RL environment with an emphasis on price prediction and strategy building. PPO has strong performance across several datasets, sometimes outperforming conventional buy-and-hold strategies in terms of market price prediction and conditional flexibility. However, the majority of the study concentrates on the histories of specific stock markets, suggesting that further studies into multi-

asset portfolios and a wider range of financial instruments are necessary to confirm its wider applicability and resilience across financial markets.

Yajun Hu et al. [46] introduced a distributed dynamic pricing strategy for e-retailers utilizing the Dueling Deep Q-Network (DQN) algorithm, framed within a presale environment modeled as a Markov decision process (MDP). The goal of the study is to optimize price decisions by taking customer behavior and inventory backlog into account. Dueling DQN algorithm's strong profit maximizing skills and flexibility to different presale models emphasize its effectiveness in dynamic pricing strategies designed for e-commerce scenarios. To evaluate the study's practical application, real-world validation is essential, as indicated by its dependence on simulated surroundings. A subsequent study endeavors to improve pricing systems' security and transparency in e-commerce platforms by investigating the incorporation of blockchain technology.

Jian Liu et al. [47] presented an innovative framework for dynamic pricing on e-commerce platforms, leveraging deep reinforcement learning (DRL). The dynamic pricing problem is expressed as a Markov Decision Process (MDP), with states represented by various business data sets. It uses a novel reward function, which is the difference in revenue conversion rates, instead of traditional revenue-based rewards. The cold start problem in MDPs is also addressed by pretraining and evaluating carefully selected historical data. This work's limitations include separate training and pricing for each product and using only product-related features as the environment state.

Mattew Huber et al. [48] introduced a novel approach to neural network architecture optimization using reinforcement learning, focusing on learning policies within an abstract embedding space for network optimization. The methodology has shown early success in optimizing topologies for common classification issues, and its goal is to gradually enhance network performance across a range of workloads. When this method is used to transfer learning scenarios, it adapts networks to new tasks effectively and without requiring a significant amount of retraining. The study can only

be applied so broadly since it relies on assessment from a single categorization problem. To prove its scalability and resilience in real-world scenarios, future studies should investigate how well it works with other neural network topologies and workloads.

DRL can also be applied effectively in inventory and logistics management scenarios. Toshiyuki Demizu et al. [49] have done his work on a model-based deep reinforcement learning approach for inventory management of new smartphone products, aiming to optimize stock levels and minimize lost opportunities and defective inventory. The approach solves data scarcity issues common to new product launches by fusing online planning with offline model learning. When compared to conventional approaches, evaluation based on actual sales data shows improved profitability, efficiency, and customer satisfaction. Its generalizability will need to be determined through more testing in a variety of product categories and retail environments. Furthermore, the use of historical data for model training presents challenges in rapidly changing market environments, indicating potential directions for future study to improve flexibility and the capacity to make decisions in real-time.

DELLMM, a deep reinforcement learning-based logistics management model is introduced by Li Yang et al. [50] which is enhanced with blockchain technology, aimed at optimizing distribution and resource balance in dynamic transport networks. The approach aims to increase trust and transparency in logistics operations by incorporating blockchain. The outcomes of the experiment demonstrate notable advancements in important parameters such as sustainability, trust enhancement, operability, efficiency, and latency reduction. However, the report admits the need for more research into blockchain technology integration and underlines the importance of real-world validation to determine its usability and scalability in a variety of logistical contexts.

Guoquan Wu et al. [51] introduced a distributional reinforcement learning approach tailored for optimizing inventory management in multi-echelon supply chains. The approach integrates risk-sensitive formulations to improve policy optimization and

places a strong emphasis on striking a balance between exploration and exploitation to reduce the danger of less-than-ideal results. In particular, the experimental findings show considerable gains in reducing low-probability, and high-severity events, as well as overall performance as compared to standard benchmarks. This research demonstrates how distributional reinforcement learning may be used to increase supply networks' resilience and effectiveness in operation.. To confirm its application to various supply chain architectures and market dynamics, more study is necessary. Furthermore, the algorithm's computational complexity presents difficulties for scalability in large-scale applications.

Deep Reinforcement Learning-based Ordering Mechanism (DRLOM) designed for optimizing inventory is designed by Devan S. Kurian et al. [52] for optimizing inventory ordering in multi-echelon linear supply chains. The approach formulates and solves the ordering management issue as an agent-based reinforcement learning model via proximal policy optimization. Experiments show that DRLOM outperforms evolutionary computing techniques and conventional ordering heuristics in terms of minimizing overall inventory costs in a variety of issue scenarios. The study emphasizes how deep reinforcement learning may improve supply chain efficiency by managing inventories more effectively. Nevertheless, a present research deficit is highlighted by the requirement for validation in more intricate and dynamic supply chain contexts. Moreover, the evaluation's dependence on particular case studies limits the findings' wider relevance and generalizability to various supply chain contexts.

Qian Zhou et al. [53] designed a joint pricing and inventory management system that incorporates deep reinforcement learning (DRL) to account for reference price effects on consumer behavior. The model is based on an infinite-horizon Markov Decision Process (MDP) and employs the Double Deep Q-Network (TN-DDQN) algorithm to optimize pricing and ordering decisions. The method seeks to optimize overall discounted revenues for merchants by taking into account factors such as market volatility and customer sensitivity to price fluctuations. The study finds that ignoring the current price's impact on future demand and customer's memory of past prices can

harm profits. The model requires significant computing power and may need adjustments for practical use due to potential oversimplifications.

Rui Wang et al. [54] introduced a deep reinforcement learning (DRL) model to tackle the joint pricing and inventory control problem for perishables with a positive lead time. The model aims to maximize predicted profits over a finite horizon by utilizing the Double Deep Q-Network (TN-DDQN) algorithm to identify near-optimal ordering and pricing strategies. It takes into consideration both lost-sales and backlog situations, in which consumer demand is influenced by the current price and follows a Poisson pattern. The study finds that dynamic pricing techniques beat fixed pricing strategies because they alter prices depending on inventory levels and item shelf life. The model only addresses a single agent's pricing and inventory control, neglecting the interactions and impact of multiple agents in supply chains.

Dawei Qiu et al. [55] introduced a novel deep reinforcement learning (DRL) method, prioritized deep deterministic policy gradient (PDDPG), for optimizing pricing strategies in electric vehicle (EV) charging. It addresses the challenge of discrete charging levels by using multi-dimensional continuous state and action spaces, showing superior performance over traditional methods. The approach solves the challenge in multi-dimensional continuous state and action spaces by fusing the ideas of DDPG with a prioritized experience replay technique. PDDPG offers improved solution optimality and lower processing needs by capturing the discrete character of EV charging, in contrast to previous techniques. Case studies demonstrate that PDDPG achieves higher profit and computational efficiency compared to state-of-the-art reinforcement learning techniques such as Q-learning and deep Q networks (DQN). A limitation of this study is that it does not yet incorporate the realistic variability of EV traveling patterns and wholesale prices, which is proposed as future work.

Pei-Yung Chou et al. [56] presented a Deep Deterministic Policy Gradient (DDPG) algorithm applied to the user association and video quality selection problem in Mobile Edge Computing (MEC) for live video streaming. The methodology addresses this challenge by formulating it as a non-linear integer programming problem, leveraging

Lagrangian multipliers to derive a closed-form solution. According to simulation results, QoE can be significantly improved, especially in situations when there are a lot of users and not enough wireless resources. Video quality and user association can be optimized better than with baseline approaches. The paper emphasizes how deep reinforcement learning may enhance MEC performance in real-time video streaming applications. However the non-linear programming problem's computational complexity poses challenges to scalability in real-world applications.

Chencheng Chen et al. [57] introduced a reinforcement learning enhanced agent-based modeling and simulation approach (RL-ABMS) to address spatial-temporal pricing in ride-sourcing platforms. The study uses actor and critic neural networks to dynamically optimize pricing strategies using the Proximal Policy Optimization (PPO) method. Dynamic pricing alone improved platform profit by 1.25 times, while spatial-temporal pricing increased it even further to 1.85 times. Improved supply-demand coordination was also shown by the approach's notable reduction in the number of idle drivers or cars. The study emphasizes how spatial-temporal pricing techniques might improve ride-sourcing platform’s operational efficiency and profitability. However because ride-sourcing systems are sophisticated and rely on simulation findings and they need to be rigorously validated in a variety of real-world settings. Potential avenues for further research include examining the impact of regulatory frameworks and traffic dynamics on pricing methods.

Seung Lee et al. [58] introduced a novel privacy-preserving distributed deep reinforcement learning (DRL) framework for optimizing the energy management and dynamic pricing of multiple smart electric vehicle charging stations (EVCSs) integrated with photovoltaic systems and energy storage. The study utilizes a hierarchically distributed methodology in which local DRL agents find the best-selling pricing and charging/discharging schedules by applying the soft actor-critic method. The system protects data privacy by implementing federated reinforcement learning (FRL), which trains global and local models without exchanging sensitive operational data amongst EVCSs. Adaptive pricing techniques and optimal energy consumption under variable environmental conditions are achieved by the methodology, as demonstrated by numerical examples. The study's reliance on simulations and the

difficulty of federated learning implementation in actual EVCS setups, however, provide obstacles to practical adoption.

Eduardo J. Salazar et al. [59] presented a reinforcement learning-based pricing and incentive approach for demand response (DR) in smart energy systems, integrating both price-based and incentive-based models to manage consumer demand efficiently. The model optimizes short-term and long-term price strategies, successfully balancing supply and demand, by utilizing real-time and time-of-use pricing schemes together with Q-learning with memory exchange. The study shows notable gains in demand displacement and load factor, underscoring the model's potential to lower power prices and increase grid dependability. Still, more validation in various real-world circumstances is required due to the dependence on simulation results. To improve the suggested technique and handle potential peak rebound effects, future studies should take into account different consumer types, elasticity variables, and nodal pricing effects.

Alexander Kastius et al. [60] examined the application of reinforcement learning (RL) to dynamic pricing under competitive market conditions, employing Deep Q-Networks (DQN) and Soft Actor Critic (SAC) algorithms. The paper acknowledges the complexity and dimensional problems of monopoly environments while highlighting the improved performance of SAC over DQN in duopoly and oligopoly market settings. Dynamic programming (DP) was used to verify both methods, and the results showed that SAC performs better in cases with more complexity. SAC's trouble with straightforward fixed-price plans and DQN's incapacity to manage a variety of scenarios are two drawbacks, nevertheless. The work emphasizes the need to collect large amounts of observational data, and it makes recommendations for future work to improve data efficiency and investigate multi-task reinforcement learning applications for various product sectors.

Angel Fraija et al. [61] presented a Demand Response Aggregator (DRA) model that leverages reinforcement learning (RL) for transactive policy generation, integrating a convex optimization problem on the customer side to manage privacy and avoid

penalties through price discounts. The model assures convergence and accelerates the DRA convergence process through offline training, allowing for adaptive Time-of-Use (ToU) tariffs and near-optimal pricing strategies. The methodology is verified through the use of residential agents, exhibiting effectiveness in load distribution and comfort preservation, as well as better results than with conventional reinforcement learning techniques. To completely address practical implementation issues, more research is needed on the model for heterogeneous residential agents and real-world applications.

2.4 Conclusion

This chapter offers an in-depth comprehension of contemporary research pertaining to supply chain risk and pricing policy optimization, encapsulating the key findings and breakthroughs in the field and based on the research gap, the objectives of this thesis work are fixed up.

Chapter 03

Methodology

3.1 Introduction

The first part of the methodology presents a machine-learning approach for supply chain risk optimization. For any machine learning model to work, one needs to follow certain steps for successful completion of the work. All the steps begin with the data collection process. In the later part, with the aim to optimize pricing strategies of a supply chain, differential price response models and various reinforcement learning techniques are introduced including DQN and SARSA.

3.2 Supply chain risk optimization using machine learning

In this section, the policy of risk optimization of a supply chain is discussed in detail. Starting from the data collection, data preprocessing, application of various popular machine learning algorithms is reviewed in this chapter. Moreover, two hybrid algorithms are and discussed.

3.2.1 Data Collection

The dataset used in this study is from DataCo Global Supply Chain [62], which includes a collection of the company's sold products, financial information (profit, loss, total sales, etc.), shipping information, and customer information including sales, demographics, and transaction information. The information spans 53 columns related to clothing, sports, and electronic supplies and includes information on 180,520 customers. Areas of important registered activities in the DataCo Global dataset are provisioning, production, sales, and commercial distribution. It also allows the correlation of Structured Data with Unstructured Data for knowledge generation.

3.2.2 Structured data formation

Machine learning algorithms can quickly and readily understand highly organized structured data, which is often characterized as quantitative data. Yet, unstructured data, which is frequently classified as qualitative data, cannot be processed and analyzed using conventional data tools and techniques.

Structured data is typically stored in tabular formats, like databases or spreadsheets, where data points are organized into rows and columns. Examples include transaction records, inventory data, and customer details. This data is easy to enter, store, query, and analyze, making it highly suitable for machine learning algorithms. Techniques like regression analysis, classification, and clustering are often applied to structured data to uncover patterns, predict outcomes, and optimize processes.

Unstructured data, on the other hand, includes text, images, audio, and video files that do not fit neatly into rows and columns. Examples are emails, social media posts, and multimedia content. To extract useful insights from this kind of information, more sophisticated processing methods like computer vision, deep learning, and natural language processing (NLP) are needed. Tools and techniques like tokenization, sentiment analysis, and neural networks are employed to handle the complexity and variety inherent in unstructured data.

In supply chain management, the integration of structured and unstructured data can enhance decision-making and operational efficiency. For instance, structured transaction records can be combined with unstructured customer feedback from social media to gain a comprehensive understanding of customer satisfaction and product performance. The DataCO Supply Chain dataset is a comprehensive resource that includes both structured and unstructured data, offering a robust platform for advanced data analysis and machine learning applications. Here are some key features of the dataset.

Transaction Records: Over 180,000 unique order records that include details like order IDs, product categories, quantities ordered, customer locations, and payment methods.

Customer Data: Information on more than 30,000 unique customers, enabling detailed segmentation and personalized marketing strategies.

Product Information: Data on over 10,000 unique products, providing insights into inventory management and product popularity.

Geographical Coverage: Data spans multiple countries, allowing for regional analysis and global supply chain insights.

Fraud Indicators: Contains markers for potential fraud, aiding in the development of fraud detection models.

By leveraging both structured data (like transaction records and customer demographics) and unstructured data (like tokenized clickstream logs), organizations can optimize their supply chain operations, improve financial performance, and enhance overall efficiency

3.2.3 Data analysis and preprocessing

The next step after the formation of structured data is to do the data preprocessing and after that data analysis is done. Data modelling is done in the next step. The flowchart and the framework of the model is now presented in the next page in Figure 3.1.

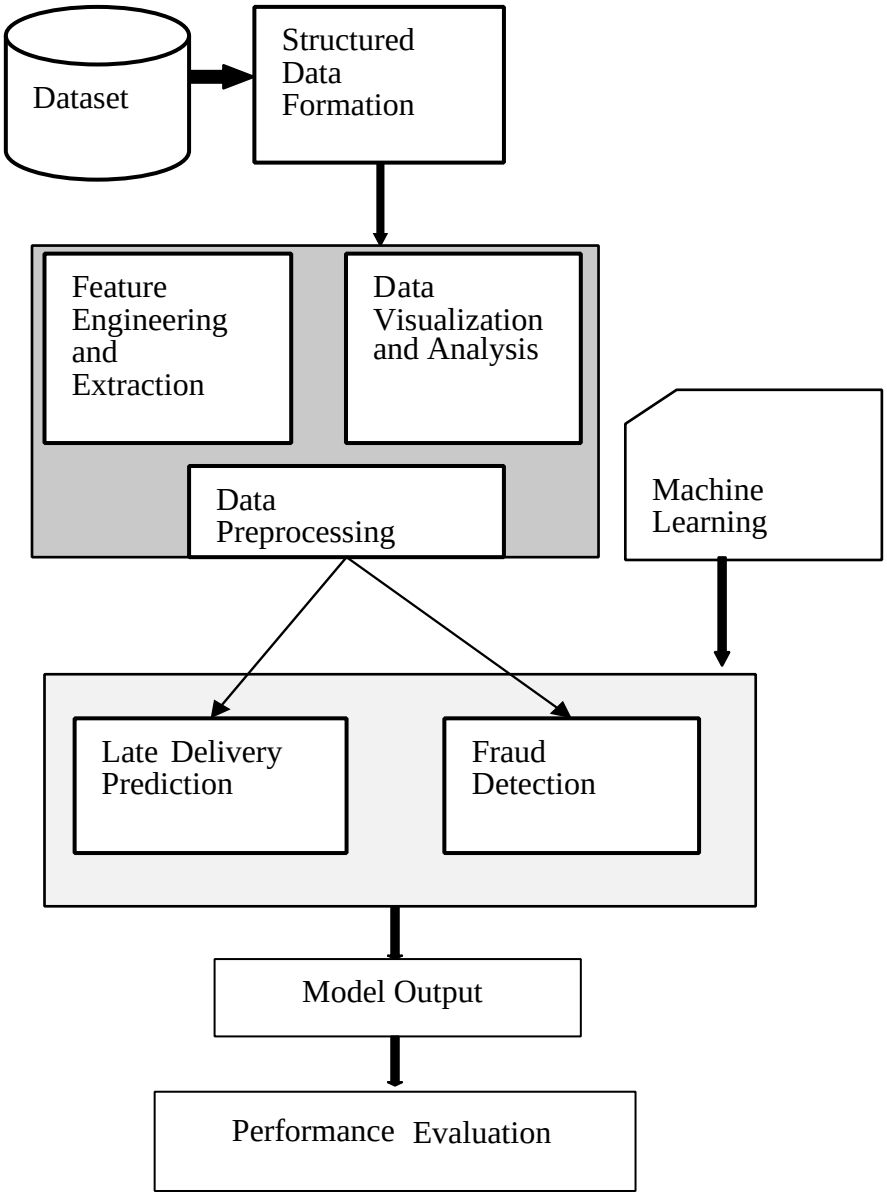

Figure 3.1: Framework of model (Analytics model of supply chain risk optimization)

For the trend analysis it is needed to visualize how the product price is affecting the sales per customer. A dotted line plot graph typically represents a series of data points connected by dotted lines to show a trend or relationship between the data points. This type of graph is often used to display information over time or to illustrate changes in variables. Figure 3.2 represents that visualization through dotted line plot.

Figure 3.2: Relation between product prize and sales per customer

The next stage of the work is the formation of the heatmap from the dataset. A heatmap represents a graphical visualization of data where the individual values contained in a matrix are represented as colors. In the context of machine learning, a heatmap is often used to visualize the correlation or relationship between different features or variables in a dataset. From the heatmap analysis of this dataset, all the necessary features are extracted. In total **22** features are considered for late delivery analysis and fraud detection.

The data analysis of fraud transactions in the DataCO supply chain dataset involves a comprehensive examination of several key aspects. The dataset is grouped by payment type, including Transfer, Cash, Payment, and Debit, and the number of transactions for each payment type is calculated per region. This information is then visualized in a bar chart titled "Different Types of Payments Used in All Regions." The X-axis lists the order regions, and the y-axis shows number of payments, with different colors representing the various payment methods. This bar chart offers a comparative view of payment preferences across regions, highlighting which payment methods are most popular in specific areas. Such insights are crucial for tailoring payment processing

systems to regional preferences, potentially improving customer satisfaction and operational efficiency. Figure 3.3 represents the bar chart which indicates different types of payment used.

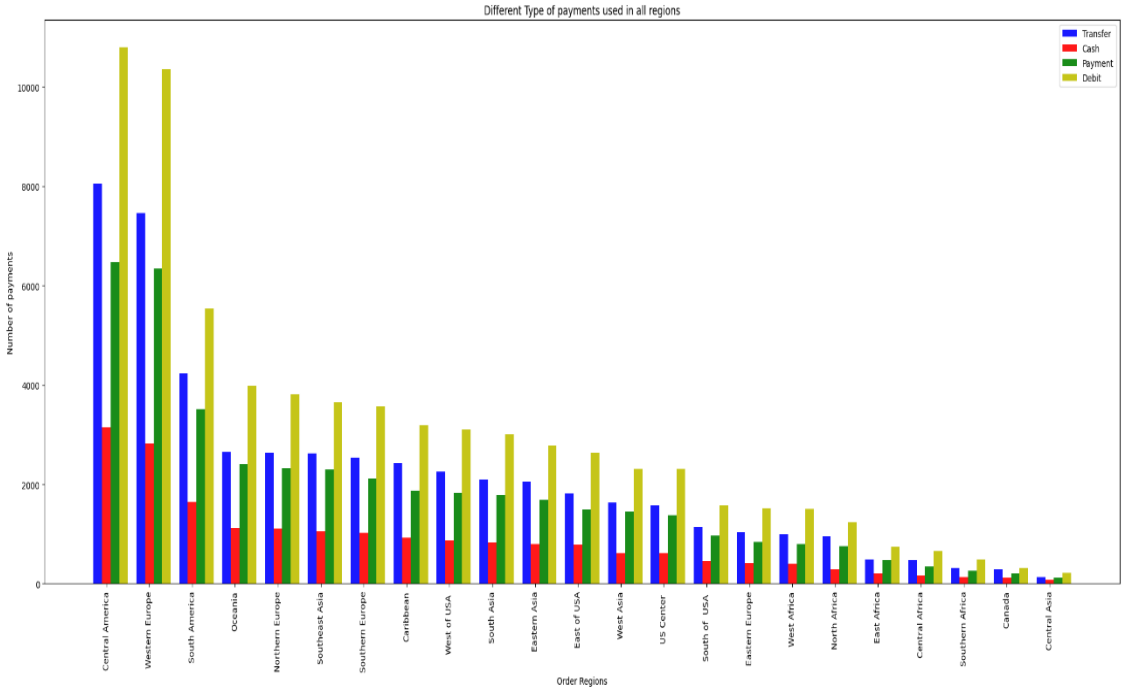

Figure 3.3: Visualization of payment method used

Finally, to identify regions with the highest suspected fraud transactions, the data is filtered to include only those transactions marked as 'SUSPECTED_FRAUD' and using the 'TRANSFER' payment method. The results are displayed in a pie chart titled "Regions with Highest Fraud," which shows the percentage distribution of suspected fraud cases across different regions. This pie chart helps pinpoint areas with a higher concentration of fraudulent activity, which is essential for targeted fraud prevention measures. By focusing on these regions, supply chain managers can implement more stringent security protocols and monitor transactions more closely, thereby reducing the risk of fraud. This pie chart is presented below in percentage at Figure 3.4 .

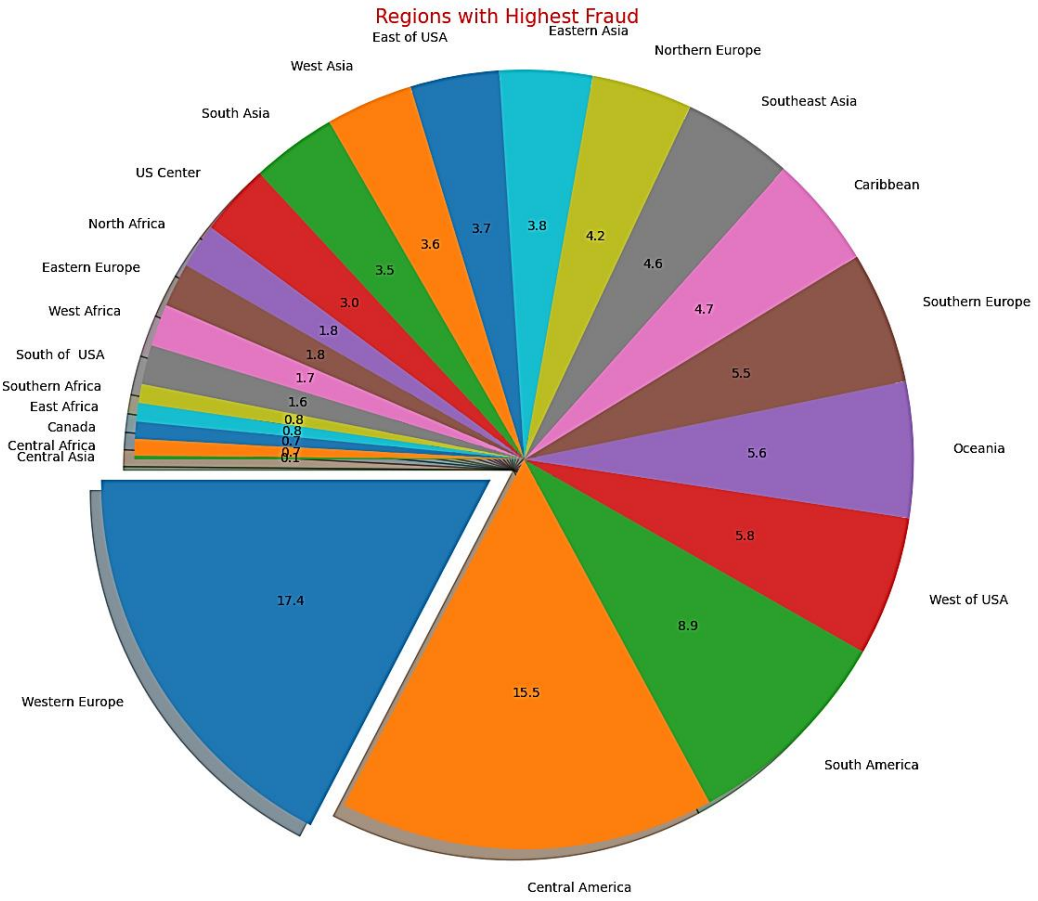

Figure 3.4: Pie chart representing the regions with fraud

After that, Recency, Frequency and Monetary (RFM) analysis is done for the customer segmentation purpose. RFM (Recency, Frequency, Monetary) analysis is a vital marketing tool used to assess customer value by examining their purchasing behavior. This technique segments customers based on how recently they made a purchase (Recency), how often they make purchases (Frequency), and how much they spend (Monetary). Customers with a total score of 11 or 12 were classified as "Champions," indicating high value across all three dimensions. Those with a score of 10 were deemed "Loyal Customers," and a score of 9 identified "Recent Customers." Other segments included "Promising" (score of 8), "Customers Needing Attention" (score of 7), "Can't Lose Them" (score of 6), "At Risk" (score of 5), and "Lost" (score less than 5).

After that, customer segmentation is done and Recency, Frequency and Monetary (RFM) analysis is done for this purpose. The customer base is segmented into 8 portions and 16.9% of them are promising customers, 11% of customers need more attention, 33.2% are recent customers. Whereas 0.6% and 10.5% of the customers are champions and loyal customers respectively which is shown in Figure 3.5.

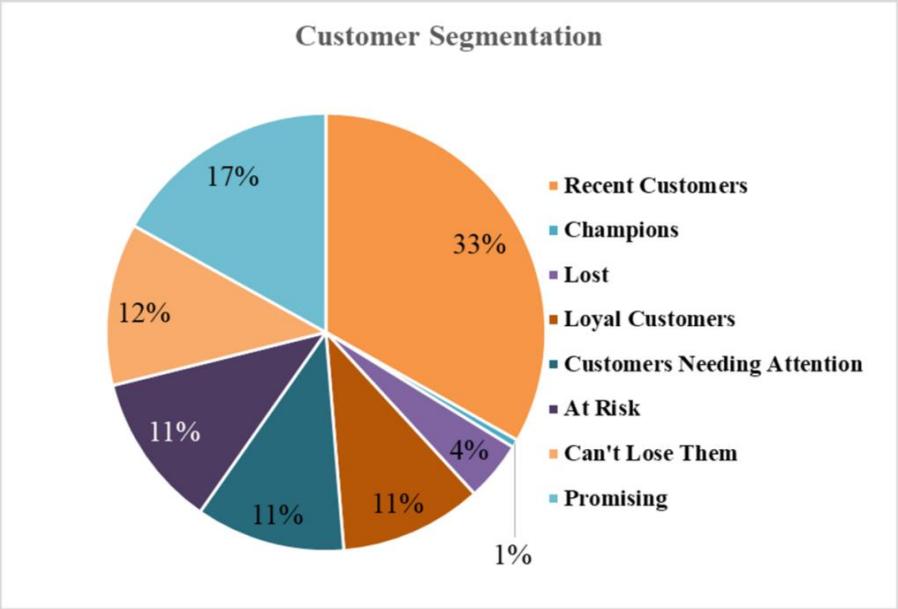

Figure 3.5: Pie chart representing customer's segmentation.

After the data analysis, data modeling is done for fraud detection, prediction and late delivery prediction.

3.2.4 Machine learning algorithms used

For implementing the supply chain risk optimization, in this research a total of 14 Machine learning algorithms are used. They are-

1. Logistic Regression
2. Gaussian Naive Bayes
3. Support Vector Machines
4. K nearest Neighbors
5. Linear Discriminant Analysis

6. Random Forest Classifier
7. Extra Trees Classifier
8. Extreme Gradient Boosting (XGB Classifier)
9. Decision Tree Classifier
10. Ada Boost Classifier
11. Histogram Gradient Boosting Classifier
12. Light GBM Classifier
13. Multi-Layer Perceptron (MLP) Classifier
14. Hybrid Model (MLP + Random Forest + Extra Trees Classifier)

Here some popular but no so efficient basic algorithms are used. Additionally, some ensemble algorithms are used which produced good results. But, the best result are produced by the hybrid algorithm developed in this research work. All the algorithms and their results along with their performance matrices and confusion matrix will be discussed in the results chapter.

3.3 Supply chain pricing policy optimization using deep reinforcement learning

The traditional price optimization process involves a demand model that captures various factors influencing demand, such as regular prices, discounts, marketing activities, seasonality, competitor prices and cross-product cannibalization. This research extends the traditional models by incorporating temporal dependencies and optimizing price schedules dynamically by reinforcement learning techniques.

3.3.1 Environment design

The environment models the demand and profit based on price changes. The core functions of the model are defined as follows:

Price-Demand Function: Models the demand at time step t for the current price p_t and previous price p_{t-1} which is presented in Eqn. (3.1).

$$\boldsymbol{q_t}(\boldsymbol{p_t}, \boldsymbol{p_{t-1}}, \boldsymbol{q_0}, \boldsymbol{k}, \boldsymbol{a}, \boldsymbol{b}) = max(0, q_0 - k \cdot p_t - a \cdot \sqrt{\max(0, p_t - p_{t-1})} + b. \sqrt{\max(0, p_{t-1} - p_t)} \quad (3.1)$$

Profit Function: Calculates the profit at time step t by using Eqn. (3.2).

$$\boldsymbol{profit_t(p_t, p_{t-1}, q_0, k, a, b, unit\_cost)} = q_t(p_t, p_{t-1}, q_0, k, a, b) \cdot (p_t - unit\_cost) \quad (3.2)$$

Total Profit Function: Computes the total profit over a series of time steps as in Eqn. (3.3).

$$\boldsymbol{profit_{total(p, unit_{cost}, q_0, k, a, b)}} = \sum_{t=0}^{T} profit_t(p[t], p[t-1], q_0, k, a, b, unit_{cost}) \quad (3.3)$$

The impact of temporal dependencies on the pricing optimization process is explored by incorporating a price-demand function that accounts for recent price changes. Eqn. (3.4) is the function that acts as the optimization basis which is defined as

$$\boldsymbol{d(p_t, p_{t-1}) = d_0 - k.p_t - a.s((p_t - p_{t-1})^+ + b.s((p_{t-1} - p_t)^-} \quad (3.4)$$

where:

$$x^+ = \max(0, x)$$

$$x^- = \max(0, -x)$$

s(x)= $\sqrt{x}$

3.3.2 Real-time online based marketplace data

Here, in this research, no simulated environment is used. Real-time data is obtained from one of the leading online marketplace-based T-Shirt selling brand. “ABC” company is doing online-based T-shirt business for over 5 years now. Data has been collected by following proper procedures and the company identity is remained anonymous. The data which are collected are-

T (Time Step): The parameter T represents the number of time steps in the simulation, defining the length of the simulation horizon. For instance, T is set to 10, the simulation will run for 10-time steps. At each time step, decisions are made, and outcomes are observed, allowing the model to evaluate the effects of these decisions over the entire

simulation period. In the case of ABC company, Each T represents time period of 2.5 months.

Maximum Unit Price: The unit price parameter defines the maximum possible price in the simulation, creating an upper limit for the price grid. This parameter is crucial for setting realistic boundaries within which the pricing strategy operates. In the case of ABC company, the maximum unit price is 300 Tk.

The price step: The price step parameter specifies the step size for the price grid, determining its granularity. In the case of ABC company, the price step is considered ass 20 Tk, which means price changes occur at 20 Tk per unit at a specified time.

q_0(initial demand): The parameter q_0 represents the initial demand at the start of the year, setting the baseline demand before any price adjustments are made. This baseline is critical for understanding how subsequent pricing decisions affect demand. In the case of ABC company, the initial demand is 1000 units.

The price elasticity coefficient (k): It indicates how sensitive the demand is to changes in the price. This parameter helps in modeling the relationship between price changes and demand fluctuations, reflecting consumer behavior. In the case of ABC company, The price elasticity coefficient (k) is 2.

Unit cost: This parameter represents the cost to produce or acquire one unit of the product or service. It is used to calculate profit by subtracting the cost from the selling price. This parameter is essential for profit calculation and for making pricing decisions that ensure profitability. In case of ABC company, their unit cost is around 98-104 tk, so it's considered to as 100 tk based on the opinion of their chief financial officer. For ABC company,

Unit cost= Cost of raw materials+ Cost of making+ Transportation Cost+ Marketing cost (3.5)

This marketing cost varies from time to time. But as they are a saturated kind company now, their present marketing cost is not that much.

Shock Constant(a_q): The parameter a_q is the coefficient that affects the demand change due to positive price shocks, representing how much the demand decreases when the price increases. For ABC company, a_q is 300, a positive price shock (price increase) will cause the demand to decrease significantly. This coefficient helps model the adverse effects of price increases on demand.

Shock Constant(b_q): It is the coefficient that affects the demand change due to negative price shocks, representing how much the demand increases when the price decreases. For ABC company, b_q is 100, a negative price shock (price decrease) will cause the demand to increase, but not as significantly as it decreases with a positive shock. This asymmetrical response reflects the typical market behavior where consumers are less responsive to price decreases compared to increases.

3.3.3 Implementation of price optimization algorithms

In this part of the thesis, traditional constant price optimization and greedy dynamic price optimization techniques are discussed and then the implementation of Deep reinforcement learning is presented.

3.3.3.1 Constant price optimization

The optimal constant price is the price that maximizes profit over the entire period. The profit is evaluated for a range of price levels, and the price with the highest profit is selected. The pseudocode is now given below-

Pseudocode

```
FOR each price in price grid:
        Compute total profit for constant price over T time steps
        Store profit and corresponding price
Select price with maximum profit
```

3.3.3.2 Greedy dynamic price optimization

Next, a greedy algorithm is employed to optimize the price schedule dynamically. Starting with the optimal constant price, the price is optimized iteratively for each time step. The pseudocode for greedy dynamic price optimization is now given below

```
Pseudocode:
Initialize price schedule with optimal constant price
FOR each time step t in T:
    FOR each price in price grid:
        Compute total profit for the price at time t, keeping other prices constant
Select price with maximum profit for time step t
```

3.3.3.3 Deep reinforcement learning approaches

The implementation of the two algorithms employed in this study is explained in this section of the paper. The description is not based on general discussion, rather it is kept more detailed with the execution of the algorithm in line with the problem description of the study.

3.3.3.3.1 Deep Q Network (DQN) implementation

The DQN algorithm is employed to learn the ideal pricing strategy by estimating the Q-function via a deep neural network. The Deep Q-Network (DQN) algorithm, introduced by Mnih et al. [21], integrates Q-learning and deep neural networks in order to address complicated reinforcement learning issues. The DQN has been particularly effective in environments with high-dimensional state spaces and action spaces. The pseudocode of a DQN algorithm is now given below-

Pseudocode:

```
Initialize policy and target networks with random weights
Initialize replay memory
FOR each episode in num_episodes:
    Initialize state
    FOR each time step t in T:
        Select action using epsilon-greedy policy
        Execute action and observe reward and next state
        Store transition in replay memory
        Sample random batch from replay memory
        Compute target Q-values using target network
        Compute loss and update policy network
        Update target network periodically
```

3.3.3.3.2 Neural network architecture of DQN

The neural network of DQN serves as the function approximator for Q-values, often referred to as the Q-network. Conventional Q-learning involves storing Q-values in a list, which is only practical in contexts with a limited state-action space. Nevertheless, in more complex environments with vast state-action spaces, it is impractical to maintain a Q-table. Therefore, DQN employs a neural network to approximate the Q-values.

The design of the implemented neural network involves mainly three layers, including an input layer, multiple hidden layers, and an output layer:

Input Layer: The input layer takes the current state of the environment as input. The size of the input layer is equal to the dimensionality of the state space. For instance, if the state is represented as a vector of length 2*T, the input layer will have 2*T neurons.
Hidden Layers: The network typically consists of multiple hidden layers to capture complex patterns in the data. In this research, a network with three hidden layers, each with 128 neurons is used. Each hidden layer uses the Rectified Linear Unit (ReLU)

activation function, which helps the network learn non-linear representations. The ReLU function is defined as f(x) = max(0, x), introducing non-linearity while avoiding the vanishing gradient problem often encountered with other activation functions like sigmoid or tanh.

Output Layer: There are as many neurons in the output layer as there are possible actions in the environment. Each neuron represents the Q-value corresponding to a specific action given the current state. The output layer does not use an activation function, as the raw Q-values is needed for the action selection process. The neural network takes the current state as input and outputs the Q-values for all possible actions. In order to reduce the loss between the target Q-values derived from the Bellman equation and predicted Q-values, the network parameters (weights and biases) are modified during training. The network effectively learns to approximate the optimal Q-value function, guiding the agent to make the best decisions.

Replay Memory: Replay Memory, also known as Experience Replay, is a technique used to store and reuse past experiences. It addresses two major issues in reinforcement learning: the correlation between consecutive transitions and the inefficient use of past experiences.

Capacity: The replay memory has a predefined capacity, such as 10,000 transitions. Once the memory is full, the oldest transitions are discarded to make room for new ones.

Storing transitions: Each time the agent interacts with the environment, the resulting transition is stored in the replay memory. This ensures a diverse set of experiences are available for training.

Sampling: During training, a random batch of transitions is sampled from the replay memory. This random sampling helps break the correlation between consecutive transitions, leading to more stable and efficient learning.

Epsilon-Greedy policy: The epsilon-greedy policy is a simple yet effective strategy to balance exploration and exploitation during training. Exploitation is selecting the most well-known action for maximum rewards, whereas exploration entails trying new acts to learn about their consequences. The epsilon-greedy policy controls this balance using the epsilon parameter (ε).

The epsilon-greedy policy operates as follows:

Random action with probability ε: With probability ε, the policy selects a random action. This encourages exploration, allowing the agent to discover potentially better actions that it might not have chosen otherwise.

Best action with probability 1-ε: The strategy chooses the action with the highest Q-value forecasted by the Q-network with chance 1-ε. This encourages exploitation, allowing the agent to leverage its current knowledge to increase the rewards.

Epsilon decay: To ensure a good balance between exploration and exploitation, the epsilon value is typically decayed over time. This means that the agent starts with a high epsilon value (encouraging exploration) and gradually reduces it to a lower value (encouraging exploitation) as training progresses. The decay can be implemented as an exponential decay, linear decay, or any other suitable schedule.

Q-Learning update rule: The goal of the value-based reinforcement learning technique known as Q-learning is to become proficient in the optimal Q-value function, which stands for the highest possible expected accumulated reward for every state-action pair.. The Q-learning update rule is used to iteratively update the Q-values based on the agent's interactions with the environment. Bellman equation for Q learning, which provides a recursive relationship for the Q-values is:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\big(r + \gamma.\max_{a'} Q(s',a') - Q(s,a)\big) \quad (3.6)$$

Where-

- Q*(s, a)* is the Q-value for state s and action a
- α (alpha) is the learning rate
- *r* is the reward
- γ(gamma) is the discount factor
- s' is the next state
- a' is the next action

Current Q-value: The current Q-value Q(*s*,*a*) represents the expected return of taking action *a* in state *s*, based on the agent's current knowledge. This value is updated using the observed reward and the estimated future reward.

Learning rate (α): The rate of learning establishes the amount that Q-value is adjusted during the update. A higher learning rate results in larger adjustments, allowing the agent to learn more quickly but potentially causing instability. A lower learning rate results in smaller adjustments, leading to slower but more stable learning.

Reward (r): The reward is the immediate return achieved after taking action *a* in state *s*. This value provides feedback on the immediate outcome of the action. The reward(r) may be positive (indicating a desirable outcome) or negative (indicating an undesirable outcome).

Discount factor (γ): The discount factor measures the importance of future rewards. A value closer to 1 means future rewards are highly appreciated, while a value close to 0 means immediate rewards are preferred. The discount factor helps balance the consideration of short-term and long-term returns.

Next state value: The next state value $\max_{a'} Q(s', a')$ is the maximum Q-value for the next state s' over all possible actions a'. This value estimates the best possible future return starting from the next state. The Q-learning update rule uses this estimate to incorporate the potential future rewards into the current Q-value.

In the context of DQN, the Q-values are approximated using one neural network. The update rule for the target Q-value, y_j is-

$$\boldsymbol{y_j = r_j + \gamma max_{a'} Q(s_{j+1}, a'; \theta^-)} \quad (3.7)$$

Here's what each term represents:

r_j: The reward received after taking action a_j in state s_j.

γ: The discount factor, which determines the significance of future rewards.

$max_{a'} Q(s_{j+1}, a'; \theta^-)$: The maximum predicted Q-value for the next state s_{j+1} over all possible actions a', using the target network parameters θ^-.

The Algorithm of the DQN algorithm that is used in this research paper for supply chain pricing policy optimization is now given below in Algorithm 1.

Algorithm 1: Algorithm for Implemented DQN Algorithm

Initialize replay memory D to capacity N.

Initialize policy network Q with weights θ.

Initialize target network Q′ with weights θ′

For episode = 1 to M:

Initialize state s_t as a zero vector of size 2T

Initialize reward_trace and price_schedule.

For t=1 to T:

Select action a_t with probability ϵ.

Otherwise, select $a_t = argmax_a Q(s_t, a; \theta)$

Execute action at, observe reward r_t and next state s_{t+1}

Store transition (s_t, a_t, s_{t+1}, r_t, a_{t+1}) in D.

If (memory D contains more than batch size B):

Sample random mini-batch of transitions (s_j, a_j,s_{j+1},r_j,a_{j+1}) from D

Compute y_j:

$y_j=r_j$ if s_{j+1} is terminal.

Otherwise, $y_j = r_j + \gamma max_{a'} Q\left(s_{j+1}, a'; \theta^-\right)$

Perform gradient descent step on $y_j - Q\left(s_j, a_j; \theta\right)^2$ with respect to θ.

Set $s_t=s_{t+1}$

Append reward r_t to reward_trace.

Plot average price schedules for the episodes.

Display best profit results from the training.

The flow chart of the implemented DQN algorithm is now given below in Figure 3.6-

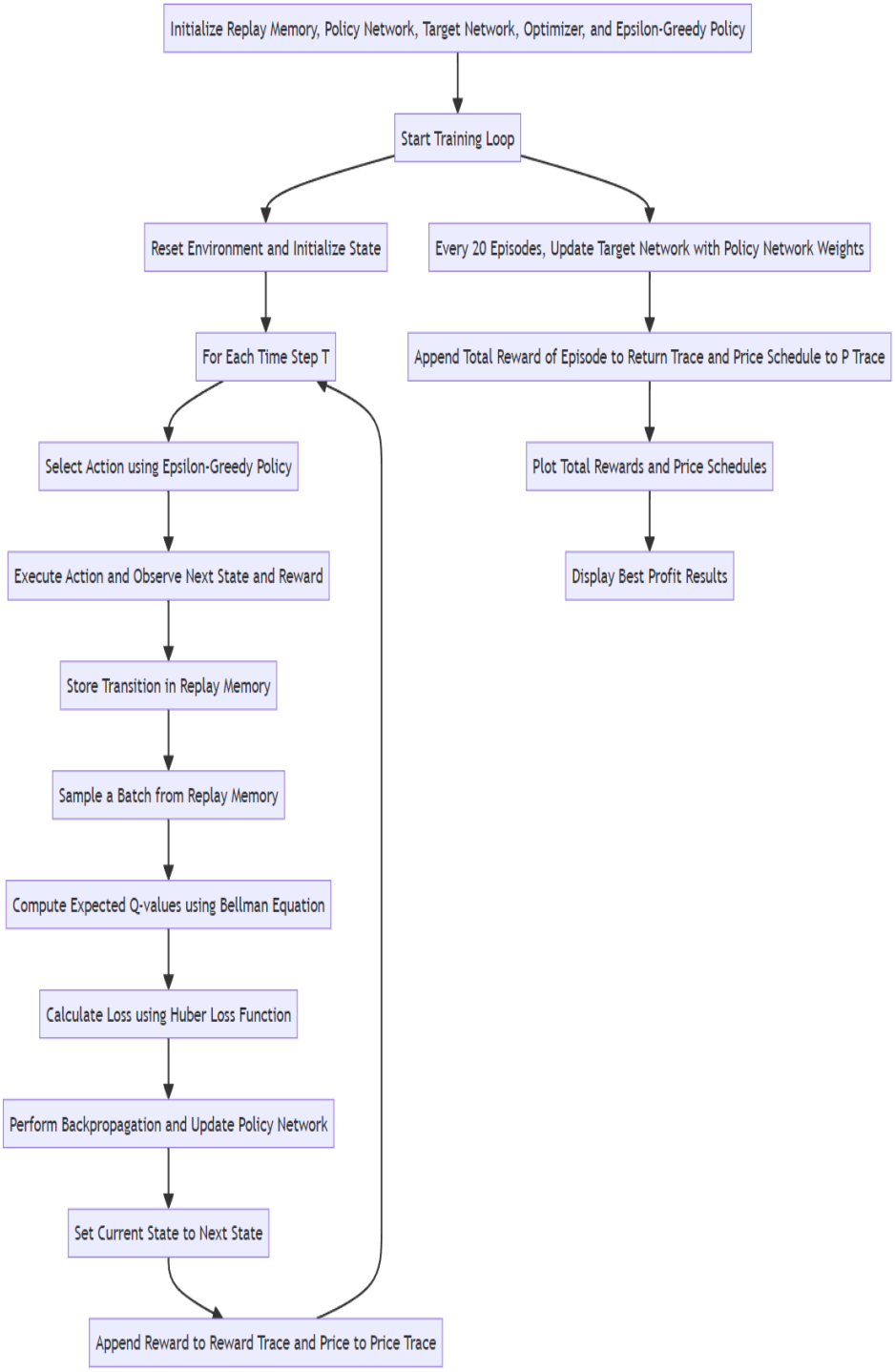

Figure 3.6: Flow chart of the implemented DQN algorithm

3.3.3.3.2 SARSA implementation

The SARSA algorithm is another reinforcement learning approach that updates the Q-values based on the action actually taken by the agent. The general SARSA algorithm working process is now given in Pseudocode form below-

```
Pseudocode:
Initialize policy network with random weights
Initialize replay memory
FOR each episode in num_episodes:
    Initialize state
    Select initial action using epsilon-greedy policy
    FOR each time step t in T:
        Execute action and observe reward and next state
        Select next action using epsilon-greedy policy
        Store transition in replay memory
        Sample random batch from replay memory
        Compute target Q-values using policy network
        Compute loss and update policy network
```

SARSA (State-Action-Reward-State-Action) is an on-policy reinforcement learning algorithm. This means that it updates its Q-values based on the action actually taken by the agent, as opposed to Q-learning which updates based on the optimal action from the next state. A detailed breakdown of the SARSA policy is now give below. The descriptions of the factors that are quite different from DQN is stated here.

Q-Values (Quality Values): In SARSA, Q-values represent the expected future rewards of taking a specific action in a given state, and then following the policy thereafter. Bellman equation for Q learning in SARSA is the same as DQN, which provides a recursive relationship for the Q-values is:

$$\boldsymbol{Q(s,a) \leftarrow Q(s,a) + \alpha\big(r + \gamma Q(s',a') - Q(s,a)\big)} \quad (3.8)$$

But the update rule of saras is different. The SARSA update rule modifies the Q-value of a state-action pair based on the observed reward and the Q-value of the next state-action pair:

$$\boldsymbol{y_j = r_j + \gamma Q(s_{j+1}, a_{j+1}; \theta)} \quad (3.9)$$

where:

r_j: The reward received after taking action a_j in state s_j.

γ: The discount factor.

$Q(s_{j+1}, a_{j+1}; \theta)$The predicted Q-value for the next state s_{j+1} and the next action a_{j+1}, using the current network parameters θ.

Policy: In SARSA, the policy used to select actions is the same as the policy being learned. This makes SARSA an on-policy method. The epsilon-greedy policy is commonly used, where the agent selects a random action with a probability of ϵ\epsilon$\epsilon$ and the best-known action with a probability of $1-\epsilon$. In the SARSA algorithm implemented with a neural network (like DQN but for SARSA), the loss function is used to train the network to predict Q-values more accurately.

Predicted Q-Value: The neural network takes the current state and outputs Q-values for all possible actions.

Target Q-Value: The target Q-value is calculated using the SARSA update rule. It considers the immediate reward and the Q-value of the next state-action pair.

The algorithm for the implemented algorithm for the specified problem specified in this research is now given below in algorithm 2.

Algorithm 2: Algorithm for Implemented SARSA Algorithm

Initialize replay memory D to capacity N.

Initialize policy network Q with weights θ.

Initialize optimizer (Adam) with learning rate 0.005.

Initialize epsilon-greedy policy with starting, ending, and decay values.

For episode = 1 to M:

Initialize state s as a zero vector of size 2T

Initialize reward_trace and price_schedule.

Select initial action a_t using policy network and epsilon-greedy policy.

For t=1 to T:

Execute action a_t in the environment.

Observe next state s_{t+1} and reward r_t

Select next action a_{t+1} using policy network and epsilon-greedy policy.

Store transition (s_t, a_t, s_{t+1}, r_t, a_{t+1}) in D.

If (memory D contains more than batch size B):

Sample random mini-batch of transitions (s_j, a_j,s_{j+1},r_j,a_{j+1}) from D

Compute y_j:

$y_j=r_j$ if s_{j+1} is terminal.

Otherwise, $y_j = r_j + \gamma Q(s_{j+1}, a_{j+1}; \theta)$

Perform gradient descent step on $y_j - Q(s_j, a_j; \theta)^2$ with respect to θ.

Set $s_t=s_{t+1}$

Set $a_t=a_{t+1}$

Append reward r_t to reward_trace.

Append action a_t to price_schedule.

Plot total rewards for each episode with moving average and standard deviation.

Plot average price schedules for the episodes.

Display best profit results from the training.

The flowchart of implemented SARSA algorithm is now given below in Figure 3.7-

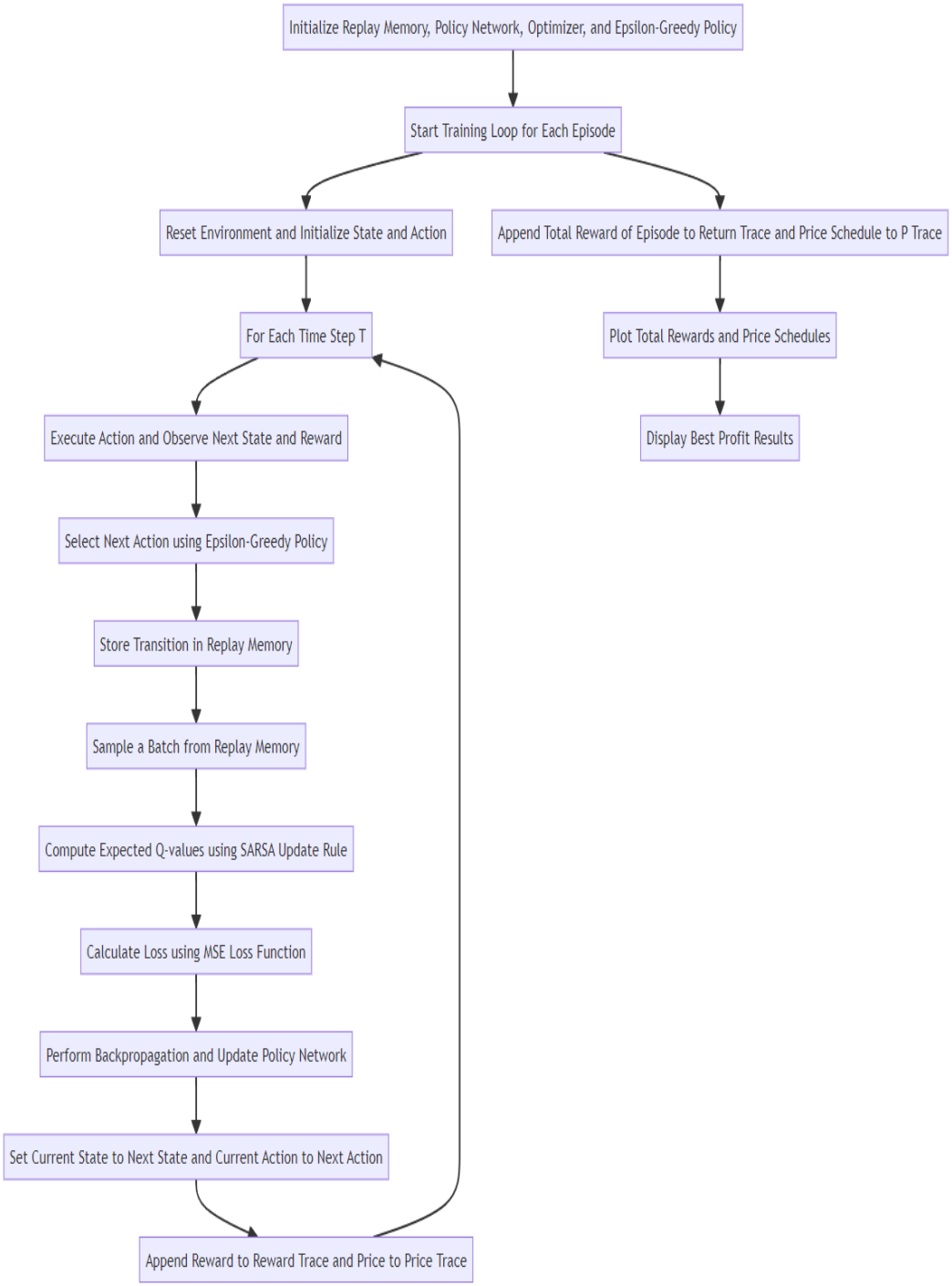

Figure 3.7: Flowchart of implemented SARSA algorithm

3.4 Conclusion

In conclusion, the careful study of different algorithms has formed a strong basis for supply chain optimization. By combining advanced models and precise data processing, important information regarding the dataset used in this research has been obtained. This information is vital in implementing the machine learning algorithm for supply chain risk optimization. For the implementation of deep reinforcement learning algorithms, both the DQN and SARSA algorithm have been discussed in detail. All the practical implementations of the pricing optimization problem that are solved have been discussed. This chapter gives the reader a detailed idea about how one can apply ML and DRL in supply chain optimization process.

Chapter 04

Results

4.1 Introduction

This chapter presents an in-depth comparative analysis of the outcome of the implemented model. The performance of the applied machine learning models is first thoroughly compared in-depth in this section. Confusion metrics and several performance factors will be utilized to evaluate the model's efficacy in resolving the thesis's classification issue. In the later part of the chapter, the results obtained from the application of the deep reinforcement learning algorithm is analyzed. Both the results obtained by implementing DQN and SARSA are analyzed and compared.

4.2 Results obtained for supply chain risk optimization problem

In this part, the results that are obtained by implementing the machine learning algorithms for solving supply chain risk optimization are discussed in detail. First, the confusion matrix of each of the algorithm implemented is discussed and how each result is obtained by implementing different types of algorithms is briefly explained. In the later part of this section, the comparative performance is analyzed in brief.

4.2.1 Confusion matrix

By grouping the performance of a classification model, a confusion matrix offers a brief synopsis by categorizing predictions into True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). This evaluation tool helps assess the model's accuracy across various classes, offering insights for enhancement.

In the data modelling phase of the work the required results are obtained. As the problem is a classification problem all the classification algorithms are implemented. Through careful screening 13 different algorithms are selected for comparison of results and one hybrid algorithm is also implemented. The hybridization is done with

the combination of Multi-Layer Perceptron (MLP) Classifier, Random Forest and Extra Trees classifier.

MLP Classifier is good at capturing complex relationships in the data, while Random Forest and Extra Trees Classifier excel at handling different aspects of the data through ensemble learning and feature randomness. By combining these models and training a meta-model on their predictions, one can effectively leverage their collective strengths to make accurate and precise forecasts. This can result in better performance across multiple evaluation metrics such as accuracy, F1 score, and recall. Equation 4.1,4.2 and 4.3 below indicates accuracy, recall and F1 score.

The accuracy can be determined using Eqn. (4.1)

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \tag{4.1}$$

The recall can be determined by Eqn. (4.2)

$$\text{Recall} = \frac{TP}{TP+FN} \tag{4.2}$$

The F1 score can be determined by Eqn. (4.3)

$$\text{F1 score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4.3}$$

4.2.1.1 Logistic Regression

Logistic regression excels in binary classification by offering clear interpretability, simplicity for foundational analysis, and efficiency with linearly separable data [63]. Figure 4.1 represents the confusion matrix of logistic regression applied in the test dataset for predicting fraud status and late delivery status.

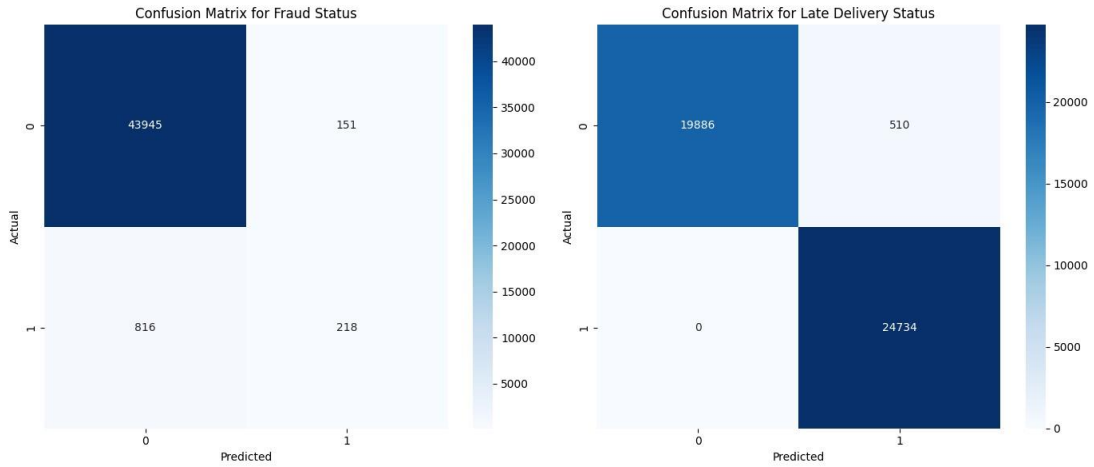

Figure 4.1: Confusion matrix of Logistic Regression model

4.2.1.2 Gaussian Naive Bayes

Gaussian Naive Bayes excels in straightforwardness and efficiency, handling high-dimensional and small datasets well, with a probabilistic framework that clarifies feature contributions [64]. Figure 4.2 represents the confusion matrix of Gaussian naïve Bayes applied in the test dataset for predicting fraud status and late delivery status

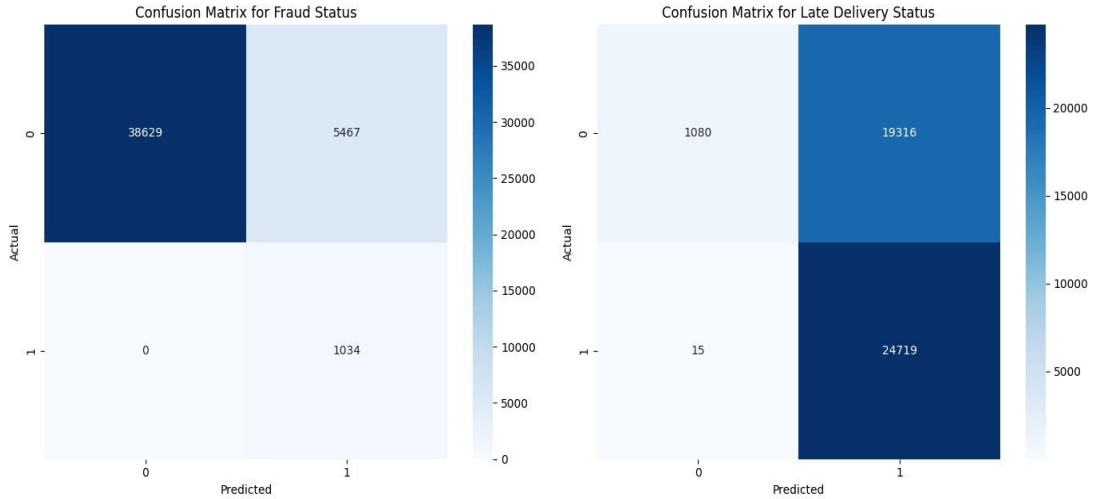

Figure 4.2: Confusion matrix of Gaussian Naïve Bayes model

4.2.1.3 Support Vector Machine

Support Vector Machines (SVM) excel in handling high-dimensional data and reducing overfitting, with kernel functions enhancing their ability to model complex patterns [65]. Figure 4.3 represents the confusion matrix of support vector machine applied in the test dataset for predicting fraud status and late delivery status.

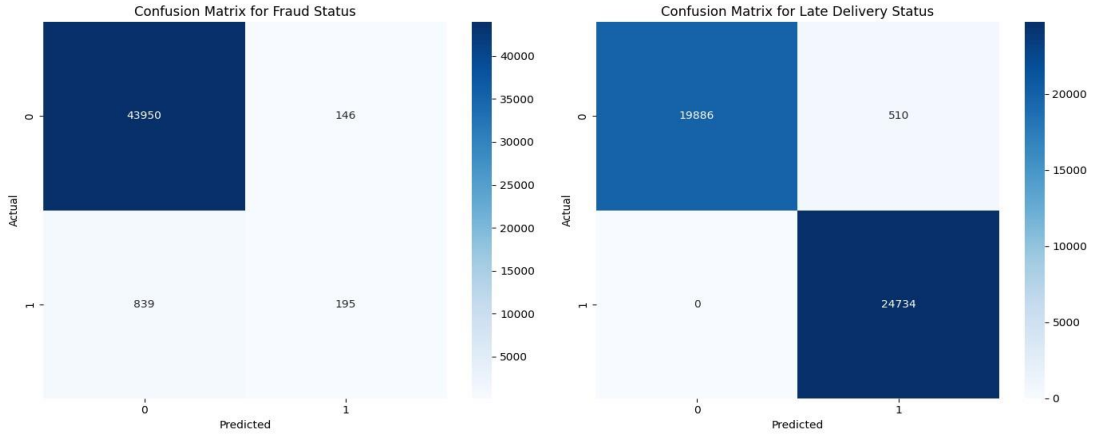

Figure 4.3: Confusion matrix of Support Vector Machine model

4.2.1.4 K-nearest Neighbors

K-nearest neighbors (KNN) is simple and interpretable, adapt to different data distributions, and allow easy integration of new data with strong performance and minimal tuning [66]. The confusion matrix of K-nearest neighbors is given in Figure 4.4.

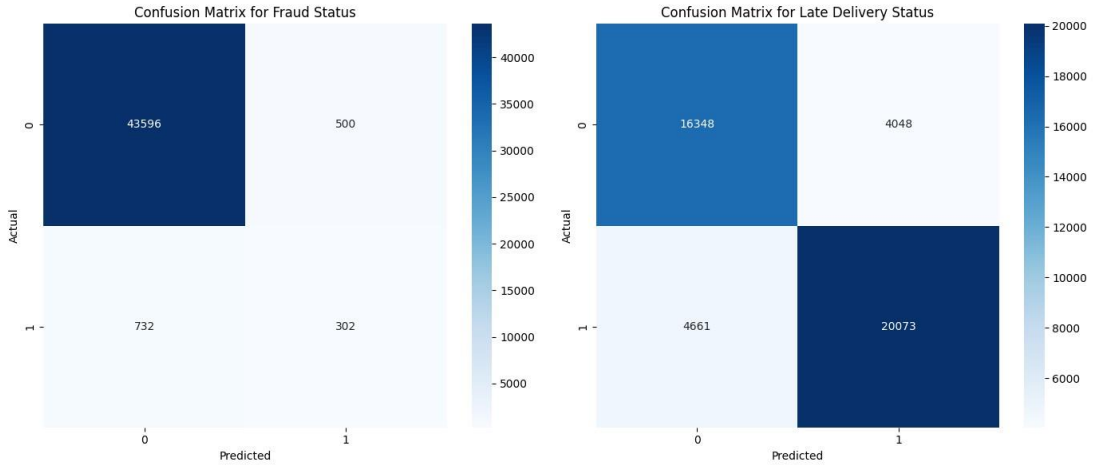

Figure 4.4: Confusion matrix of K-nearest Neighbors model

4.2.1.5 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) reduces dimensionality while preserving class distinctions, enhancing efficiency and maximizing the ratio of between-class to within-class variance for optimal separability [67]. The confusion matrix of Linear Discriminant Analysis is given in Figure 4.5.

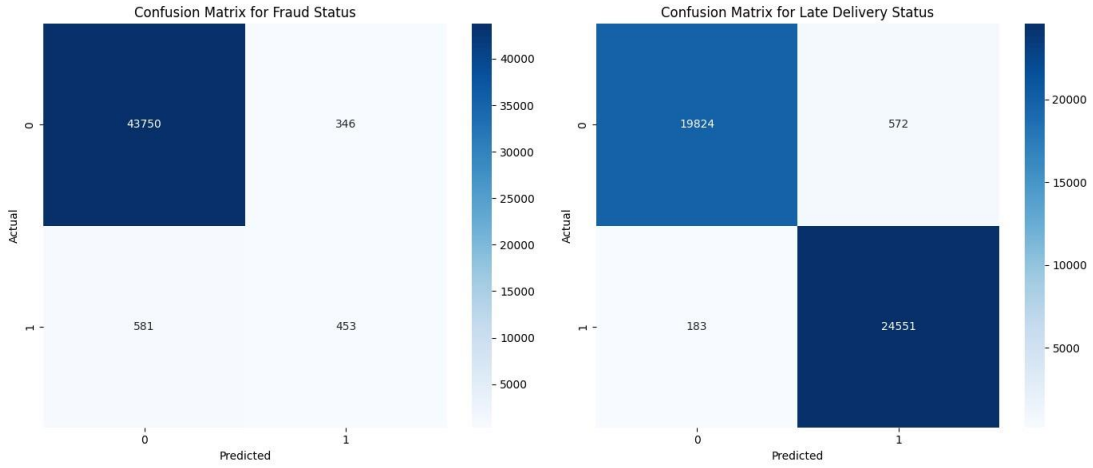

Figure 4.5: Confusion matrix of Linear Discriminant Analysis model

4.2.1.6 Random Forest Classifier

Random Forest handles high-dimensional data and reduces overfitting by averaging predictions from multiple trees, enhancing generalization and performing internal feature selection to boost predictive accuracy [68]. The confusion matrix of Random Forest Classifier is given in Figure 4.6.

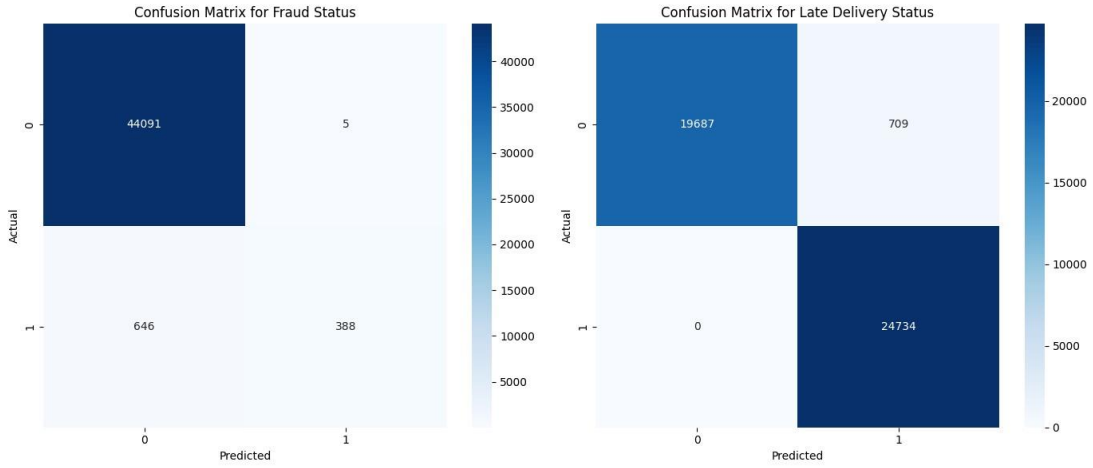

Figure 4.6: Confusion matrix of Random Forest Classifier model

4.2.1.7 Extra Trees Classifier

Extra Trees Classifier reduces overfitting by using random feature selection and averaging tree predictions, enhancing robustness and offering easy implementation with automatic feature selection [69]. The confusion matrix of Extra Trees Classifier is given in Figure 4.7.

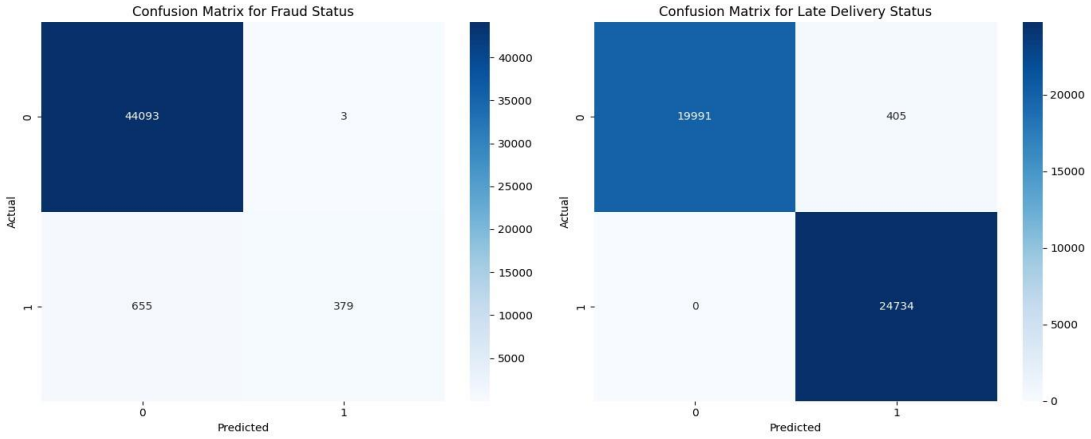

Figure 4.7: Confusion matrix of Extra Trees Classifier model

4.2.1.8 Extreme Gradient Boosting (XGB) Classifier

XGBoost is highly efficient and accurate, featuring built-in regularization, managing missing values, and offering intuitive feature importance for enhanced interpretability [70]. The confusion matrix of Extreme Gradient Boosting Classifier is given in Figure 4.8.

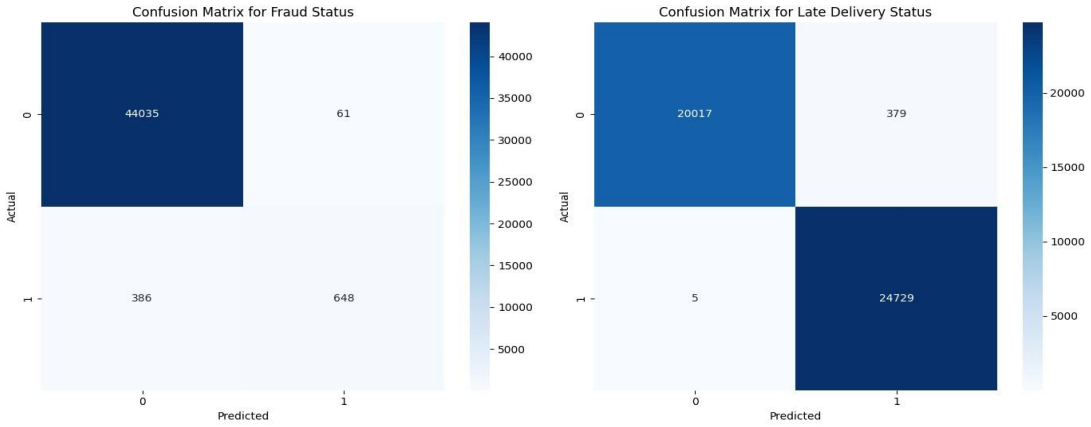

Figure 4.8: Confusion matrix of Extreme Gradient Boosting Classifier model

4.2.1.9 Decision Tree Classifier

Decision Tree Classifier is valued for its interpretability and simplicity, visually mapping decisions, and avoids overfitting with early stopping to ensure balanced, generalizable, and robust models [71]. Figure 4.9 represents the confusion matrix of Decision Tree Classifier applied in the test dataset for predicting fraud status and late delivery status.

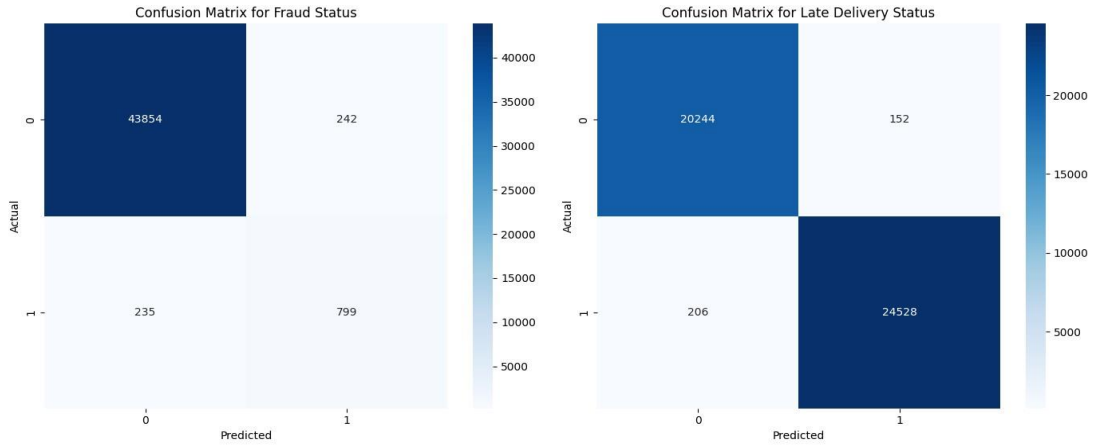

Figure 4.9: Confusion matrix of Decision Tree Classifier model

4.2.1.10 Ada Boost Classifier

AdaBoost Classifier combines weak learners into a strong model, effectively addressing overfitting by emphasizing difficult instances and enhancing interpretability through weighted voting mechanisms [72]. Figure 4.10 represents the confusion matrix of Ada Boost Classifier applied in the test dataset for predicting fraud status and late delivery status.

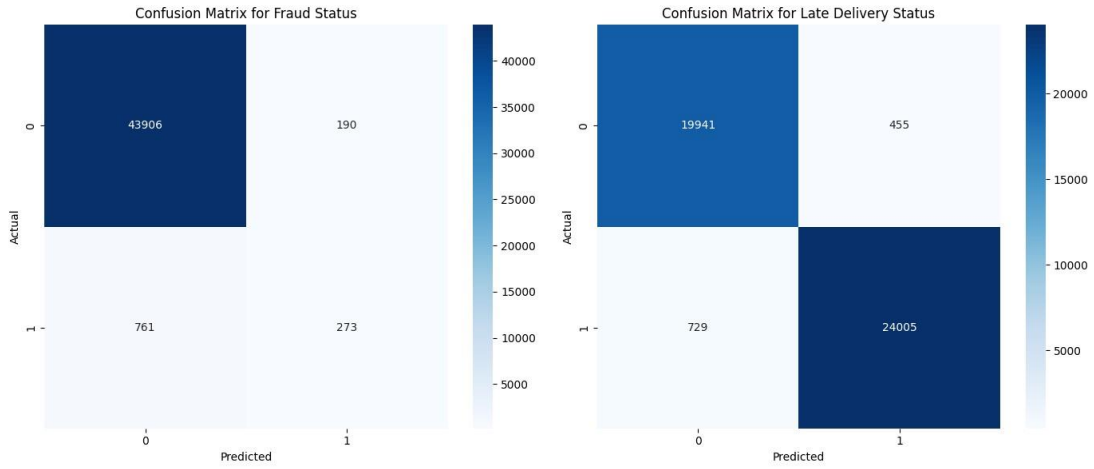

Figure 4.10: Confusion matrix of Ada Boost Classifier model

4.2.1.11 Histogram Gradient Boosting Classifier

Histogram Gradient Boosting Classifier handles complex datasets and reduces overfitting through a sequential approach, refining the model to minimize errors and maintain high accuracy [73]. Figure 4.11 represents the confusion matrix of Histogram Gradient Boosting Classifier applied in the test dataset for predicting fraud status and late delivery status.

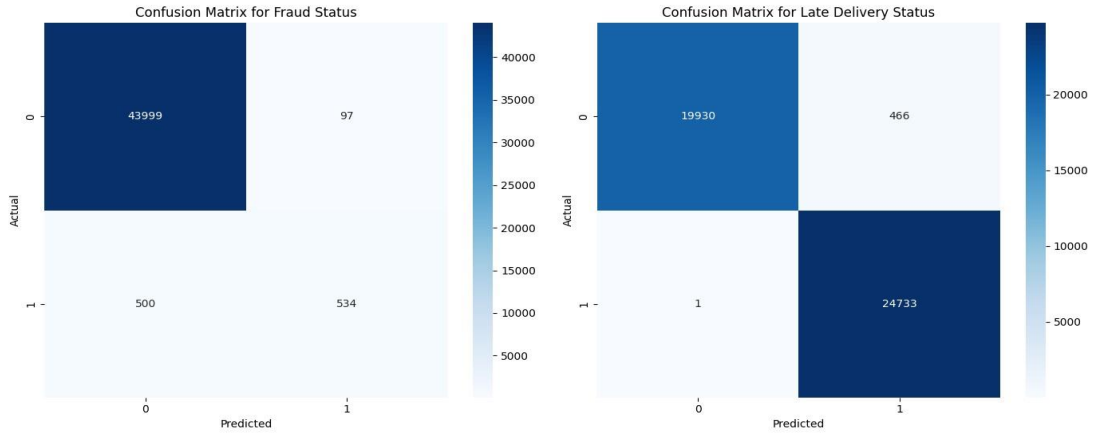

Figure 4.11: Confusion matrix of Histogram Gradient Boosting Classifier model

4.2.1.12 Light GBM Classifier

LightGBM Classifier efficiently handles large datasets with a histogram-based approach and leaf-wise growth, enabling faster training and lower memory usage while managing categorical features effectively [74]. The confusion matrix of the Light GBM Classifier is given in Figure 4.12.

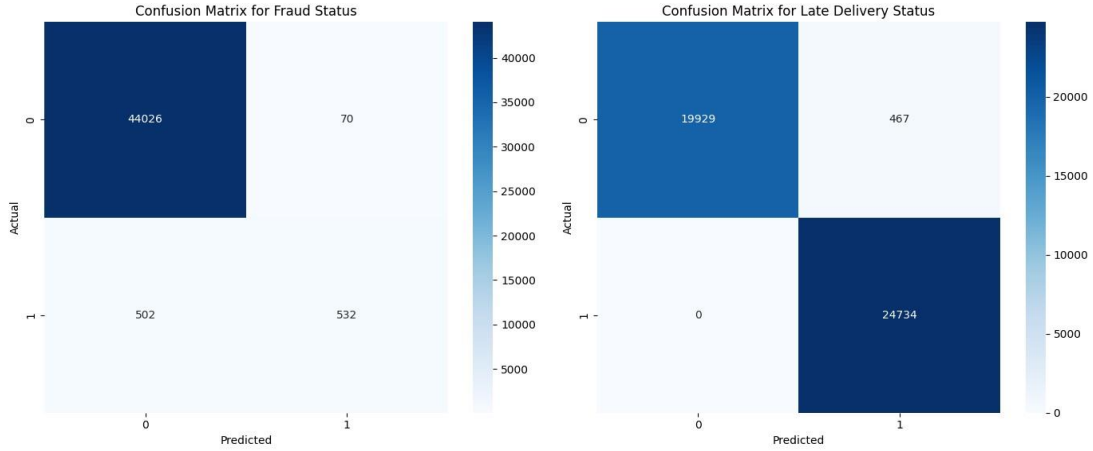

Figure 4.12: Confusion matrix of Light GBM Classifier model

4.2.1.13 Multi-layer Perceptron (MLP) Classifier

Multi-layer Perceptron (MLP) Classifier models complex, non-linear relationships with its layered architecture, capturing patterns while balancing interpretability and predictive power in classification tasks [75]. The confusion matrix of Multi-layer Perceptron (MLP) is given in Figure 4.13.

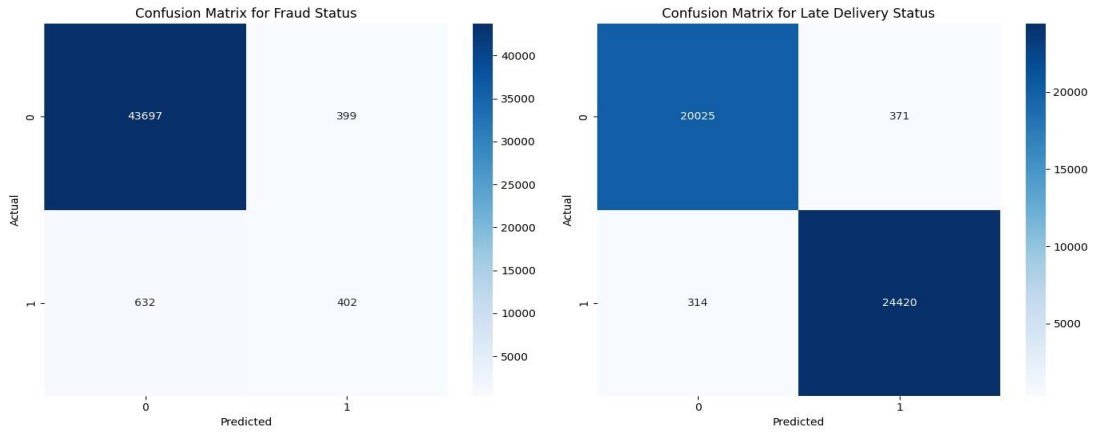

Figure 4.13: Confusion matrix of Multi-layer Perceptron (MLP) Classifier model

4.2.1.14 Hybrid Model 2

A hybrid model combines multiple classifiers to create a powerful ensemble, integrating deep learning for complex pattern recognition with robustness, leading to improved accuracy, reduced complexity, and minimized overfitting risks [76]. The better hybrid model out of the two depicted in this research uses MLP, Random Forest and Extra Trees Classifier algorithms. The confusion matrix of the hybrid model used in this research is given in Figure 4.14.

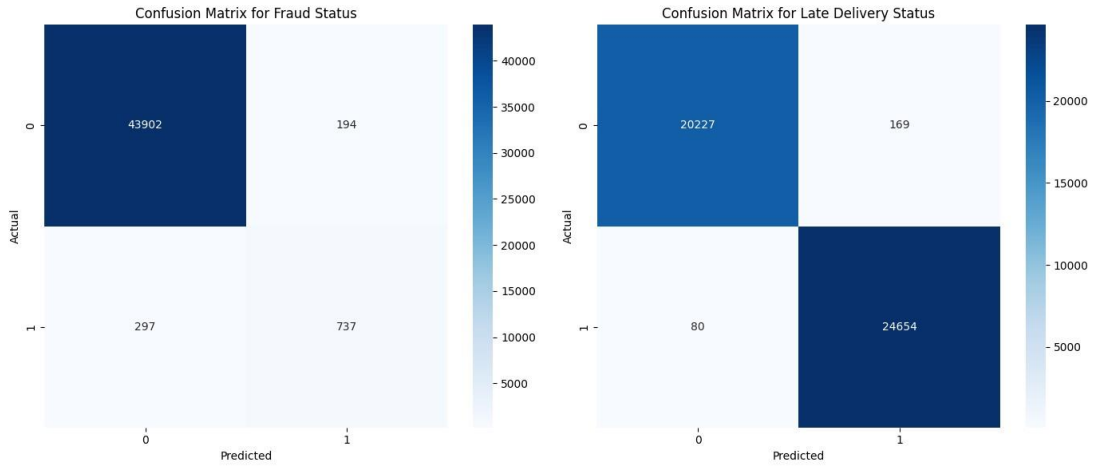

Figure 4.14: Confusion matrix of Hybrid model 2

4.2.2 Comparative Performance Analysis

In this section, a comparative analysis is presented, showcasing the recall, F1-Score, and accuracy metrics of the models employed throughout this thesis.
Now the results obtained for the fraud prediction are presented in Table 4.1.

Table 4.1: Fraud detection and prediction

Model Name	**Accuracy (%)**	**Recall (%)**	**F1 Score (%)**
Logistic Regression	97.857	59.076	31.076
Gaussian Naive Bayes	87.886	15.905	27.445
Support Vector Machines	97.824	57.60	28.633
K nearest Neighbors	97.270	37.655	32.897
Linear Discriminant Analysis	97.945	56.695	49.427
Random Forest Classifier	98.635	98.156	58.038
Extra Trees Classifier	98.541	99.214	53.531
Extreme Gradient Boosting (XGB)Classifier	98.90	90.932	70.990
Decision Tree Classifier	98.912	75.980	76.382
Ada Boost Classifier	97.892	58.963	36.472
Histogram Gradient Boosting Classifier	98.635	84.602	62.393
Light GBM Classifier	98.732	88.372	65.036
Multi-Layer Perceptron (MLP) Classifier	97.719	58.064	33.381
Hybrid Model 1 (Decision Tree + Random Forest + AdaBoost)	98.92	75.03	76.78
Hybrid Model 2(MLP + Random Forest + Extra Trees Classifier)	99.151	89.93	79.286

From Table 4.1, it is seen that all the parameters of the hybrid model 2 performed better than other algorithms. Here accuracy, recall and F1 score are provided. Recall

and F1 score are evaluation metrics used in machine learning, particularly in binary classification tasks.

Recall, also known as sensitivity or true positive rate, measures the proportion of actual positive instances correctly identified by the model. It emphasizes the ability to capture all positive cases without false negatives. The ratio of true positives (TP) to the total of false negatives (FN) is used to compute it. F1 score, on the other hand, is the harmonic mean of precision and recall. It provides a balanced measure of model performance by considering both precision (ability to correctly classify positive instances) and recall. F1 score ranges between 0 and 1, where 1 indicates perfect precision and recall, while 0 indicates poor performance.

In summary, recall focuses on minimizing false negatives, while F1 score considers a balance between precision and recall, which makes it a more comprehensive measure of model accuracy. So, for the classification task in this paper F1 score is more relevant and a good result is obtained in many algorithms. The hybrid model produced a F1 score of 79.286% which is 45.8% more than MLP classifier and 26.2% more the Extra trees classifier. So, the hybridization made a good impact on the results obtained. For fraud detection some of the previous works on DataCo supply chain dataset is presented in Table 4.2 for validation.

Table 4.2: Accuracy comparison of fraud detection with previous studies

Corresponding work	Algorithm	Accuracy
Constante-Nicolalde[39]	Random Forest	81.55%
	R part	76.1%
Lahcen Tamym[40]	ANN	98%
	Decision Tree	99.04%
This research	Hybrid Model 2(MLP + Random Forest + Extra Trees Classifier)	99.151%

The next stage of the work is the late delivery status predictions in the dataset which is presented in Table 4.3.

Table 4.3: Late delivery status prediction

Model Name	Accuracy (%)	Recall (%)	F1 Score (%)
Logistic Regression	98.869	97.979	98.979
Gaussian Naive Bayes	57.165	56.134	71.889
Support Vector Machines	98.869	97.979	98.979
K nearest Neighbours	80.702	83.217	82.173
Linear Discriminant Analysis	98.327	97.723	98.485
Random Forest Classifier	98.428	97.213	98.582
Extra Trees Classifier	99.101	98.388	99.127
Extreme Gradient Boosting (XGB)Classifier	99.166	98.510	99.145
Decision Tree Classifier	99.166	99.331	99.219
Ada Boost Classifier	97.376	98.139	97.593
Hist Gradient Boosting Classifier	98.945	98.111	99.046
Light GBM Classifier	98.965	98.146	99.064
Multi-Layer Perceptron (MLP) Classifier	98.863	97.979	98.973
Hybrid Model 1(MLP + Random Forest + Extra Trees Classifier)	99.151	98.03	99.03
Hybrid Model 2 (MLP + Random Forest + Extra Trees Classifier)	99.452	99.093	99.602

From Table 4.3, it is also seen that in all the parameters our hybrid model 2performed slightly better than other algorithms. Here accuracy, recall and F1 score are really close for all algorithms because of the nature of the dataset. As all the results are pretty much close, cross validation is done for validation of the results obtained. Model used is LGBM Classifier for validation and results are-

Cross-validation accuracy of fraud detection: 0.96 (+/- 0.02)

Cross-validation accuracy of late delivery: 0.99 (+/- 0.01)

So, the company can optimize its decisions more effectively and minimize its risks from the results that are obtained from this research.

4.3 Results obtained for supply chain pricing optimization problem

This part of the chapter provides a comprehensive analysis of the results obtained from implementing and comparing the SARSA (State-Action-Reward-State-Action) and DQN (Deep Q-Network) reinforcement learning algorithms in a price optimization problem. It is required to mention that the results obtained from the traditional constant price optimization method is 8.7 lakhs Tk and for greedy dynamic price optimization it is 9.23 lakhs Tk. The goal of these experiments was to evaluate the efficiency, stability, and profitability of each algorithm by examining their performance in maximizing returns over a series of episodes. Through detailed examination of the data, visualizations, and outcomes, the aim is to draw meaningful insights into how these algorithms behave in different scenarios and under various conditions. This analysis will not only compare the raw performance metrics but also delve into the underlying dynamics that drive the observed results.

4.3.1 DQN algorithm results

The DQN algorithm, which uses a neural network to approximate the Q-value function, demonstrated significant potential in optimizing pricing strategies. The neural network enables the agent to estimate the expected return of different actions from any given state, allowing it to select actions that maximize long-term profit. The results from the DQN implementation showed a clear upward trend in returns, with the algorithm achieving some of the highest profit results observed in the experiments. The best profit result achieved by the DQN algorithm was 10.78 lakhs Tk, with the corresponding optimal pricing schedule being [280, 260, 240, 220, 180, 280, 260, 240, 220, 160] in Tk. This pricing schedule suggests that the algorithm favored maintaining high price points, likely because higher prices led to higher profits under the given demand model. Other notable profit results included 10.75 lakhs Tk with a pricing schedule of [280, 260, 240, 200, 280, 260, 240, 220, 200, 160], and 10.63 lakhs Tk

with a schedule of [280, 260, 240, 220, 200, 280, 260, 220, 200, 160]. These results indicate a consistent pattern where the DQN algorithm prioritized higher price points, adjusting slightly based on the state and previous prices.

Figure 4.15 below represents the output obtained by implementing DQN algorithm-

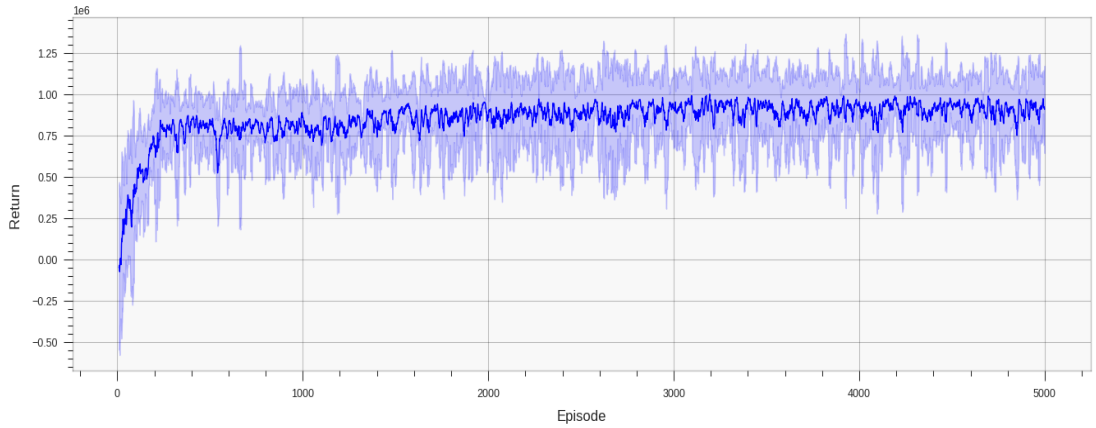

Figure 4.15: DQN algorithm implementation

The learning curve for the DQN algorithm, depicted in Figure 5.1, shows the return over 1000 episodes, with a smoothed average return and standard deviation. The initial episodes exhibited high variability in returns, reflecting the exploration phase driven by the epsilon-greedy policy. However, as the training progressed, the returns stabilized around the midpoint of the episodes, indicating that the agent was successfully learning to exploit profitable pricing strategies. The stabilization in returns towards the latter half of the episodes suggests that the DQN algorithm converged to a policy that consistently generated high profits. The best 5 profit results are now listed below in Table 4.4.

Table 4.4: Best profit results for DQN Algorithm

Best profit Results (in Tk)	Price Schedule (in Tk)
10.61 lakhs	[280, 260, 240, 220, 180, 280, 260, 220, 200, 160]
10.63 lakhs	[280, 260, 240, 220, 200, 280, 260, 220, 200, 160]
10.74 lakhs	[280, 260, 240, 200, 280, 260, 240, 220, 200, 160]
10.75 lakhs	[280, 260, 240, 200, 280, 260, 240, 220, 200, 160]
10.78 lakhs	[280, 260, 240, 220, 180, 280, 260, 240, 220, 160]

4.3.2 SARSA algorithm results

The SARSA algorithm, an on-policy reinforcement learning method, was also implemented to compare its performance with DQN. Unlike DQN, which updates its policy based on the estimated Q-values of all possible actions, SARSA updates its policy based on the actual actions taken. This on-policy approach means that the agent's learning process is closely tied to its exploration strategy, leading to potentially more varied and robust exploration of the action space.

The best profit result achieved by the SARSA algorithm was 9.98 lakhs, with the corresponding optimal pricing schedule being [280, 240, 220, 160, 280, 240, 180, 280, 240, 180]. This result is slightly lower than the best result obtained by DQN, but still demonstrates significant profitability. Other notable profit results included 9.62 lakhs Tk with a pricing schedule of [280, 240, 220, 220, 200, 180, 280, 240, 180, 160], and 9.57 lakhs Tk with a schedule of [280, 240, 220, 160, 280, 240, 180, 280, 240, 240]. The SARSA algorithm exhibited a broader range of pricing strategies compared to DQN, reflecting its more exploratory nature.

Figure 4.16 below represents the output obtained by implementing DQN algorithm-

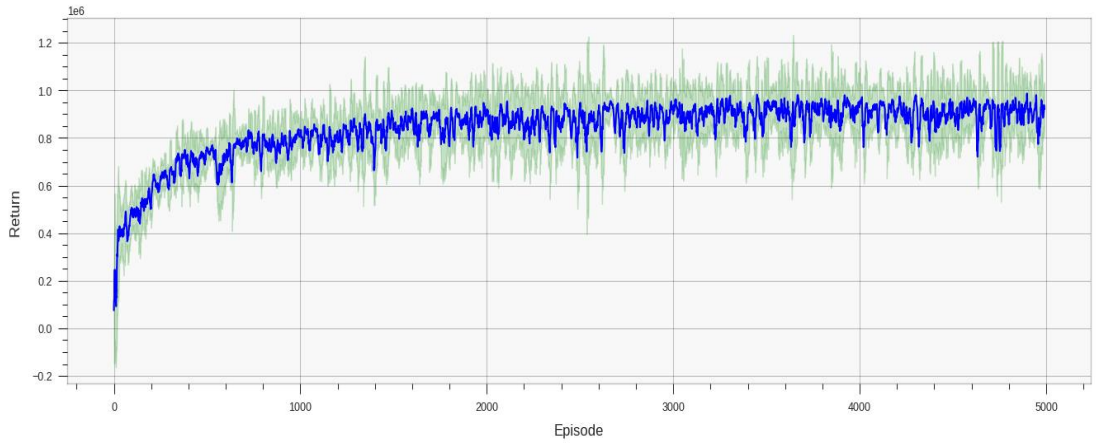

Figure 4.16: SARSA algorithm implementation

The learning curve for the SARSA algorithm, shown in Figure 5.2, also depicts the return over 1000 episodes with a smoothed average return and standard deviation. The curve indicates an initial rise in returns, followed by fluctuations as the agent explores different pricing strategies. The returns stabilized towards the latter part of the episodes, indicating that the agent was converging to an optimal policy. However, the SARSA learning curve exhibited more variability throughout the episodes compared to DQN, highlighting the on-policy nature of SARSA which might be more sensitive to the specific actions taken during learning. Table 4.5 below represents the best 5 profit results.

Table 4.5: Best profit results for SARSA algorithm

Best profit Results (in Tk)	Price Schedule (in Tk)
9.33 lakhs	[240, 240, 240, 200, 180, 280, 240, 220, 160, 260]
9.38 lakhs	[280, 240, 240, 240, 200, 180, 280, 240, 180, 160]
9.57 lakhs	[280, 240, 220, 160, 280, 240, 180, 280, 240, 240]
9.62 lakhs	[280, 240, 220, 220, 200, 180, 280, 240, 180, 160]
9.98 lakhs	[280, 240, 220, 160, 280, 240, 180, 280, 240, 180]

Now, Table 4.6 shows the comparison of the implemented DRL algorithms and traditional price optimization methods.

Table 4.5: Comparison of best profit results

Algorithm Name	Best Profit Results (in Tk)
Constant price optimization	8.71 lakhs
Greedy dynamic price optimization	9.23 lakhs
DQN	10.78 lakhs
SARSA	9.98 lakhs

So, it is clear from the table that DQN algorithm gives 7.4% more profit results than SARSA algorithm and 19% more profit results than constant price optimization. So, this can be concluded that DQN is best performing algorithm.

4.4 Conclusion

This chapter concludes with comprehensive results and a comparison of results. The comparison between various ML algorithms shows that the hybrid algorithm outperformed all other algorithms for supply chain risk optimization. For supply chain pricing policy optimization, from the DRL algorithms, DQN performed better than SARSA. So, the results clearly indicate machine learning can be effectively used in the process of supply chain optimization.

Chapter 05

Conclusion

5.1 Introduction

In this thesis, we delved into the optimization of supply chain processes through the lens of machine learning (ML) and deep reinforcement learning (DRL) techniques. The comprehensive analysis encompassed a variety of approaches and methodologies that demonstrate the potential of these advanced computational techniques to address complex problems inherent in supply chain management.

5.2 Summary of findings

In the supply chain pricing policy optimization, DRL methods, such as Deep Q-Network (DQN) and State-Action-Reward-State-Action (SARSA), show promise in optimizing dynamic pricing strategies within supply chains. By leveraging real-world data, these algorithms have the ability to continuously adapt to market changes, thereby enhancing profitability and operational efficiency.

The practical application of DRL in the Bangladeshi online marketplace for T-shirts underscored its effectiveness in real-world scenarios. The focus on a single product allowed for a detailed analysis and demonstrated the scalability of the DRL framework to other products and markets.

In the supply chain risk optimization ML techniques, including supervised and unsupervised learning, play a crucial role in various aspects of SCM, such as inventory management, risk mitigation, and demand forecasting. For instance, the hybrid Bayesian-optimized Light Gradient-Boosting Machine (LightGBM) model effectively predicts backorder risks, thereby enhancing supply chain robustness and resilience.

The integration of quantum computing with ML presents new frontiers in SCM, offering significant efficiency gains and improved solution quality despite the current infancy of quantum computing technologies.

ML models such as Support Vector Machines (SVM), Bayesian networks, and neural networks have been successfully employed to predict and mitigate financial and operational risks within supply chains. These models enable proactive risk management, thus ensuring continuity and profitability in supply chain operations.

The hybrid model developed in this research outperformed all traditional ML algorithms and all the ensemble algorithms too. The performance matrix indicates its validity and proof. So, this type of hybrid model is very useful and significant in the case of supply chain optimization.

5.3 Future research directions

The findings from this research pave the way for several future research directions that can further enhance the application of ML and DRL in supply chain management:

Integration with Emerging Technologies: The integration of ML with emerging technologies such as the Internet of Things (IoT), blockchain, and edge computing offers promising avenues for research. These technologies can provide richer datasets and more robust frameworks for real-time decision-making and risk management.

Advanced DRL Algorithm Development: Future research should focus on developing more sophisticated DRL algorithms that can handle a wider range of supply chain scenarios and products. Enhancing the scalability and adaptability of these algorithms will be crucial for their broader application in diverse market conditions.

Cross-Disciplinary Approaches: Adopting cross-disciplinary approaches that incorporate insights from fields such as economics, behavioral science, and operations research can provide a more holistic understanding of supply chain dynamics. This can lead to the development of more comprehensive and effective optimization strategies

Data Quality and Management: Ensuring high-quality data is critical for the accuracy of ML models. Future research should explore advanced data preprocessing techniques, robust data management systems, and methods for handling missing or noisy data to improve model performance.

Ethical and Sustainable AI: The ethical implications of AI and ML in SCM, including issues related to data privacy, algorithmic bias, and transparency, should be a key area of focus. Additionally, research should investigate how these technologies

can contribute to sustainable supply chain practices, such as reducing carbon footprints and promoting circular economy principles.

Validation in Diverse Contexts: While the current research has shown promising results in specific contexts, further validation is needed across different industries, regions, and market conditions. This will help to establish the generalizability and robustness of the proposed algorithms and models.

5.4 Concluding remarks

The intersection of ML and DRL with SCM represents a transformative shift in how businesses can optimize their operations, manage risks, and enhance overall efficiency. As the complexity of supply chains continues to grow, the adoption of these advanced computational techniques will become increasingly essential. The findings of this research underscore the significant potential of ML and DRL in addressing the challenges of modern supply chains and provide a foundation for future advancements in this critical field.

By continuing to explore and develop these technologies, researchers and practitioners can contribute to more resilient, efficient, and sustainable supply chain systems that are better equipped to meet the demands of a rapidly evolving global marketplace.

References

[1] Masroor Alam, "Supply Chain Management Practices and Organizational Performance in Manufacturing Industry: SCM and Organizational Performance," South Asian Journal of Social Review, Vol. 1, No. 1, pp. 42-52, 2022.

[2] Sanjoy Kumar Paul, Priyabrata Chowdhury, Ripon Kumar Chakrabortty, Dmitry Ivanov, and Karam Sallam, "A mathematical model for managing the multi-dimensional impacts of the COVID-19 pandemic in supply chain of a high-demand item," Annals of Operations Research, pp. 1-46, 2022.

[3] Vinay Singh, Shiuann-Shuoh Chen, Minal Singhania, Brijesh Nanavati, and Agam Gupta, "How are reinforcement learning and deep learning algorithms used for big data based decision making in financial industries–A review and research agenda," International Journal of Information Management Data Insights, Vol. 2, No. 2, pp. 100094, 2022.

[4] Parshv Chhajer, Manan Shah, and Ameya Kshirsagar, "The applications of artificial neural networks, support vector machines, and long–short term memory for stock market prediction," Decision Analytics Journal, Vol. 2, pp. 100015, 2022.

[5] Vinay Singh, Shiuann-Shuoh Chen, Minal Singhania, Brijesh Nanavati, and Agam Gupta, "How are reinforcement learning and deep learning algorithms used for big data based decision making in financial industries–A review and research agenda," International Journal of Information Management Data Insights, Vol. 2, No. 2, pp. 100094, 2022.

[6] Du Ni, Zhi Xiao, and Ming K. Lim, "A systematic review of the research trends of machine learning in supply chain management," International Journal of Machine Learning and Cybernetics, Vol. 11, pp. 1463-1482, 2020.

[7] Stephen Marsland, "Machine learning: an algorithmic perspective," Chapman and Hall/CRC, 2011.

[8] Hannah Wenzel, Daniel Smit, and Saskia Sardesai, "A literature review on machine learning in supply chain management," In Artificial Intelligence and Digital Transformation in Supply Chain Management: Innovative Approaches for

Supply Chains, Proceedings of the Hamburg International Conference of Logistics (HICL), Vol. 27, pp. 413-441, Berlin: epubli GmbH, 2019.

[9] Sanjeev Gupta, Michael Keen, Alpa Shah, and Genevieve Verdier, "Digital revolutions in public finance," International Monetary Fund, 2017.

[10] Tom Michael Mitchell, "Machine learning," Vol. 1, New York: McGraw-Hill, 2007.

[11] Geetika Sharma, and Chander Prabha, "Applications of Machine Learning in Cancer Prediction and Prognosis," In Cancer Prediction for Industrial IoT 4.0: A Machine Learning Perspective, pp. 119-135, Chapman and Hall/CRC, 2021.

[12] Yu-Chen Lo, Stefano E. Rensi, Wen Torng, and Russ B. Altman, "Machine learning in chemo informatics and drug discovery," Drug Discovery Today, Vol. 23, No. 8, pp. 1538-1546, 2018.

[13] Priyanka Govender, Stephen Gbenga Fashoto, Leah Maharaj, Matthew A. Adeleke, Elliot Mbunge, Jeremiah Olamijuwon, Boluwaji Akinnuwesi, and Moses Okpeku, "The application of machine learning to predict genetic relatedness using human mtDNA hypervariable region I sequences," PLoS ONE, Vol. 17, No. 2, 2022.

[14] Noucaiba Sbai, Loubna Benabbou, and Abdelaziz Berrado, "Multi-echelon Inventory System Selection: Case of Distribution Systems," International Journal of Supply and Operations Management, Vol. 9, No. 1, pp. 108-125, 2022.

[15] Exforsys Inc., "Supply Chain Optimization," 3 September 2007, Retrieved 22 February 2024.

[16] Ewout Reitsma, Per Hilletofth, and Eva Johansson, "Supply chain design during product development: a systematic literature review," Production Planning & Control, Vol. 34, No. 1, pp. 1-18, 2023.

[17] Abeer Aljohani, "Predictive Analytics and Machine Learning for Real-Time Supply Chain Risk Mitigation and Agility," Sustainability, Vol. 15, No. 20, pp. 15088, 2023.

[18] Rabia Musheer Aziz, Rajul Mahto, Kartik Goel, Aryan Das, Pavan Kumar, and Akash Saxena, "Modified genetic algorithm with deep learning for fraud transactions of ethereum smart contract," Applied Sciences, Vol. 13, No. 2, pp. 697, 2023.

[19] Richard S. Sutton, and Andrew G. Barto, "Reinforcement Learning: An Introduction," 2nd ed., Cambridge, MA, USA: MIT Press, 2018.

[20] Yuxi Li, "Deep reinforcement learning: An overview," arXiv preprint arXiv:1701.07274, 2017.

[21] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller, "Human-level control through deep reinforcement learning," Nature, Vol. 518, No. 7540, pp. 529-533, 2015.

[22] Gavin A. Rummery, and Mahesan Niranjan, "On-line Q-learning using connectionist systems," Technical Report CUED/F-INFENG/TR 166, Cambridge Univ., Cambridge, UK, 1994.

[23] Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver, "Rainbow: Combining improvements in deep reinforcement learning," In Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI), New Orleans, LA, USA, pp. 3215-3222, 2018.

[24] Ruida Zhao, Xiaoshan Wang, and Steven E. Shreve, "Optimal dynamic pricing for inventory control under competition," Manufacturing & Service Operations Management, Vol. 22, No. 1, pp. 1-20, 2020.

[25] Md. Anisur Rahman Islam, Md. Shahinul Islam, Md. Nazrul Islam, and Md. Nasir Uddin Islam, "E-commerce in Bangladesh: Status and potential," In Proceedings of the 3rd International Conference on Computer Communication and Information Technology (C3IT), Hooghly, India, pp. 1-6, 2021.

[26] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G. Bellemare, and Joelle Pineau, "An introduction to deep reinforcement learning," Foundations and Trends in Machine Learning, Vol. 11, No. 3-4, pp. 219-354, 2018.

[27] Konstantina Kalaitzi, Matthaios Matopoulos, Michael A. Bourlakis, and Wendy Tate, "Supply chain strategies in an era of natural resource scarcity," International Journal of Operations & Production Management, Vol. 39, No. 10, pp. 1195-1225, 2019.

[28] Chee Yew Wong, Sakun Boon-Itt, and Christina W. Y. Wong, "The contingency effects of environmental uncertainty on the relationship between supply chain

integration and operational performance," Journal of Operations Management, Vol. 29, No. 6, pp. 604-615, 2011.

[29] Yang Lei, Hou Qiaoming, and Zhao Tong, "Research on Supply Chain Financial Risk Prevention Based on Machine Learning," Computational Intelligence and Neuroscience, Vol. 2023, pp. 1-15, 2023.

[30] T. Nguyen, and Thi-Lich Nghiem, "Predicting Risks for Supply Chain Management Networks with Machine Learning Algorithm," Journal of Trade Science, Vol. 11, No. 1, pp. 23-34, 2023.

[31] Shehu Sani, Han Xia, Jelena Milisavljevic-Syed, and Konstantinos Salonitis, "Supply Chain 4.0: A Machine Learning-Based Bayesian-Optimized LightGBM Model for Predicting Supply Chain Risk," Machines, Vol. 11, No. 9, pp. 1-15, 2023.

[32] Vignesh M, Senthil Prabhu S, Prabhu K, and R. M. Subramanian, "Hybrid Quantum-Enhanced Machine Learning for Supply Chain Optimization," International Journal for Research in Applied Science and Engineering Technology, Vol. 8, No. 6, pp. 42-53, 2023.

[33] Bo Zhang, Wen Jun Tan, Wentong Cai, and Allan N. Zhang, "Multi-agent Reinforcement Learning for Improving Supply Chain Visibility in Inventory Management," 2023 IEEE/ACM 27th International Symposium on Distributed Simulation and Real Time Applications (DS-RT), pp. 117-118, 2023.

[34] Meike Schroeder, and Sebastian Lodemann, "A Systematic Investigation of the Integration of Machine Learning into Supply Chain Risk Management," Logistics, Vol. 5, No. 3, pp. 1-20, 2021.

[35] Abeer Aljohani, "Predictive Analytics and Machine Learning for Real-Time Supply Chain Risk Mitigation and Agility," Sustainability, Vol. 15, No. 10, pp. 1-25, 2023.

[36] Lei Li, Shaojun Ma, Xu Han, Chundong Zheng, and Di Wang, "Data-driven online service supply chain: a demand-side and supply-side perspective," Journal of Enterprise Information Management, Vol. 34, No. 1, pp. 365-381, 2021.

[37] Mohammad Rishehchi Fayyaz, Mohammad R. Rasouli, and Babak Amiri, "A data-driven and network-aware approach for credit risk prediction in supply chain

finance," Industrial Management & Data Systems, Vol. 121, No. 4, pp. 785-808, 2021.

[38] Chaoliang Han, and Qi Zhang, "Optimization of supply chain efficiency management based on machine learning and neural network," Neural Computing and Applications, Vol. 33, pp. 1419-1433, 2021.

[39] Fabián-Vinicio Constante-Nicolalde, Paulo Guerra-Terán, and Jorge-Luis Pérez-Medina, "Fraud prediction in smart supply chains using machine learning techniques," In International Conference on Applied Technologies, pp. 145-159, Cham: Springer International Publishing, 2019.

[40] Lahcen Tamym, AhmedNait Sidi Moh, Lyes Benyoucef, Moulay Driss El Ouadghiri, “Goods and Activities Tracking Through Supply Chain Network Using Machine Learning Models”. IFIP InternationalConference on Advances in Production Management Systems, Nantes, France. pp.3-12, Sep 2021.

[41] Reza R. Afshar, Jay Rhuggenaath, Yang Zhang, and Uzay Kaymak, "An automated deep reinforcement learning pipeline for dynamic pricing," IEEE Transactions on Artificial Intelligence, Vol. 4, No. 3, pp. 428-437, 2022.

[42] Seyed Abdol Hosseini, Majid Ghorbani, Alireza A. Oghazian, and Rahim Rostami, "Joint pricing and inventory management in a competitive market using reinforcement learning: a combination of the agent-based and simulation-optimization approaches," International Journal of Management Science and Engineering Management, Vol. 18, No. 2, pp. 1-15, 2023.

[43] Jin Song, and Chen Zhang, "Towards optimal pricing of demand response - A nonparametric constrained policy optimization approach," IEEE Transactions on Smart Grid, Vol. 12, No. 1, pp. 238-249, 2023.

[44] Inderpal Singh, "Dynamic pricing using reinforcement learning in the hospitality industry," IEEE Transactions on Artificial Intelligence, Vol. 3, No. 2, pp. 178-185, 2022.

[45] Davide Chiumera, "Deep reinforcement learning for quantitative finance: Time series forecasting using proximal policy optimization," Journal of Finance and Data Science, Vol. 4, No. 1, pp. 50-60, 2022.

[46] Yajun Hu, Deyi Liang, Qi Chen, and Yijun Liang, "Distributed dynamic pricing strategy based on deep reinforcement learning approach in a presale mechanism," Sustainable Blockchain and Computer Systems, Vol. 1, No. 2, pp. 210-225, 2023.

[47] Jian Liu, Yizhong Zhu, Xiang Wang, Yali Dai, and Xiaole Wang, "Dynamic pricing on e-commerce platform with deep reinforcement learning: A field experiment," E-commerce Research and Applications, Vol. 10, No. 4, pp. 456-472, 2021.

[48] Matthew Huber, and Rahul Vadhera, "Learning policies for neural network architecture optimization using reinforcement learning," In FLAIRS Conference Proceedings, Vol. 34, No. 5, pp. 198-207, 2023.

[49] Toshiyuki Demizu, Yan Fang, and Haoyi Ma, "Inventory management of new products in retailers using model-based deep reinforcement learning," Expert Systems with Applications, Vol. 229, pp. 1-20, 2023.

[50] Lingyang Yang, and Aishia Maxwell, "Information retrieval and optimization in distribution and logistics management using deep reinforcement learning," International Journal of Information Systems and Supply Chain Management, Vol. 16, No. 3, pp. 120-135, 2023.

[51] Guanghui Wu, Manuel Ángel de Castro Sáez, and Martín Martínez, "Distributional reinforcement learning for inventory management in multi-echelon supply chains," Digital Chemical Engineering, Vol. 3, pp. 1-15, 2023.

[52] Devan S. Kurian, Vini Mathai, Prabha A. R., and John G., "Deep reinforcement learning-based ordering mechanism for performance optimization in multi-echelon supply chains," Applied Stochastic Models in Business and Industry, Vol. 38, No. 2, pp. 150-165, 2022.

[53] Qian Zhou, Yanan Yao, and Shengqi Fu, "Deep reinforcement learning approach for solving joint pricing and inventory problem with reference price effects," Expert Systems with Applications, Vol. 182, pp. 1-25, 2022.

[54] Rui Wang, Xueming Guo, and Qing Li, "Solving a joint pricing and inventory control problem for perishables via deep reinforcement learning," International Journal of Production Research, Vol. 59, No. 6, pp. 1821-1840, 2021.

[55] Dandan Qiu, Yanchao Ye, Dimitrios Papadaskalopoulos, and Goran Strbac, "A deep reinforcement learning method for pricing electric vehicles with discrete

charging levels," IEEE Transactions on Smart Grid, Vol. 11, No. 4, pp. 3202-3212, 2020.

[56] Pei-Ying Chou, Wen-Yi Chen, Chia-Yu Wang, Ruei-Hao Hwang, and Wei-Tsong Chen, "Pricing-based deep reinforcement learning for live video streaming with joint user association and resource management in mobile edge computing," IEEE Transactions on Wireless Communications, Vol. 21, No. 6, pp. 4523-4535, 2022.

[57] Chencheng Chen, Feng Yuan, Di Ma, and Jian Zhang, "Spatial-temporal pricing for ride-sourcing platform with reinforcement learning," Transportation Research Part C: Emerging Technologies, Vol. 130, pp. 103187, 2021.

[58] Seung Lee, and Dong-Hoon Cho, "Dynamic pricing and energy management for profit maximization in multiple smart electric vehicle charging stations: A privacy-preserving deep reinforcement learning approach," Applied Energy, Vol. 304, pp. 117678, 2021.

[59] Eduardo J. Salazar, Maria Jiménez, and Eduardo Soler, "Reinforcement learning-based pricing and incentive strategy for demand response in smart grids," Energies, Vol. 16, No. 2, pp. 439, 2023.

[60] Anton Kastius, and Robert Schäfer, "Dynamic pricing under competition using reinforcement learning," Journal of Revenue and Pricing Management, Vol. 21, No. 3, pp. 230-245, 2021.

[61] Angel Fraija, Kodjo Agbossou, Nesma Henao, Koffi Kelouwani, and Maxime Fournier, "A discount-based time-of-use electricity pricing strategy for demand response with minimum information using reinforcement learning," IEEE Access, Vol. 10, pp. 85649-85660, 2022.

[62] Fabian Constante, Fernando Silva, and António Pereira, "DataCo SMART SUPPLY CHAIN FOR BIG DATA ANALYSIS," Mendeley Data, Vol. 5, 2019.

[63] Stephen R. Olayinka, and Stan H. Kost, "Using logistic regression model selection towards interpretable machine learning in mineral prospectivity modeling," Geochemistry, 2021.

[64] Nathan S. Patel, Aaron D. Johnson, Rakesh K. Kumar, Kyle A. Adams, George R. Smith, and Karishma A. Mehta Bajaj, "A Bayesian optimized discriminant analysis model for condition monitoring of face milling cutter using vibration

datasets," Journal of Nondestructive Evaluation, Diagnostics and Prognostics of Engineering Systems, Vol. 5, No. 2, 2022.

[65] Bolaji A. Olayinka, Kayode A. Adebayo, Babatunde S. Fadeyi, Sunday G. M. Ekundayo, Oladayo M. Osoba, and Oluwaseun P. Akinnuwesi, "Application of support vector machine algorithm for early differential diagnosis of prostate cancer," Data Science and Management, Vol. 6, No. 1, 2023.

[66] Uthman F. Akinnuwesi, Jiamin D. Zhou, Lei Li, and Lalit M. Khandelwal, "Nearest neighbor machine translation," Computer Science, Computation and Language, 2021.

[67] Lawrence R. Johnson, Ibrahim A. Al-Turjman, and Guanghui M. Xu, "Saliency-based multilabel linear discriminant analysis," IEEE Transactions on Cybernetics, Vol. 52, No. 10, 2021.

[68] Yung-Mei Yang, Chia-Ming Yuan, and Zahra A. Akhiat, "A new noisy random forest based method for feature selection," Cybernetics and Information Technologies, Vol. 21, No. 2, 2021.

[69] Md. Abdul Rahman, and Iftekhar M. Shamsul Hossain, "A novel hybrid feature selection and ensemble-based machine learning approach for botnet detection," Scientific Reports, Vol. 13, No. 1, 2023.

[70] Tarek A.-M. Mohammed, Mohamad F. Hassan, Brian S. Smith, Sami S. Patel, Michael D. Miller, and Marc B. Vaulet, "Gradient boosted trees with individual explanations: an alternative to logistic regression for viability prediction in the first trimester of pregnancy," Computer Methods and Programs in Biomedicine, Vol. 213, 2022.

[71] Vishal Gupta, and Paulo C. E. Costa, "Recent advances in decision trees: An updated survey," Artificial Intelligence Review, Vol. 56, No. 5, 2023.

[72] Saeed S. Ahmed, and Zahra S. Shaikh, "AdaBoost Ensemble Approach with Weak Classifiers for Gear Fault Diagnosis and Prognosis in DC Motors," Applied Sciences, Vol. 14, No. 7, 2024.

[73] Rahul R. Smith, Kiran D. Adams, Alyssa M. Brown, Babatunde B. Singh, Samuel R. Patel, and Zara Z. Shah, "Efficient Machine Learning Techniques to Classifying Cardiovascular Disease and Improve Prediction Analysis," Sukkur IBA Journal of Emerging Technologies, Vol. 6, No. 2, 2023.

[74] Fatima H. Malik, and Ehab S. A. Alzamzami, "Light gradient boosting machine for general sentiment classification on short texts: a comparative evaluation," IEEE Access, Vol. 8, 2020.

[75] Emmanuel E. Oladele, Thomas S. Smith, Samuel E. Olayinka, David O. Isaac, and Ikechukwu P. Okoro, "Application of artificial intelligence in predicting the dynamics of bottom hole pressure for under-balanced drilling: Extra tree compared with feed forward neural network model," Petroleum, Vol. 8, No. 2, 2022.

[76] Arthur A. Chen, Charles Olayinka, and Chris R. Khan, "A review of ensemble learning and data augmentation models for class imbalanced problems: combination, implementation and evaluation," Expert Systems with Applications, 2023.

Appendix-A

Our company has been successfully operating on an internet platform for the past five years. Our primary media sales channel is the internet, particularly Facebook platform, Daraz and Instagram.

The information used in this study is from the beginning of 2021. The earnings from selling various t-shirts over a two-year period beginning in 2021 is around 8.5 lakhs taka. At that time, the T-shirts were sold for different prices. The pricing policy applies a discounted price and an increased price based on the circumstances.

This data is provided solely for research purposes.

Niaz Uddin

Chief Financial Officer,

A" ..d, Bangladesh.

Real World Supply chain i

ORIGINALITY REPORT

19%	15%	16%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	de.overleaf.com Internet Source	1%
2	arxiv.org Internet Source	1%
3	www.mdpi.com Internet Source	1%
4	Michael Hu. "The Art of Reinforcement Learning", Springer Science and Business Media LLC, 2023 Publication	<1%
5	www.researchgate.net Internet Source	<1%
6	www.geeksforgeeks.org Internet Source	<1%
7	eprints-phd.biblio.unitn.it Internet Source	<1%
8	mobt3ath.com Internet Source	<1%