National University of Computer and Emerging Sciences, Lahore

PortFlow

Khadeeja Wasif 21L-5177 BS(CS)
Haleema Haq 21L-5183 BS(CS)
Maryam Akbar 21L-5347 BS(CS)

Supervisor: Dr. Sobia Tariq Javed
Co-Supervisor: Dr. Zareen Alamgir

Final Year Project

May 5, 2025

Anti-Plagiarism Declaration

This is to declare that the above publication was produced under the:

Title: PortFlow

is the sole contribution of the author(s) and no part hereof has been reproduced as it is the basis (cut and paste) that can be considered Plagiarism. All referenced parts have been used to argue the idea and cited properly. I/We will be responsible and liable for any consequence if a violation of this declaration is determined.

Date: 13 March, 2025

Name: Khadeeja Wasif

Signature:

Name: Haleema Haq

Signature:

Name: Maryam Akbar

Signature:

Author's Declaration

This states Authors' declaration that the work presented in the report is their own and has not been submitted/presented previously to any other institution or organization.

Abstract

PortFlow is a cutting-edge AI-powered berth allocation system revolutionizing port operations by leveraging deep reinforcement learning and hybrid AI-heuristic solutions. Dynamically adapting to vessel arrival and changing operating restrictions, the system minimizes vessel waiting time, releases berth operations and optimizes resource usage. Rigorous simulations and performance evaluation demonstrate that the new scheduling engine combined with a simple-to-use interactive dashboard greatly enhances port operators' decision-making. The study demonstrates the pioneering potential of AI-powered solutions in addressing complex maritime logistics challenges and opens the door to scalable, sustainable port management optimization.

Executive Summary

PortFlow is an innovative solution to the centuries-old berth allocation and port management challenges. It is crafted to counter the increasing complexity of global trade and the changing nature of port operations, where heuristic-based scheduling methodologies have proven inadequate. The report introduces a new AI-based solution that uses deep reinforcement learning algorithm in conjunction with hybrid AI-heuristic models to allocate berths in real-time dynamically. With the use of vessel arrival data, congestion strategies and operating limitations, PortFlow aims to reduce waiting times, decrease operating costs and increase overall port efficiency.

The mission of the project is rooted in the necessity to overcome inefficiencies in existing berth allocation systems that have the effect of producing excessive delays, unnecessary fuel consumption and underutilization of assets. PortFlow upgrades the situation by building a dynamic scheduling engine that constantly learns and adapts from real-world port circumstances. A unifying feature of the system is an interactive dashboard, which provides port operators with real-time analysis and predictive recommendations, as well as manual override capability to provide human insight that can augment automated decision-making.

As an integral section of the report, the literature review brings together traditional optimization techniques of genetic algorithms, MILP models and metaheuristics with recent developments in AI and reinforcement learning. This review highlights the gaps in static methods while emphasizing the advantages of adaptive, data-driven approaches. Drawing from this comprehensive academic and practical background, the project illustrates how the application of several optimization techniques creates a powerful and versatile system adaptable to different port settings.

It describes detailed software requirement specifications and system design considerations and accompanies them with report documents. The system architecture is designed to achieve a balance between performance, reliability, security and usability alongside the existing port infrastructures. Functional requirements include AI-driven berth scheduling, dynamic reallocations due to unexpected delays, historical data analysis and effective interface design, which enable seamless achievement of a system designed to meet the operational requirements of current maritime logistics.

The steps taken to gather data, prepare it and apply reinforcement learning to power the scheduling engine are covered in detail in the methodology section. An AI-heuristic hybrid approach is used for more effective decision-making with comprehensive comparisons and evaluations showing it performs well over traditional approaches. This framework is supported by extensive testing and risk assessment, overcoming technical, ethical and practical challenges to ensure the system remains reliable and adaptable

in various real-world situations.

In conclusion, the PortFlow project not only provides a practical means of maximising berth allocation but also advances broader sustainable development goals by promoting efficient resource use, reducing environmental impacts and enhancing global trade flows. The document clearly outlines future developments, including new scalability across different port settings and the integration of real-time vessel tracking. As a result, PortFlow is an innovative effort to revolutionise port operations, offering the maritime logistics industry both immediate operational benefits and long-term potential.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

The growing complexity of world trade and port activities has created great problems in berth allocation and resource utilization. Effective berth allocation is crucial to reducing vessel waiting times, reducing operating costs and enhancing the overall efficiency of a port. The conventional methods of berth allocation have been based on heuristic and metaheuristic optimization methods, yet these are not very effective with dynamic and uncertain real-world port situations.

With the development of artificial intelligence (AI), reinforcement learning (RL) and hybrid AI-based approaches, have surfaced as viable alternatives for berth allocation optimization. These methods make use of real-time information and adaptive learning models to improve scheduling decisions in a dynamic manner. Our project, PortFlow, targets the creation of an AI-based berth allocation system that consolidates ship and port operational information to optimize scheduling efficiency. Through the implementation of machine learning and optimization algorithms, the system will offer port operators better decision-making tools, maximizing berth utilization and congestion reduction.

The aim of this report is to introduce the research, design, implementation and evaluation of our berth allocation system based on AI. The report describes the problem, methodologies used, system architecture, experimental results and future improvements.
This document is organized as follows: chapter 2 describes the project vision. Chapter 3 covers the liter ature reviews and related work. Chapter 4 thoroughly discusses the software requirement specifications (SRS). Chapter 5 proposes the methodology and approach of the system. Chapter 6 addresses high- and low-level diagrams.

1.1 Purpose of this Document

This report discusses the design, implementation and assessment of our deep reinforcement learning-based berth allocation optimization system. Our project identifies inefficiencies in berth scheduling by utilizing RL models like DQN, D3QN etc. and hybrid AI-heuristic models to enhance port efficiency.

By incorporating actual port data, our system optimizes berth allocation dynamically with respect to vessel arrivals, congestion and operational limitations. The hybrid method guarantees adaptive decision-making with reliability. In addition to this, a user friendly dashboard that displays these insights is to be developed.

Our primary research question is:”Can berth allocation by AI be more effective in reducing waiting times of ships and better utilizing port resources than conventional heuristic approaches?” This document presents our methodology, system architecture, implementation, evaluation and future enhancements.

1.2 Intended Audience

This paper is directed towards port authorities, shipping industry professionals and logistics operators who need AI-based berth allocation and port efficiency solutions. It is also applicable to shipping lines and terminal operators interested in optimizing vessel schedules, eliminating congestion and maximizing resource use through automation.

1.3 Definitions, Acronyms and Abbreviations

AI: Artificial Intelligence
RL: Reinforcement Learning
DQN: Deep Q-Network
D3QN: Dueling Double Deep Q-Network

1.4 Conclusion

Thus, this introduction emphasizes increasing berth allocation difficulties and the requirement for AI-optimized optimization in order to enhance port efficiency. It highlights how conventional methods are not efficient for real-time scheduling and changing port conditions hence machine learning and hybrid AI-heuristic models are necessary to optimize berth allocations. This chapter also sets the goal of our project, to create an intelligent berth allocation system that reduces waiting times for vessels, lessens congestion and improves decision-making for port operators. The system will also include a real-time dashboard that gives port authorities visual information about berth usage, vessel schedules and optimization suggestions, allowing for more transparency and control over operations.

Chapter 2 Project Vision

The general aim of PortFlow is to transform berth allocation and port management through the incorporation of AI-based optimization methods into standard scheduling models. Heuristic-based and manual scheduling are the traditional berth allocation schemes that do not effectively deal with dynamic vessel arrival, congestion and operational constraints. These inefficiencies lead to delay, increased fuel consumption and ineffective utilization of resources.

PortFlow will provide adaptive berth scheduling that dynamically responds to real-world variations in port operations through deep reinforcement learning and hybrid AI-heuristic frameworks. The system will be integrated into existing port management operations, with real-time analysis and berth allocation by automated processes through its interactive dashboard. Port authorities and logistics participants will benefit from it through data-driven decision-making, which will enhance operating efficiency, reduce vessel waiting time and reduce environmental footprint.

2.1 Problem Domain Overview

PortFlow intersects port management, artificial intelligence and maritime logistics, addressing berth allocation and port congestion. Classical berth allocation uses predefined scheduling rules and heuristics that cannot handle vessel arrival dynamics, congestion and operational constraints. This inefficiency translates into higher costs, wasted resources and delays.

2.1.1 AI-Driven Berth Allocation Optimization

PortFlow leverages a hybrid approach combining AI-driven optimization with deep reinforcement learning to create an adaptive berth scheduling system. By continuously analyzing real-time data on vessel schedules, berth occupancy, and congestion levels, the system intelligently allocates berths to minimize vessel waiting times and enhance port efficiency. Unlike traditional methods, PortFlow learns from historical patterns and dynamically adjusts to evolving port conditions, enabling smarter and more responsive decision-making. This leads to better resource utilization, reduced congestion, and significant cost savings across port operations.

2.1.2 Interactive Dashboard for Port Management

A key component of PortFlow is its interactive dashboard, which provides real-time insights into berth occupancy, vessel schedules, and AI-generated berth allocation suggestions. This allows port operators to monitor berth availability, consider intelligent scheduling recommendations and make manual adjustments when necessary.

By combining AI-powered optimization with an intuitive user interface, PortFlow enhances operational efficiency, reduces vessel idle time and transforms traditional port management into a more intelligent and automated system.

2.2 Problem Statement

Current berth allocation practices in many ports rely heavily on manual scheduling and static heuristic-based decision-making. These traditional methods are not able to respond effectively to dynamic port environments such as unpredictable vessel arrivals, fluctuating congestion levels and changing berth availability. As a result ports face challenges such as prolonged vessel waiting times, inefficient resource utilization, and increased operational costs. The demand for higher throughput, faster turnaround times and optimized berth usage further highlights the limitations of these outdated systems.

To address these inefficiencies, this project proposes the development of an AI-driven berth allocation system that integrates port data, machine learning techniques and predictive analytics. The system will continuously adapt to evolving port conditions, dynamically optimize berth scheduling decisions and provide an interactive dashboard for operational support. This approach aims to enhance decision-making and minimize delays. It also contributes to a more intelligent and responsive port management framework.

2.3 Problem Elaboration

Berth scheduling is a complex and dynamic procedure that has immediate effect on vessel turnaround time, port efficiency and operational cost. Traditional berth scheduling methods utilize static scheduling policies and manual decision-making, thus being unable to deal with real-time vessel arrival fluctuations, congestion and resource availability. These inefficiencies result in extra delays, poor berth utilization and extra fuel consumption, which amount to economic and environmental issues in maritime logistics.

2.3.1 Uncertainty in Vessel Arrivals and Cargo Handling

One of the biggest problems with berth scheduling is managing unpredictable vessel arrival times. Vessels can arrive earlier or later than expected, upsetting the planned berth assignments and creating idle berths or congestions. Ports must also manage various types of cargoes, as vessels carrying containers, bulk, or liquid cargoes need specialized berth facilities, cranes and equipment for cargo handling. Without an intelligent, data-based scheduling program, berth usage is not efficient, creating congestions and slowing down the overall port operation.

2.3.2 Inefficient Berth Rescheduling and Adaptability

Another serious issue is real-time rescheduling of berths in the event of unforeseen delays. Conventional approaches struggle to reassign berths efficiently when ships are delayed by bad weather, machinery breakdowns, or customs clearance delays. Without an adaptive automated scheduling system, port authorities have to manually update schedules, which results in inconsistent decision-making, waste of resources and inefficiencies in operations.

2.3.3 Lack of Data-Driven Decision-Making

Port operators often lack real-time visibility into berth utilization and scheduling efficiency. Without a centralized, AI-driven dashboard, it is challenging to track berth occupancy, forecast congestion patterns and optimize resource allocation proactively. A more sophisticated system is required to process large volumes of real-time port data, predict potential delays and dynamically rebalance berth assignments while providing clear, actionable insights through an interactive dashboard.

2.4 Goals and Objectives

The AI-based berth allocation system designed in this project is designed to improve efficiency, minimize congestion and maximize resource utilization in port operations. Conventional berth scheduling techniques cannot cope with dynamic changes in ship arrivals, congestion at the port and operating constraints, which results in delay and underused berths. This project uses deep reinforcement learning and hybrid AI-heuristic models to design an adaptive, smart berth scheduling system. The major goals and objectives of this project are stated below.

2.4.1 AI-Powered Berth Optimization

The main objective of this project is to create a berth allocation system capable of dynamic adaptation to actual port conditions in real time. The system will make intelligent berth assignment decisions to achieve optimum efficiency and minimum waiting times by processing data on vessel arrival, berth availability, cargo type and congestion levels. The conventional approaches are unable to manage last-minute adjustments, but the AI-based model will enable adaptive scheduling and real-time berth reallocation based on changing port conditions.

2.4.2 Reduction of Ship Waiting Time and Congestion

One of the principal objectives is minimizing waiting time for vessels through optimal berth allocation and dynamic realignment in response to the occurrence of unexpected deviations such as delayed arrival, extended cargo operation, or unexpected consignments. By minimizing port congestion, the system will minimize turn-around time, minimize fuel usage due to idle ships and maximize oceanic logistics efficiency as a whole.

2.4.3 Intelligent Decision-Making Through Data-Driven Insights

One of the biggest drawbacks of current berth allocation systems is that they do not track or predict real-time, which limits the effectiveness of decisions made by the current system. PortFlow will have AI-driven berth recommendations with a manual override option for port operators. This option will enable port authorities to monitor berth utilization, predict congestion patterns and make predictive scheduling choices, thereby maximizing overall operational management and efficiency.

2.4.4 Adaptive Rescheduling and Real-Time Adjustments

Unforeseen disruptions such as bad weather, breakdown of ships, or sudden priority changes at the last minute will typically require rescheduling berths. Rescheduling berths and resource redistribution will be automatically taken care of by the AI model without any intervention, ensuring continuity even in unexpected situations. This prevents delays caused by inefficient rescheduling and reduces the workload of port operators.

2.4.5 User-Friendly Dashboard for Port Operators

In order to facilitate mass adoption, the system will include an intuitive, real-time dashboard showing graphical representations of berth allocations, ship status and system-recommended suggestions. The interface will be simple enough to be used by non-technical port operators with unambiguous, actionable information but permit manual overrides when required.

2.4.5.1 Scalability and Adaptability for Different Port Environments

The system is made scalable to different ports with different working constraints. Depending on whether the port is handling container ships, bulk ships, or tankers, the AI model can manage different requirements under berth limit, cargo type and priority constraints. This makes the system applicable in different port environments with less redesign. With the achievement of these objectives, PortFlow will provide the next-generation berth allocation AI solution that ensures smarter, adaptive and optimized port operations at lower costs, eased congestion and lower environmental impact.

2.5 Project Scope

The scope of the project is to design and develop an AI-driven berth allocation system to optimize vessel scheduling, berth and port resource usage. The system will employ deep reinforcement learning and hybrid AI-heuristic models to reassign and assign berths dynamically in real-time based on vessel arrival, congestion and operational considerations. The project will also include an interactive dashboard to provide port operators with real-time berth monitoring, predictive analytics and AI-driven berth recommendations to support decision-making. The project will be able to:

- AI-based dynamic berth allocation to minimize waiting times and optimize maximum efficiency.
- An interactive dash to display berth occupation, ship schedules and AI-based berth suggestions to port operators.
- Automated rescheduling of berths to handle unplanned delays, priority shipments and operational disruptions efficiently.

The system will be built using a mix of Python, web technology and machine learning libraries for the dashboard.It will process data and leverage reinforcement learning models to refine berth allocation decisions with time.

2.5.1 Project Deliverables

The project deliverables include the following and encompass key components essential for the development and evaluation of the AI-driven berth allocation system.

- A functional AI-driven berth allocation system capable of real-time berth optimization.
- A user-friendly dashboard for berth monitoring, manual adjustments and AI-based recommendations.
- Documentation of the system architecture, design, implementation and testing processes.
- A performance evaluation report comparing AI-driven scheduling with traditional berth allocation methods.

2.5.2 Out Of Scope

The following aspects are beyond the scope of this project and will not be included in the current implementation.

- The system will not provide real-time vessel tracking via AIS (Automatic Identification System) for the time being; it will rely on existing port data.
- It will not include mobile applications; the dashboard will be web-based.
- The project does not cover physical port infrastructure upgrades; it focuses solely on software-based berth optimization.
- Third-party system integrations (e.g., external logistics platforms or customs databases) are not included in the initial implementation.

This project scope will serve as a guide throughout development, ensuring a focused, efficient and AI-driven berth allocation solution that improves port operations, reduces congestion and enhances decision-making for port authorities.

2.6 Sustainable Development Goal (SDG)

The berth allocation system based on AI designed in this project supports several Sustainable Development Goals (SDGs) by increasing port efficiency, decreasing congestion and maximizing the use of resources through smart scheduling. The most applicable SDGs addressed in this project are SDG 8 (Decent Work and Economic Growth), SDG 9 (Industry, Innovation and Infrastructure), SDG 11 (Sustainable Cities and Communities) and SDG 13 (Climate Action).

SDG 8 (Decent Work and Economic Growth) is facilitated by effective berth management, which minimizes delays, maximizes resource utilization and enhances overall port productivity. Through the automation of berth allocation and minimizing inefficiencies, the system enables smoother global trade flow, which creates improved economic opportunities for shipping companies, port workers and logistics operators.

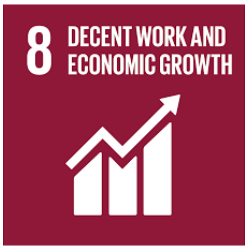

Figure 2.1: Decent Work and Economic Growth - Promote sustained, inclusive economic growth, full employment and decent work for all.

SDG 9 (Industry, Innovation and Infrastructure) is covered by the application of artificial intelligence to conventional port operations. The project incorporates data-driven decision-making, predictive analysis and real-time berth planning, updating port infrastructure and facilitating the digital revolution of maritime logistics.

Figure 2.2: Industry Innovation and Infrastructure - Build resilient infrastructure, promote sustainable industrialization and foster innovation.

SDG 11 (Sustainable Cities and Communities) is applicable as the system reduces port congestion, resulting in smoother logistics, less traffic disruption and reduced emissions caused by idling vessels. Optimizing berth allocation, the project promotes sustainable urban development and increases port-city integration.

Figure 2.3: Sustainable Cities and Communities - Make cities inclusive, safe, resilient and sustainable

SDG 13 (Climate Action) is also taken care of since optimized berth scheduling decreases excessive fuel use and emissions by waiting ships. Through minimizing idling, increasing the efficiency of scheduling and reducing congestion at ports, the system assists in decreasing the environmental footprint of shipping operations and, in the process, contributing to the worldwide carbon footprint reduction drive in the shipping sector.

Figure 2.4: Climate Action - Take urgent action to combat climate change and its impacts.

2.7 Constraints

The project is subject to certain constraints that may impact its development and implementation.

- The system must employ accurate and up-to-date vessel arrival, berth availability and port congestion data to operate effectively.
- High-performance reinforcement learning algorithms are computationally demanding, which may be a limitation in real-time processing for large ports.
- Ports vary in size, infrastructure and operational processes, which makes system flexibility among different environments challenging.
- Most ports use aged scheduling systems that must be carefully integrated to implement them smoothly.
- Weather conditions, mechanical failures and regulatory hold-ups are still set to hamper berth allocation despite AI optimization.
- While AI-driven, the system has to allow human intervention, i.e., human operators must be able to override AI decisions when necessary.

2.8 Business Opportunity

Congestion and inefficient berth allocation cause billions of dollars in yearly losses from delayed shipments, wasted fuel and unused resources. The international maritime community is looking for intelligent, data-driven solutions to make port operations more efficient and turn-around times faster. This project offers a major business opportunity by providing an AI-based berth allocation system that maximizes efficiency, minimizes costs and facilitates sustainable logistics. The implementation of AI in port management is in its infancy, so PortFlow is a competitive edge for ports seeking to update their scheduling processes. The scalability and flexibility of the system also enable commercial deployment at different ports, providing potential partnerships with shipping companies, port authorities and logistics companies.By optimizing operations and reducing environmental impact, the system aligns with global trends towards green, smart and automated port operations and is a shrewd investment in the maritime logistics of the future.

2.9 Stakeholders Description/ User Characteristics

The AI-based berth scheduling system will be utilized by various stakeholders of maritime port and logistics operations. The users play a critical role in efficient berth scheduling, handling cargo and port operations. The system will ease their work by automating berth allocation, minimizing delays and providing real-time insights through an interactive dashboard.

2.9.1 Port Authorities

Port authorities will be required to oversee berth allocation, port traffic management and operational efficiency. They will employ the system to monitor berth utilization, scrutinize AI-suggested berth proposals and reschedule manually when necessary. Incorporating real-time data, the system will allow port authorities to reduce congestion, improve turnaround time and optimize port resource allocation.

subsection Terminal Operators Terminal operators oversee the cargo handling, ship berthing and resource allocation within a port. All these operations completely rely on optimized berth planning with an eye to maximizing crane and storage yard utilization and efficient use of labor.They will receive immediate notification of vessel schedules and berths from the system so they can schedule and perform cargo activities efficiently. By decreasing uncertainty in berth scheduling, the system will minimize idle time for handling equipment and enhance overall coordination of operations.

2.9.2 Shipping Lines and Ship Owners

Shipping lines and ship owners depend on smooth port operations to keep turnaround time and costs to a minimum. Berthing delay results in increased fuel consumption, late delivery schedules and interrupted supply chain schedules. Through real-time berth availability, predictive scheduling and optimized berth allocation, the system will enable ship owners to plan arrivals better, saving idle time and fuel consumption

2.9.3 Logistics and Supply Chain Managers

Logistics and supply chain managers require reliable berth scheduling information to efficiently plan cargo movements. Port delays can lead to inland transportation and warehousing bottlenecks. The system will offer them congestion predictions, vessel scheduling information and AI-based predictive analytics, allowing them to make proactive adjustments to transport plans and reduce supply chain disruptions.

2.9.4 Government and Regulatory Bodies

Government authorities and port regulatory organizations govern compliance, trade rules and environmental sustainability. They need fact-based insights on port efficiency, congestion levels and environmental effects. The system will provide analytics and performance reports that can support regulatory organizations in policy-making, environmental evaluations and infrastructure planning.

2.9.5 Stakeholders Summary

The users of the berth allocation system are port authorities, terminal operators, shipping lines, logistics managers and regulatory agencies. Port authorities and terminal operators will employ the system to schedule berth allocation, minimize congestion and maximize cargo handling efficiency. Shipping companies and ship operators will enjoy optimized scheduling and minimum waiting times, which will result in less fuel consumption and more efficient supply chain management. Logistics managers will have improved visibility of berth availability, enabling effective transport planning. Government and regulatory agencies will use the system's performance analytics for infrastructure planning and policy development.

2.9.6 Stakeholders' Key High-Level Goals and Issues

Port authorities seek to maximize berth allocation, minimize congestion and enhance port efficiency but are hindered by manual scheduling inefficiencies and real-time adjustments. Terminal operators need precise berth scheduling to coordinate cargo handling, crane operations and workforce allocation, but random berth changes frequently upset their planning. Shipping lines and ship owners want reduced waiting time and fuel expenditure, but berth delays mean extended idle time, higher operating costs and disruptions to the supply chain. Supply chain and logistics managers rely on predictable berth scheduling information to optimize transportation operations, but unreliable berth availability means delayed transport and warehouse inefficiency. Government and regulatory agencies require complete port performance analysis to inform trade policy and environmental sustainability efforts but lack real-time access to port efficiency information. PortFlow will overcome these obstacles by offering an AI-driven berth allocation system that improves decision-making, berth productivity and operational transparency, allowing for a more intelligent, responsive approach to port management.

2.10 Conclusion

This chapter presented the vision, challenges and objectives of the AI-based berth allocation system, emphasizing its potential to optimize port operations with the aid of smart scheduling and real-time decision-making. The inefficiency of berth allocation, congestion and mismanagement of resources was mentioned as a problem, along with the necessity of an adaptive AI-based solution. The important objectives and operational issues were determined to assure the system complies with industry demands. Moreover, the project's potential in aligning with Sustainable Development Goals and its business viability were investigated, highlighting its applicability in port modernization and maritime logistics. With the application of deep reinforcement learning and hybrid AI-heuristic models, this system will minimize ship waiting times, enhance berth usage and offer real-time analytics through an interactive dashboard, thereby revolutionizing conventional port scheduling into a wiser, more efficient process

Chapter 3 Literature Review / Related Work

Berth Allocation Problem (BAP) is a fundamental operational problem in maritime logistics with direct impacts on port efficiency, vessel turnaround time and supply chain performance. With rising volumes of international trade, ports are facing mounting pressure to minimize delays, lower operating costs and optimize the utilization of resources. Traditional approaches to BAP, such as deterministic optimization models and heuristic algorithms, have been unable to meet dynamic and uncertain conditions, such as varying vessel arrival times, uncertain cargo handling requirements and varying port conditions. New developments in Artificial Intelligence (AI), particularly in machine learning (ML) and reinforcement learning (RL), have introduced new hope in resolving these problems. AI-based approaches have potential for real-time flexibility, predictive decision-making and higher scalability and are therefore highly appropriate to dynamic and intricate port operations.

This review combines seminal research articles on berth allocation with focus on the evolution from traditional optimization approaches to modern AI-based approaches. The discussed papers span a wide range of methodologies, from genetic algorithms and metaheuristics to deep reinforcement learning and AI-optimization hybrid models. Based on their strengths, weaknesses and feasibility to modern port operation, this review aims to provide an integrative overview of the state-of-the-art berth allocation research. It also emphasizes the potential of AI-based solutions to overcome current challenges, such as real-time responsiveness, scalability and coordination with other port operations.

3.1 Definitions, Acronyms and Abbreviations

BAP: Berth Allocation Problem - The challenge of assigning berths to arriving vessels to optimize port efficiency.

MQ-BAP: Multi-Quay Berth Allocation Problem - A berth allocation problem involving multiple quays and dynamic scheduling.

GA: Genetic Algorithm - An evolutionary optimization technique inspired by natural selection.

CSA: Cuckoo Search Algorithm - A metaheuristic optimization method inspired by cuckoo bird nesting behavior.

BSP: Berth Scheduling Problem - The process of determining the optimal schedule for vessels at berths to maximize port efficiency and minimize delays.

EA:Evolutionary Algorithm - A heuristic optimization technique inspired by natural evolution, often used for solving complex problems.

ACO: Ant Colony Optimization - A nature-inspired algorithm that mimics the foraging behavior of ants to solve complex problems.

DQN: Deep Q-Network - A reinforcement learning algorithm used for sequential decision-making.

D3QN: Dueling Double Deep Q-Network - An advanced reinforcement learning model that enhances policy evaluation.

MILP: Mixed-Integer Linear Programming - A mathematical approach for solving optimization problems with integer and continuous variables.

RL: Reinforcement Learning - A branch of machine learning where an agent learns to make optimal decisions through trial and error.

DRL:Deep Reinforcement Learning - A combination of reinforcement learning and deep learning, enabling the agent to handle high-dimensional input spaces.

MDP:Markov Decision Process (MDP) - A mathematical framework for modeling decision-making in situations where outcomes are partly random and partly under the control of a decision-maker.

DTBs: Dynamic Time Buffers (DTBs) - Time allowances incorporated into scheduling to account for uncertainties and variations in vessel arrival times.

MARL:Multi-Agent Reinforcement Learning - A reinforcement learning framework where multiple agents interact and learn in a shared environment.

PPO : Proximal Policy Optimization - A policy optimization algorithm used in reinforcement learning to improve decision-making policies.

SLR : Systematic Literature Review - A structured method for reviewing and synthesizing existing research on a specific topic.

Simulation-Based Optimization: A technique that combines simulations with optimization algorithms to improve decision-making.

Port Congestion: A situation where excessive vessel arrivals exceed a port's handling capacity, causing delays.

Vessel Turnaround Time: The total time a vessel spends in a port, from arrival to departure, including berthing and cargo operations.

Hybrid Optimization Model: A combined approach that integrates multiple optimization techniques for improved performance.

3.2 Detailed Literature Review

This section provides an in-depth review of relevant literature on berth allocation, focusing on traditional heuristics, advancements in deep reinforcement learning and AI-based approaches. Each study is examined for its contributions, strengths, limitations, and relevance to the proposed research, with a particular emphasis on optimization techniques for port scheduling, congestion management and operational efficiency.

3.2.1 An Improved Deep Reinforcement Learning Approach for Berth Allocation[1]

Below is the summary, critical analysis, and relevance of the selected article to the current research.

3.2.1.1 Summary:

This paper introduces a Deep Reinforcement Learning (DRL) method for berth and yard scheduling optimization in bulk cargo terminals. It models the problem as a Markov Decision Process (MDP) and proposes an improved PS-D3QN algorithm (Prioritized Experience Replay Softmax-based Dueling Double DQN). The research tests the model with real port data and concludes that PS-D3QN performs better than conventional scheduling algorithms in reducing vessel waiting time and lowering operational costs.

3.2.1.2 Critical analysis of the research item:

This approach applies actual port data for verification, thus providing a more reliable outcome. The deep reinforcement learning (DRL) method learns dynamically rather than relying on fixed rules and it handles both berth and yard scheduling, which contributes to improved port efficiency.

However, the approach requires massive training data, which may not always be available in actual conditions. Additionally, the computational demands for deep reinforcement learning models can be high.

3.2.1.3 Relationship to the proposed research work:

This paper is very pertinent as it directly uses DRL to berth allocation. Our research could extend this work by investigating hybrid models that combine heuristic and RL-based methods.

3.2.2 Optimizing Berth Allocation in Maritime Transportation with Quay Crane Setup Times Using Reinforcement Learning[2]

Below is the summary, critical analysis, and relevance of the selected article to the current research.

3.2.2.1 Summary:

This paper introduces a reinforcement learning (RL)-based approach for berth allocation that considers quay crane setup times. The proposed method uses both offline (greedy-insert algorithm) and online (reinforcement learning) strategies to optimize berth allocation dynamically. The study demonstrates that RL-based scheduling can significantly reduce vessel waiting times and enhance cargo handling efficiency compared to traditional methods.

3.2.2.2 Critical analysis of the research item:

This method utilizes reinforcement learning, making it adaptable to real-time changes in vessel schedules. It takes into account quay crane setup times, an important factor often overlooked in berth allocation studies. Moreover, it demonstrates superior performance over traditional first-come-first-served (FCFS) approaches.

However, the study focuses only on quay crane setup and berth allocation, ignoring other port operations. The effectiveness of reinforcement learning depends on training quality and computational resources, which may limit scalability. Additionally, there is no actual-world implementation provided—only mathematical modeling without supporting numerical experiments.

3.2.2.3 Relationship to the proposed research work:

This paper is highly relevant because it applies AI (specifically reinforcement learning) to berth allocation. Our research can build upon this by incorporating additional AI techniques, such as hybrid AI-heuristic models or real-world testing with historical berth data.

3.2.3 Discrete Dynamic Berth Allocation Optimization in Container Terminal Based on Deep Q-Network[3]

Below is the summary, critical analysis, and relevance of the selected article to the current research.

3.2.3.1 Summary:

This paper presents a profound reinforcement learning solution with a Deep Q-Network (DQN) for berth allocation with stochastic ship arrival times and varying load capacities. The model possesses a striking decrease in vessel waiting times compared to traditional optimization methods like Ant Colony Optimization (ACO) and Genetic Algorithms (GA). Specifically, the DQN-based model reduced overall waiting time by approximately 57 percent compared to previous methods.

3.2.3.2 Critical analysis of the research item:

The DQN-based model adapts dynamically to uncertain and dynamic vessel arrivals, allowing for more responsive and efficient scheduling. This method significantly enhances efficiency compared to traditional heuristic approaches.

However, the model requires substantial computational power for training and its performance is highly dependent on the quality of data and careful parameter tuning.

3.2.3.3 Relationship to the proposed research work:

As our work is all about the application of deep learning in berth management, reinforcement learning utilized in this paper is closest to what we are targeting. We can take advantage of its experience in schedule optimization using AI-based methods.

3.2.4 Berth Allocation for Multiple Container Terminal Joint Operation Using Dueling Double Deep Q-Network[4]

Below is the summary, critical analysis, and relevance of the selected article to the current research.

3.2.4.1 Summary:

This paper introduces a Dueling Double Deep Q-Network (D3QN) solution to the Multi-Terminal Dynamic and Continuous Berth Allocation Problem (MDC-BAP). The solution maximizes berth allocation with water depth and integration problems at the terminals as constraints. Solution quality is demonstrated to be improved by 3.7 percent using D3QN compared to other traditional optimization algorithms, including Proximal Policy Optimization (PPO) and traditional Deep Q-Networks (DQN).

3.2.4.2 Critical analysis of the research item:

This approach beats existing best practices on larger-sized berth allocation issues and features models that respond dynamically to real ship arrivals and terminal restrictions.

However, the computational cost is too expensive for real-time implementation. Additionally, a significant quantity of training data must be used to ensure that the model generalizes well across different port situations.

3.2.4.3 Relationship to the proposed research work:

The study is particularly suited to our berth allocation AI system. Using reinforcement learning methods like D3QN can have some interesting ramifications on optimizing the scheduling process effectively under dynamic conditions.

3.2.5 A Self-Adaptive Evolutionary Algorithm for the Berth Scheduling Problem Towards Efficient Parameter Control[5]

Below is the summary, critical analysis, and relevance of the selected article to the current research.

3.2.5.1 Summary:

This work addresses Berth Scheduling Problem(BSP) optimization through a self-tuning Evolutionary Algorithm (EA). The proposed approach adjusts the crossover and mutation probabilities dynamically in order to optimize berth allocation efficiency and minimize vessel turnaround time. The performance is compared to traditional deterministic and adaptive EAs and is seen to be significantly improved in terms of the solution quality without increasing the computational complexity.

3.2.5.2 Critical analysis of the research item:

The self-adaptive EA obtains higher berth scheduling effectiveness compared with conventional evolutionary methods. The approach most balances quality of optimization and computation time. It also considers realistic scheduling constraints like vessel turnaround and departure delays.

However, the method does not include deep learning-based optimization methods. The reliance on parameter tuning for evolutionary algorithms can reduce generalizability across various port conditions.

3.2.5.3 Relationship to the proposed research work:

As our research centers on berth allocation using AI, this paper provides insight into heuristic-based optimization. Incorporating reinforcement learning (RL) approaches, however, would introduce flexibility and real-time decision-making to our research.

3.2.6 Genetic Algorithm for the Dynamic Berth Allocation Problem[6]

Below is the summary, critical analysis, and relevance of the selected article to the current research.

3.2.6.1 Summary:

This paper suggests a genetic algorithm (GA) to address the dynamic berth allocation problem (DBAP). The model considers vessel arrival uncertainty and optimizes berth allocation by reducing service time and enhancing quay crane scheduling efficiency.

3.2.6.2 Critical analysis of the research item:

Using a metaheuristic approach to dynamic berth allocation, this method provides experimental data with real port statistics. It also considers dynamic constraints such as fluctuating ship size and quay capacity, ensuring its applicability to real-world port operations.

However, GA-based approaches do not generalize to highly dynamic environments very well. Additionally, it lacks real-time adaptability compared to reinforcement learning models.

3.2.6.3 Relationship to the proposed research work:

Whereas genetic algorithms can be applied to berth allocation, our emphasis on AI-based techniques indicates that reinforcement learning may provide a more adaptive and flexible means of dynamic berth scheduling.

3.2.7 Berth Allocation Model for Container Terminal using Genetic Algorithm Technique[7]

Below is the summary, critical analysis, and relevance of the selected article to the current research.

3.2.7.1 Summary:

This paper proposes a Genetic Algorithm-based model (GAMBA) of berth allocation at Apapa Wharf in Nigeria to minimize vessel delay times. Quay length and handling time are identified as critical parameters influencing port efficiency in the research. The finding indicates the efficiency benefits of a 250m increase in quay length are comparable to handling time reduction by 0.0025 h/m, proposing alternative optimization options instead of costly port expansions.

3.2.7.2 Critical analysis of the research item:

The Genetic Algorithm is a strong method of optimization for berth allocation. Research based on real data from the Apapa container terminal adds to its practical significance.

However, dynamic uncertainties of vessel arrivals are not taken into consideration in the model. Additionally, there is no direct comparison with modern AI-based methods such as deep reinforcement learning.

3.2.7.3 Relationship to the proposed research work:

Usage of Genetic Algorithms to the berthing allocation is applicable to our study, particularly in the search for other optimization methods. The study does not, however, incorporate an adaptive model to accommodate real-time adjustments, a characteristic which our methodology seeks to mitigate by the use of deep learning.

3.2.8 Dynamic and Continuous Berth Allocation Using Cuckoo Search Optimization[8]

Below is the summary, critical analysis, and relevance of the selected article to the current research.

3.2.8.1 Summary:

This work employs the Cuckoo Search Algorithm (CSA) as a metaheuristic optimization for berth allocation under a dynamic continuous setting. In this paper, berth allocation is presented as a Mixed-Integer Linear Programming (MILP) model and is compared against solutions of Genetic Algorithms (GA) and MILP. Results show that CSA provides improved berth allocation with fair computation time.

3.2.8.2 Critical analysis of the research item:

CSA performs better than GA and MILP-based berth allocation methods. The model simulates real-time dynamic ship arrival.

However, the research does not compare deep learning-based methods with CSA. Additionally, the impact of external port conditions (e.g., weather, congestion) is not taken into account.

3.2.8.3 Relationship to the proposed research work:

The emphasis of this research on berth allocation optimization in dynamic settings is applicable to our research, specifically in using AI-based methodologies. While CSA is a viable alternative to heuristics, reinforcement learning provides more flexibility in practical application.

3.2.9 The Berth Allocation Problem in Bulk Ports[9]

Below is the summary, critical analysis, and relevance of the selected article to the current research.

3.2.9.1 Summary:

This research formulates a Mixed Integer Linear Programming (MILP) model to optimize bulk terminal berth allocation. The model reduces vessel waiting time and berth inefficiency by taking into account dynamic vessel arrival, berth capacity and operational constraints. The research provides computational tests on real data of SAQR Port, UAE.

3.2.9.2 Critical analysis of the research item:

SThe method implies an optimization approach for berth allocation. It includes real-world constraints from a case study at SAQR Port, UAE and minimizes uncertainty in ship schedules and operational disruptions.

However, it has limited usage for container terminals, as it is used for bulk cargo. Additionally, it does not employ AI-driven strategies but instead uses conventional optimization techniques.

3.2.9.3 Relationship to the proposed research work:

PortFlow can scale MILP-based berth allocation models to enhance AI-driven scheduling for maximum berth utilization and minimum wait time.

3.2.10 Enhanced Berth Allocation Based on the Cuckoo Search Algorithm[10]

Below is the summary, critical analysis, and relevance of the selected article to the current research.

3.2.10.1 Summary:

In this paper, the Cuckoo Search Algorithm (CSA) is utilized in resolving the berth allocation problem (BAP) that tries to minimize the service cost such as waiting, handling and penalty cost of non-ideal berth allocation. CSA is compared with Genetic Algorithms (GA) and a Mixed-Integer Linear Programming (MILP) solution and proves that CSA performs better berth allocation with reasonable computation time.

3.2.10.2 Critical analysis of the research item:

CSA outperforms GA and MILP solutions for berth allocation. The study suggests additional constraints, such as a penalty for non-ideal berth allocation and a safety buffer.

However, the research is largely heuristic in character and does not include the option of deep learning. Additionally, it is very sensitive to parameter tuning and specific dataset settings.

3.2.10.3 Relationship to the proposed research work:

As our study is concerned with berth allocation optimization through AI methods, CSA is a suitable benchmark for heuristic solutions. Techniques based on deep learning such as reinforcement learning, however, can provide more scalable and flexible solutions.

3.2.11 Multi-Quay Berth Allocation Optimized with the Cuckoo Search Algorithm[11]

Below is the summary, critical analysis, and relevance of the selected article to the current research.

3.2.11.1 Summary:

This study develops an optimization model for the Cuckoo Search Algorithm (CSA) to solve the Multi-Quay Berth Allocation Problem (MQ-BAP). The model accounts for dynamic ship arrivals, actual constraints like safety windows and the effectiveness of berth assignment. The study uses actual data from the Port of Limassol, Cyprus and compares the results of CSA with Genetic Algorithms (GA) and a Mixed-Integer Linear Programming (MILP) approach. Results show that CSA significantly reduces computational time required for MILP while optimizing berth allocation.

3.2.11.2 Critical analysis of the research item:

CSA achieves savings in computational expenses when it comes to berth allocation in comparison to exact MILP methods. The model also accounts for safety time margins and preferred berthing locations as practical constraints.

However, the model does not account for deep learning adaptive optimization techniques. Additionally, the algorithm's implementation relies on hyperparameter tuning, which may limit its applicability to other ports.

3.2.11.3 Relationship to the proposed research work:

The focus of this study is AI driven berth allocation therefore the metaheuristic approach of this study is easily adapted for AI based berth allocation. However, incorporating reinforcement learning techniques could provide more flexibility and effectiveness in the control of real-time changes to the system.

3.2.12 Priority Control of Berth Allocation Problem in Container Terminals[12]

Below is the summary, critical analysis, and relevance of the selected article to the current research.

3.2.12.1 Summary:

This paper presents a Decision Support System (DSS) for berth allocation in container terminals. The DSS is a discrete event simulation model with a built-in optimization module to rank berth allocation. The study emphasizes that priority berth allocation (as opposed to first-come-first-served) is superior and reduces waiting time. The DSS is validated with a case study of a Turkish container terminal and the Port of Rotterdam.

3.2.12.2 Critical analysis of the research item:

The method proposes a Decision Support System (DSS) encompassing simulation and optimization for allocating berths. It demonstrates effectiveness through real port case studies and employs a priority-based system rather than a first-come-first-served system.

However, it primarily relies on heuristic optimization as opposed to adaptive AI-based models. Additionally, it does not apply reinforcement learning or machine learning for dynamic berth allocation. The method is limited to container terminals, which can lead to applicability constraints in mixed cargo ports.

3.2.12.3 Relationship to the proposed research work:

Since our research also focuses on AI-driven berth allocation, this study is useful for understanding priority-based berth assignment methods. However, unlike this paper, our approach will incorporate AI techniques (potentially reinforcement learning) to dynamically adjust berth allocations based on historical data rather than relying purely on a rule-based DSS.

3.2.13 Berth Allocation and Scheduling at Marine Container Terminals [13]

Below is the summary, critical analysis, and relevance of the selected article to the current research.

3.2.13.1 Summary:

The article provides a holistic overview of berth scheduling and berth allocation models in marine container terminals (MCTs). The article categorizes existing research with respect to spatial features (discrete, continuous, hybrid), hypotheses of vessel arrivals (static, dynamic, stochastic), as well as the handling time model. The work identifies key mathematical formulations (MILP, MINLP) and addresses heuristic, metaheuristic, as well as AI-based optimization methods. The paper identifies key areas of research shortfall, such as the lack of real-time adaptable models and an increasing need for energy-efficient, sustainable scheduling

3.2.13.2 Critical analysis of the research item:

The study provides an extensive literature review of berth allocation practices. It identifies existing challenges and emerging directions of research, while also including a variety of optimization techniques, from heuristics to AI algorithms.

However, it does not present new experiments or procedures. Additionally, there is a lack of attention to AI-based techniques like deep reinforcement learning (DRL).

3.2.13.3 Relationship to the proposed research work:

This review creates a reference point for comprehension on existing berth allocation models and helps to pinpoint gaps in research. The approaches that we blended with AI techniques can resolve some of the issues emphasized in this survey.

3.2.14 Distributionally Robust Optimization for the Berth Allocation Problem [14]

Below is the summary, critical analysis, and relevance of the selected article to the current research.

3.2.14.1 Summary:

This study proposes a distributionally robust optimization (DRO) model for berth allocation problem with uncertainty in vessel handling times. The mismatch between the worst-case expected delays and the actual delays is minimized by taking into account multiple probability distributions of handling time variations.

3.2.14.2 Critical analysis of the research item:

Captures uncertainty in berth scheduling within a sound optimization platform. Offers a strong mathematical foundation with precise solution methods. Compares performance with stochastic and deterministic approaches.

However, does not utilize AI or learning-based optimization techniques. Computational complexity may restrict real-time use.

3.2.14.3 Relationship to the proposed research work:

Our research aims to combine AI-driven techniques, i.e., reinforcement learning, that could augment or complement the DRO platform by allowing adaptive learning through simultaneous port operations.

3.2.15 Improved Ant Colony Algorithm for Discrete Dynamic Berth Allocation in a Case Container Terminal[15]

Below is the summary, critical analysis, and relevance of the selected article to the current research.

3.2.15.1 Summary:

This research solves the Discrete Dynamic Berth Allocation Problem (DDBAP) using the Parallel Search Structure Enhanced Ant Colony Algorithm (PACO). The algorithm performs optimized berth allocation based on actual terminal constraints like uncertainty of vessel arrival time and loading capacity. Experimental evidence with real data for the Shanghai Port shows that PACO performs better and to a larger extent than conventional heuristics.

3.2.15.2 Critical analysis of the research item:

PACO is a stronger and more efficient solution than the standard Ant Colony Optimization (ACO). The algorithm can effectively integrate actual-world constraints including ship congestion as well as scheduling uncertainty.

However, the technique is highly dependent on well-adjusted parameters to obtain the best results. Deep learning is not examined to any extent by the research and can further advance decision-making processes potentially.

3.2.15.3 Relationship to the proposed research work:

As our focus is on AI-driven berth allocation methods, the optimization technique mentioned here is adaptable. But integrating reinforcement learning or hybrid AI methods can introduce additional enhancement in addressing real-time dynamics.

3.2.16 A Survey on Reinforcement Learning in Aviation Applications [16]

Below is the summary, critical analysis, and relevance of the selected article to the current research.

3.2.16.1 Summary:

This paper gives a critical overview of Reinforcement Learning (RL) applications in the aviation industry. It mentions applications from air traffic control, collision avoidance, flight scheduling and aircraft maintenance. The paper explains value-based (Q-learning, DQN), policy-based (REINFORCE, PPO) and actor-critic (A3C, SAC) methods. It mentions multi-agent RL (MARL) for collaborative decision-making and says some challenges are availability of data, safety and real-time adjustment

3.2.16.2 Critical analysis of the research item:

Provides a detailed overview of RL techniques in aviation. Explores RL implementations in dynamic, uncertain environments like berth allocation. Emphasizes effective RL-based optimization techniques.

However, there is sparse treatment of hybrid methods that integrate RL with conventional optimization techniques. It does not address mass-scale real-time deployment issues, which are critical for operational situations.

3.2.16.3 Relationship to the proposed research work:

While the domain is different, the problem structures are analogous, particularly in dynamic scheduling and resource allocation. The techniques covered here may motivate our RL-based solution for berth allocation.

3.2.17 A Simulation Study of Collaborative Approach to Berth Allocation Problem under Uncertainty [17]

Below is the summary, critical analysis, and relevance of the selected article to the current research.

3.2.17.1 Summary:

This research focuses on collaborative berth allocation strategy in which multiple port terminals cooperate to share berths, quay cranes and container yards for improved vessel arrival handling. The study employs discrete event simulation to test collaborative and non-collaborative approaches, finding that waiting times for vessel arrivals is reduced and port throughput is increased with collaborative berth allocation.

3.2.17.2 Critical analysis of the research item:

Develops a collaborative model for berth allocation, defining efficiency as the result of resources being shared across terminals. Validates heuristically by simulation different strategies based on stochastic vessel arrivals. Tries to consider uncertainties of operational nature that are relevant in practice concerning scheduling of berths.

However, it only considers a collaborative model, which in some circumstances may not always be the best solution. It solely employs simulation-based heuristics. The work is limited to one port case study and therefore lacks generalizability.

3.2.17.3 Relationship to the proposed research work:

We focus on AI-based berth allocation systems, while this works aids in the understanding of uncertainty management and resource allocation. If collaboration between terminals is possible in the data we receive, we will implement a hybrid AI model that uses multi-terminal berth allocation methods.

3.2.18 Robust Berth Scheduling Using Machine Learning for Vessel Arrival Time Prediction[18]

Below is the summary, critical analysis, and relevance of the selected article to the current research.

3.2.18.1 Summary:

This paper explores machine learning-based prediction of vessel arrival times for improving berth scheduling. Four machine learning algorithms (Linear Regression, k-Nearest Neighbors, Decision Trees and Artificial Neural Networks) are compared for predicting vessel arrivals based on AIS data. Dynamic Time Buffers (DTBs) are introduced to address prediction uncertainty and reduce berth schedule deviations. A numerical study shows that ML-based predictions significantly improve scheduling robustness and reduce vessel waiting times.

3.2.18.2 Critical analysis of the research item:

Uses actual AIS data in forecasting ship arrivals. Compares several machine learning models to determine the most effective forecasting technique. Introduces Dynamic Time Buffers (DTBs) to enhance berth scheduling in uncertainty.

However, it does not encompass metaheuristic algorithms and reinforcement learning in berth scheduling.

3.2.18.3 Relationship to the proposed research work:

This study is extremely relevant to our work because it shows the effectiveness of machine learning in berth scheduling. But unlike in this study, unlike predicting the arrival times, we also wish to incorporate AI-optimized berth allocation in the scheduling. We can also consider hybrid AI techniques (combining ML prediction with optimization techniques) to enhance berth planning.

3.2.19 Leveraging Machine Learning and Optimization Models for Enhanced Seaport Efficiency [19]

Below is the summary, critical analysis, and relevance of the selected article to the current research.

3.2.19.1 Summary:

This paper introduces the coupling of optimization models and machine learning (ML) for optimizing seaport performance. This focuses on predictive analytics to predict congestion and reinforcement learning (RL) for berth scheduling. The authors review 124 papers and conclude by calling out key trends in data-driven port decision-making.

3.2.19.2 Critical analysis of the research item:

Identifies areas of research gaps in AI-based optimization in ports. Outlined predictive maintenance and operating control procedures. Recommends hybrid AI-optimization architectures for decision-making.

However, it is more theoretical and lacks case studies of implementation. Additionally, there is inadequate attention towards container handling and berth allocation.

3.2.19.3 Relationship to the proposed research work:

PortFlow can integrate optimization models with machine learning for data-driven berth scheduling, which can result in more efficient and quicker port operations.

3.2.20 Transforming the Shipping Industry: Integrating AI-Powered Virtual Port Operators for End-To-End Optimization [20]

Below is the summary, critical analysis, and relevance of the selected article to the current research.

3.2.20.1 Summary:

An AI-based virtual port operator presented in this study aims at improving container shipping activities through advanced predictive decision-making, dynamic resource allocation and optimization. The port productivity enhancement process Machine learning, genetic algorithms and real-time data analysis are coupled to increase port productivity, decrease vessel turnaround time and improve berth occupancy. The study presents case study port operations that have been significantly more efficient, cost-effective and sustainable demonstrating the results of the research.

3.2.20.2 Critical analysis of the research item:

Presents an all-inclusive AI port management system. Combines optimization with real-time decision-making and predictive analytics. Utilizes case studies to demonstrate the effectiveness of improved vessel turnaround and resource allocation.

However, it is aimed at improving overall port operations, not specifically targeted at berth allocation optimization. No adequate comparison of different AI models applied in port management is provided. Little attention is paid to the implementation problems, especially the assimilation with the legacy port systems.

3.2.20.3 Relationship to the proposed research work:

This study is relevant because it highlights the optimization of ports operations on AI. Our research is more narrowed down to berth allocation as opposed to port management. We intend to create a berth scheduling model that is based on AI, possibly incorporating elements from this study such as predictive decision-making and adaptability in real-time.

3.2.21 Application of Artificial Intelligence in Maritime Transportation [21]

Below is the summary, critical analysis, and relevance of the selected article to the current research.

3.2.21.1 Summary:

This research investigates AI technologies applied to maritime logistics, including autonomous ship navigation, predictive maintenance, AI-based berth scheduling and real-time traffic monitoring. The authors suggest the potential of machine learning (ML) and AI optimization models to enhance maritime safety, minimize delays and increase scheduling efficiency. It also addresses AI-enhanced ship-to-port communication to improve coordination.

3.2.21.2 Critical analysis of the research item:

Provides a comprehensive overview of AI applications in maritime transport. Discusses ship–port–vehicle cooperation and automation in berthing operations. Highlights the benefits of AI for predictive decision-making.

However, it does not present specific AI models or algorithms for implementation. It is more focused on shipping operations rather than port-side logistics.

3.2.21.3 Relationship to the proposed research work:

PortFlow can utilize AI-based predictive analytics for scheduling of berths, decreasing delays and enhancing the efficiency of cargo flow through data-driven optimization.

3.2.22 Analysis of Modern Port Technologies [22]

Below is the summary, critical analysis, and relevance of the selected article to the current research.

3.2.22.1 Summary:

This paper analyzes contemporary technologies in port automation with an emphasis on the IoT, Artificial Intelligence and Big Data for the optimization of the logistics and cargo movement. It looks into the operation of AI-powered smart ports and how they make use of decision-making in the prediction of logistics, as well as the impact that digital transformation has on maritime logistics. The authors make a synthesis of literature from more than sixty-five sources and classify them regarding their innovations that aid in boosting the efficiency and competitiveness of ports.

3.2.22.2 Critical analysis of the research item:

Addresses a wide scope of modern port technological advances. Gives a review summary of newly surfaced issues. Analyzes the use of AI and IoT in ports.

However, it does not provide any form of statistical research or validation. Additionally, it does not sufficiently cover the implementation obstacles.

3.2.22.3 Relationship to the proposed research work:

PortFlow is in tune with smarter port developments because like other smart ports, it utilizes AI in tracking of the berths, thus increasing efficiency and predictive decision-making.

3.2.23 Automated Handling of Port Containers Using Machine Learning [23]

Below is the summary, critical analysis, and relevance of the selected article to the current research.

3.2.23.1 Summary:

This paper presents an AI container detection model for automating container identification and tracking at ports. TensorFlow Object Detection API and Python (OpenCV, Numpy, Matplotlib) are utilized for detecting and classifying containers with a 98

3.2.23.2 Critical analysis of the research item:

Demonstrates actual-world AI use for container handling. Produces very high accuracy in container identification and classification. Makes use of widely available machine learning libraries (TensorFlow, OpenCV, Numpy).

However, the proposed solution is only a prototype and lacks real-world validation. It focuses solely on container identification automation and not port operations in general.

3.2.23.3 Relationship to the proposed research work:

PortFlow can have AI-powered object detection integrated to provide real-time tracking of containers with minimal human interaction and efficient cargo track aside from berth allocation for a more automation based solutions in ports. This can be an extension of the project in Future work.

3.2.24 Advances in Terminal Management: Simulation of Vehicle Traffic in Container Terminals [24]

Below is the summary, critical analysis, and relevance of the selected article to the current research.

3.2.24.1 Summary:

The study suggests a microscopic traffic simulation model to study congestion on container terminal roads and vehicle flow optimization. The model is implemented for PSA Genova Pra' container terminal and includes an emission modeling module to estimate the effects on the environment. Aimsun® simulation software is used by the authors to study truck traffic flows, predict congestion trends and study different traffic management scenarios in port efficiency optimization.

3.2.24.2 Critical analysis of the research item:

Uses simulation modeling to forecast traffic congestion and its impacts. Provides real-world application by using the model to PSA Genova Pra', a major Italian port. Includes assessment of environmental effects through emissions modeling.

However, the scope is limited, as the study is done on terminal road traffic and not on port operations in general. Generalizability is an issue based on a single case study.

3.2.24.3 Relationship to the proposed research work:

PortFlow can leverage traffic simulation expertise in congestion prediction to maximize berth allocation and cargo transfer for effective vessel turnaround and reduced delays.

3.2.25 A Framework for Understanding Reliability in Container Shipping Networks [25]

Below is the summary, critical analysis, and relevance of the selected article to the current research.

3.2.25.1 Summary:

This paper provides a critical examination of the notion of reliability in container shipping networks, classifying it as infrastructure reliability, network configuration reliability and connectivity reliability. It draws attention to the effects of disruptions (e.g., COVID-19, Suez Canal blocking) and their effects on port operations and international trade. The authors perform a systematic literature review (SLR) to examine determinants of shipping network reliability and suggest a systematic framework for understanding and countering uncertainties in maritime logistics.

3.2.25.2 Critical analysis of the research item:

Offers a quantitative model for understanding shipping network reliability. Recognizes various factors influencing container shipping reliability. Uses a systematic literature review approach to keep a wide overview.

However, the article does not present an immediate solution to enhance reliability. It also lacks an empirical foundation or case study uses of the framework.

3.2.25.3 Relationship to the proposed research work:

PortFlow can incorporate this framework of reliability to enhance berth scheduling based on AI by including uncertainties such as congestion and disruptions, making decision-making more stable and adaptive.

3.2.26 Reinforcement-learning-based parameter adaptation method for particle swarm optimization [26]

Below is the summary, critical analysis, and relevance of the selected article to the current research.

3.2.26.1 Summary:

This study introduces a reinforcement-learning-based parameter adaptation method (RLAM) for Particle Swarm Optimization (PSO), utilizing the Deep Deterministic Policy Gradient (DDPG) algorithm. A new variant, RLPSO, is proposed to enhance PSO convergence by dynamically adjusting parameters like inertia weight and acceleration coefficients. Experimental validation using CEC 2013 benchmarks shows improved performance over standard PSO variants.

3.2.26.2 Critical analysis of the research item:

Effectively tackles PSO's parameter sensitivity through adaptive learning. RLAM enables automatic tuning without manual intervention, while DDPG supports continuous action spaces. Pretraining improves convergence and broadens applicability.

However, the work relies heavily on synthetic benchmarks and lacks real-world deployment. Computational complexity from neural training is high and its robustness in constrained optimization is not addressed.

3.2.26.3 Relationship to the proposed research work:

This study supports our AI-based berth allocation research by demonstrating how reinforcement learning can adapt optimization strategies in uncertain environments. RLPSO principles may be useful for dynamic scheduling when vessel traffic is unpredictable.

3.2.27 Efficient Exploration through Bayesian Deep Q-Networks [27]

Below is the summary, critical analysis, and relevance of the selected article to the current research.

3.2.27.1 Summary:

The paper proposes Bayesian Deep Q-Networks (BDQN), a reinforcement learning method that combines Double DQN with Bayesian linear regression in the output layer. This structure allows Thompson sampling over Q-values to improve exploration. BDQN is tested on Atari games and shows faster learning and higher rewards compared to DDQN.

3.2.27.2 Critical analysis of the research item:

Strength lies in improved exploration using uncertainty estimates while preserving DDQN's efficiency. Minimal architectural changes make BDQN easy to implement. Experiments confirm better performance on standard benchmarks.

Limitations include evaluation restricted to games and reliance on linear Q-layer assumptions. Practical use in complex domains is not explored and details like update frequency may affect reproducibility.

3.2.27.3 Relationship to the proposed research work:

As our focus is AI-based berth optimization, this paper highlights the value of uncertainty-aware decision-making. BDQN concepts could inform adaptive berth scheduling where uncertain vessel arrivals require intelligent exploration.

3.3 Literature Review Summary Table

The following table provides the comprehensive summary of the literature review

Table 3.1: Summary of Relevant Studies for the Research Project

Author(s)	**Method**	**Results**	**Limitations**
Ai et al. [1]	Deep Reinforcement Learning (PS-D3QN)	Improved berth and yard scheduling efficiency, reduced vessel waiting time	High computational demands, requires large training datasets
Dai et al. [2]	RL model considering quay crane setup times	RL-based approach reduces vessel waiting time significantly	Lacks real-world implementation, only mathematical validation
Wang et al. [3]	Deep Q-Network (DQN)	57% reduction in vessel waiting time compared to GA and ACO	High computational power requirement, sensitive to data quality
Li et al. [4]	Dueling Double Deep Q-Network (D3QN)	3.7% improvement over traditional optimization techniques	Computationally expensive, requires large datasets
Dulebenets et al. [5]	Self-adaptive Evolutionary Algorithm (EA)	Optimized scheduling parameters for better efficiency	Lacks deep learning integration, requires careful parameter tuning
Arango et al. [6], Akinnuwesi et al. [7]	Genetic Algorithm (GA)	Improved quay crane scheduling and service times, optimization of quay length vs. handling time	GA struggles with real-time adaptation, ignores vessel arrival uncertainties
Aslam et al. [8], Aslam et al. [10], Aslam et al. [11]	Cuckoo Search Algorithm (CSA)	Outperforms GA and MILP methods, reduces computational cost compared to MILP	Does not integrate deep learning techniques, highly sensitive to parameter tuning

Continued on next page

Continued from previous page

Author(s)	**Method**	**Results**	**Limitations**
Umang et al.[9]	Mixed Integer Linear Programming (MILP)	Reduces vessel waiting time with real-world data	Not applicable to container terminals
Ursavas[12]	Decision Support System (DSS)	Reduces waiting time through priority scheduling	Lacks AI-based dynamic adjustments
Li et al. [13]	Literature review on berth allocation techniques	Identifies research gaps in AI-based berth optimization	No experimental validation
Agra et al. [14]	Distributionally Robust Optimization (DRO)	Minimizes worst-case expected delays	Does not utilize AI-driven decision-making
Yu et al. [15]	Parallel Ant Colony Optimization (PACO)	Outperforms standard ACO	Dependent on parameter tuning
Razzaghi et al. [16]	Survey on RL in aviation applications	Identifies relevant RL methods for logistics optimization	Lacks real-world deployment cases
Budipriyanto et al. [17]	Simulation-based collaborative berth allocation model	Reduces waiting time through resource sharing	Limited generalizability beyond case study
Kolley et al. [18]	ML-based vessel arrival time prediction (Linear Regression, ANN, KNN)	Improves scheduling robustness using Dynamic Time Buffers (DTBs)	Does not encompass metaheuristic algorithm and RL in berth scheduling
Jahangard et al. [19]	ML-Optimization hybrid approach for port efficiency	Recommends predictive maintenance and AI-based scheduling	Lacks implementation details

Continued on next page

Continued from previous page

Author(s)	**Method**	**Results**	**Limitations**
Yadav [20]	AI-powered virtual port operator for optimization	Enhances vessel turnaround and berth occupancy	Focuses on general port operations rather than berth allocation
Chen et al. [21]	AI applications in maritime logistics	Discusses AI-driven berth scheduling	Lacks implementation details
Kosiek et al. [22]	Review of modern port technologies (AI, IoT, Big Data)	Highlights smart port automation trends	No statistical validation or implementation insights
Karim et al.[23]	AI-based container detection (TensorFlow, OpenCV)	98% accuracy in container tracking	Lacks real-world deployment
Carboni et al.[24]	Traffic simulation model for container terminal congestion	Predicts congestion trends, improves traffic flow	Limited to one case study
Yue et al.[25]	Framework for container shipping reliability	Identifies key factors affecting reliability	No empirical testing, lacks direct berth allocation insights
Yin et al.[26]	RL-based parameter tuning for PSO	Enhances PSO performance, adapts to uncertainty	High computational cost, lacks real-world validation
Azizzadenesheli et al.[27]	Bayesian Deep Q-Networks for exploration	Efficient exploration using uncertainty estimates	Limited to simulated games, not applied in port settings

3.4 Conclusion

Berth Allocation Problem is a dynamic and complex problem of maritime logistics that calls for innovative solutions to enhance port responsiveness and efficiency. Traditional optimization and heuristic solutions, although very effective in specific scenarios, lack real-time responsiveness and scalability. The innovative AI-based methods, i.e., reinforcement learning and deep learning, are poised to address the aforementioned shortcomings. The methods are capable of generating real-time decisions, predictive modeling and dynamic resource allocation and therefore are most suited to modern port operations.

Still, the challenges persist in the form of the requirement of large training data, computational resources and compatibility with legacy port systems. Future research would need to concentrate on hybrid approaches that can leverage the advantages of both classical optimization techniques and AI-based techniques and their performance validated with real-world implementations. By applying the advancements in AI and IoT, ports can enable higher efficiency, sustainability and competitiveness in global trade.

Chapter 4 Software Requirement Specifications

This chapter presents a thorough analysis of the software requirements and design specifications for the AI-based berth allocation system, PortFlow. It specifies the functional requirements, system features, design constraints and other technical considerations needed to reproduce and implement the system. The chapter acts as a guide for developers, port operators and system integrators to understand clearly how the system operates, its functionalities and how it interacts with users.

4.1 List of Features

The AI-driven berth allocation system includes several key features designed to improve port efficiency, minimize vessel waiting times and enhance decision-making for port authorities. These features integrate real-time data processing, machine learning optimization and an interactive dashboard to provide a seamless berth scheduling experience.

- Automatically assigns berths to vessels based on real-time data and predictive analytics.
- Adjusts berth assignments in response to vessel delays, operational changes, or priority shifts.
- Overlooks port congestion and suggests berth assignments accordingly.
- Displays real-time berth occupancy, vessel schedules and AI-generated recommendations.
- Allows operators to override AI decisions and manually adjust berth assignments when needed.
- Can be deployed across different ports with various berth configurations and cargo types.
- Designed to handle increasing vessel traffic without performance degradation.

These features ensure seamless integration of AI-driven automation with human oversight, allowing for a smarter, more efficient berth management system.

4.2 Functional Requirements

This section outlines how the artificial intelligence-based berth allocation system (PortFlow) operates and interacts with users in order to optimize berth scheduling, reduce vessel waiting times and improve port operations. Every requirement ensures that the system operates optimally under real-time constraints, dynamic scheduling alterations and the need for manual intervention.

4.2.1 AI-Based Berth Scheduling

The system shall automatically assign berths to incoming vessels based on real-time ETA, vessel size, cargo type and berth availability. AI models shall evaluate both historical data and live port conditions to ensure efficient berth utilization. The scheduling algorithm shall prioritize:

- Minimizing vessel waiting times by selecting the most optimal berth.
- Maximizing berth utilization by considering vessel size, berth length and draft limitations.
- Handling cargo-specific requirements, ensuring that liquid cargo ships, bulk carriers and container vessels are allocated berths with the necessary handling equipment.
- Prioritizing emergency or high-value shipments when necessary, ensuring these vessels are allocated berths with minimal delay.

The system shall update berth assignments every few minutes to reflect changes in port conditions, vessel movements and berth availability.

4.2.2 Dynamic Berth Reallocation

The system shall continuously monitor vessel schedules and reallocate berths dynamically if unexpected changes occur. This includes:

- Delayed or early vessel arrivals requiring berth assignments to be adjusted.
- Priority changes, such as an emergency vessel requiring immediate docking.
- Operational issues, such as berth maintenance or equipment failures, requiring vessels to be reassigned elsewhere.

AI models shall analyze these factors and generate reallocation strategies, which are then either automatically implemented or sent to port operators for approval based on pre-set automation thresholds.

4.2.3 Interactive Dashboard

The system shall provide a real-time dashboard that offers port operators and terminal managers:

- Live berth occupancy updates, showing which berths are in use and which are available.
- Vessel tracking and scheduling data, displaying estimated and actual arrival/departure times.
- AI-generated berth allocation recommendations, allowing operators to review system decisions.
- Congestion alerts, warning operators when certain areas of the port are becoming overloaded.

- Manual berth adjustment controls, enabling users to override AI decisions if necessary.

The dashboard shall be accessible via desktop and tablet interfaces to ensure real-time monitoring for port staff.

4.2.4 Historical Data Analysis

The system shall store, process and analyze past berth utilization trends, vessel turnaround times and congestion levels to:

- Improve berth scheduling predictions based on past efficiency metrics.
- Identify bottlenecks and inefficiencies, allowing for future improvements in scheduling logic.

Users shall be able to filter reports by date range, vessel type and berth utilization metrics for detailed insights.

4.2.5 Manual Override and Custom Scheduling

While AI optimizes berth scheduling, port operators shall have full manual override capabilities. The system shall allow:

- Direct manual berth assignment, enabling operators to override AI recommendations.
- Adjusting vessel priorities, in cases where human judgment is needed for emergency or special cargo shipments.
- Locking specific berths for maintenance or reserved docking, ensuring that these spaces are not automatically assigned.

These functional requirements guarantee PortFlow's efficiency, responding to real-time changes, maximizing berth usage, anticipating congestion and delivering in-depth analytical information. Through the combination of AI-based automation and human monitoring, the system optimizes decision-making, scheduling effectiveness and port operation management, ultimately resulting in shorter waiting times, decreased fuel consumption and enhanced logistics performance.

4.3 Quality Attributes

The berth allocation system (PortFlow) powered by AI is intended to be highly performing, scalable, reliable, secure, usable and extensible. These quality characteristics determine the system's non-functional properties, which guarantee that it is efficient, flexible and easy to use when dealing with massive port operations.

4.3.1 Performance

The system will handle real-time rescheduling and berth allocation requests effectively, maintaining low response times and decision-making delays. The system should also be able to handle large volumes of incoming vessel data without reduction in speed. The AI model will handle berth assignments in near real-time, ensuring that updates to berth scheduling happen within seconds even during periods of high traffic. Performance tests will also ensure that the system does not degrade under normal and high-load conditions and remains responsive.

4.3.2 Scalability

The system should accommodate growth in port operations without sacrificing speed or efficiency. With an increase in the number of vessels, ports and users, the system needs to scale. The system should be able to manage higher real-time data inputs, larger vessel databases and higher levels of simultaneous user interactions without slowing down. The architecture will provide a distributed cloud deployment capability to allow more computational resources to be attached as the need arises.

4.3.3 Reliability

PortFlow will provide precise berth assignments and forecast congestion analysis in any operational scenario. The system should run 24/7 with little downtime and include backup mechanisms in the event of hardware or network failure. AI models will be periodically validated to ensure berth assignments remain fair, consistent and optimized. In the event of system failures, automated recovery procedures will ensure operations continue uninterrupted without data loss or mis-scheduling.

4.3.4 Security

The system will utilize robust authentication, encryption and access control methods to secure unauthorized access to sensitive port information. Berth scheduling and vessel data will be stored and transmitted securely with industry-standard encryption methods. Role-based access control (RBAC) will restrict berth assignment or override of AI-driven decisions to only authorized personnel. Regular security audits and penetration testing will also be performed to identify and mitigate possible vulnerabilities.

4.3.5 Usability

PortFlow will be made with an easy-to-use interface that can be easily operated by port operators, logistics managers and ship operators. The dashboard will have simple data visualization, real-time berth monitoring and easy-to-understand AI suggestions, so that it is easily accessible to users with different levels of technical know-how. The system will include tooltips, user manuals and responsive design so that it works smoothly on web.

4.3.6 Extensibility

The architecture of the system will be modular and flexible, making it possible to incorporate new capabilities and features without much rework. Future updates, like integration of automated customs clearance, cargo handling predictions with AI, or third-party logistics API integrations, should be simple to implement. Developers should be able to add, modify, or swap out components without impacting the main functionality, allowing the system to remain flexible with changing port needs.

4.4 Non-Functional Requirements

The following section presents the non-functional requirements of the AI-based berth allocation system (PortFlow), which will guarantee that the system is efficient, reliable and secure with high usability and scalability for large-scale port operations.

4.4.1 Performance Requirements

The system should handle berth allocation requests and rescheduling of vessels within 5 to 10 seconds in standard conditions. The system should handle concurrent requests from a minimum of 100 users, updating the berth assignments in real time without slowing down the system. AI-based decision-making should not delay manual override action, enabling the operators to reconfigure berth schedules in real time.

4.4.2 Reliability Requirements

The system uptime should be at least 99.9%, guaranteeing berth scheduling activities are always accessible. There should be automatic fault tolerance capabilities to bounce back from errors without significant disruptions. If there are system failures, users should get unambiguous messages without loss of data or wrong assignment of schedules.

4.4.3 Scalability Requirements

The system will be able to manage growing port operations, provide large-scale vessel traffic without performance detriment. It will have to support thousands of active vessels, berth allocations and port-wide scheduling requests to provide seamless functionality as more data is added. Cloud deployment shall provide on-demand scalability so that new ports can be added without incurring major changes in infrastructure.

4.4.4 Usability Requirements

The interface will be intuitive and easy to use, enabling port operators, terminal managers and logistics staff to use the system easily. Tooltips, color-coded warnings and well-defined action buttons will facilitate user interaction. The system will give real-time feedback and error messages to assist users in solving scheduling conflicts efficiently.

4.4.5 Requirements for Maintainability

System upkeep will never take more than two to three hours a month, resulting in minimal downtime. Updates, bug corrections and feature additions will be done modulename-by-modulename, such that repairing one component of the system does not involve shutting down the entire system. Monitoring tools and logs will monitor performance problems to aid in rapid troubleshooting and patching.

4.4.6 Requirements for Accuracy

The berth allocation system driven by AI needs to have at least 95% accuracy in scheduling decisions at the berth. Machine learning algorithms will be regularly fine-tuned to minimize prediction errors in berth assignments. The system will double-check berth assignments with real-time operational limitations to provide accurate scheduling at all times.

4.4.7 Availability Requirements

The system should be up and running 24/7, so that berth scheduling operations are not interrupted. Scheduled maintenance will be carried out during off-peak hours and any downtime should not be more than 30 minutes for critical upgrades. Failover provisions should be available to fail over to backup systems in the event of a catastrophic failure, so that operation is not disrupted.

4.5 Assumptions

The following assumptions have been made in specifying the specifications for PortFlow, the AI-based berth allocation system:

- Because real-time data is not available, berth allocation will be experimented with in a simulated environment, where historical data will be employed to simulate real-world conditions but it is assumed that minor adjustments will allow it to scale to real time integration.

- The system takes for granted that historical records are comprehensive, correct and representative of real port activity. Incomplete or out-of-date records might affect model performance.
- It is taken assumes that trained AI models will perform well on new, unseen data following historical patterns. Unforeseen port disruptions not in historical data may not be well predicted.
- Even being AI-powered, it is assumed that port authorities will examine berth allocation results for conformity with operational constraints and correct manually if necessary.
- The system should be efficient in processing high volumes of data, such that performance does not suffer when running thousands of berth history records.
- Data privacy and security controls will be assumed in place to keep historical berth utilization data secure against unauthorized access.
- It is anticipated that port operators, terminal managers and logistics personnel will be trained to operate the system effectively, reducing errors in manual overrides.
- As ports are dynamic environments, the system anticipates that externalities like weather delays, mechanical breakdowns and customs processing times will be factored into real-time adjustments.

These assumptions serve to outline the extent and viability of the system and guarantee that its AI models and dashboard insights are still useful in the lack of real-time data.

4.6 Use Cases

Our system will have the following use cases.

4.6.1 User Login and Authentication

<table>
<tr><td colspan="2">Name</td><td colspan="2">User Login & Authentication</td></tr>
<tr><td colspan="2">Actors</td><td colspan="2">Port Operator, Terminal Manager, Administrator</td></tr>
<tr><td colspan="2">Summary</td><td colspan="2">Only authorized users can access the system by logging in with valid credentials. The system verifies login details and grants access based on user roles.</td></tr>
<tr><td colspan="2">Pre-Conditions</td><td colspan="2">The user must be registered in the system with an assigned role (e.g., Operator, Manager, Admin).
The system must have a secure authentication mechanism in place.</td></tr>
<tr><td colspan="2">Post-Conditions</td><td colspan="2">The user is successfully logged in and redirected to their respective dashboard. Unauthorized access attempts are denied.</td></tr>
<tr><td colspan="2">Special Requirements</td><td colspan="2">The system must use secure encryption for password storage and support role-based access control (RBAC).</td></tr>
<tr><td colspan="4">Basic Flow</td></tr>
<tr><td colspan="2">Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>1</td><td>User enters email and password.</td><td>2</td><td>System validates credentials against the database.</td></tr>
<tr><td>3</td><td>If credentials are valid, access is granted.</td><td>4</td><td>User is redirected to their designated dashboard.</td></tr>
<tr><td>5</td><td>System logs the successful login.</td><td>6</td><td>Login activity is recorded for security auditing.</td></tr>
<tr><td colspan="4">Alternative Flow</td></tr>
<tr><td>A1</td><td>User enters incorrect password.</td><td>A1-1</td><td>System displays error:
Incorrect email or password.</td></tr>
<tr><td>A2</td><td>Multiple failed login attempts.</td><td>A2-1</td><td>System temporarily locks account after three failed attempts.</td></tr>
<tr><td>A3</td><td>Unauthorized user attempts login.</td><td>A3-1</td><td>System denies access and logs the failed attempt.</td></tr>
<tr><td>A4</td><td>User forgets password.</td><td>A4-1</td><td>System provides password reset option via email.</td></tr>
</table>

4.6.2 Automatic Berth Allocation

<table>
<tr><td>Name</td><td colspan="3">Automatic Berth Allocation</td></tr>
<tr><td>Actors</td><td colspan="3">Port Authority, Terminal Operator</td></tr>
<tr><td>Summary</td><td colspan="3">The system automatically assigns an optimal berth to an incoming vessel based on AI-driven scheduling. The allocation considers vessel ETA, cargo type and berth availability.</td></tr>
<tr><td>Pre-Conditions</td><td colspan="3">Vessel arrival data must be available in the system. The port must have at least one vacant berth that meets the vessel's requirements.</td></tr>
<tr><td>Post-Conditions</td><td colspan="3">The vessel is successfully assigned a berth and the berth status is updated in the system.</td></tr>
<tr><td>Special Requirements</td><td colspan="3">The system should process berth assignments within 5–10 seconds to ensure operational efficiency.</td></tr>
<tr><td colspan="4">Basic Flow</td></tr>
<tr><td colspan="2">Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>1</td><td>System detects incoming vessel (ETA, cargo type).</td><td>2</td><td>AI retrieves vessel details and checks berth availability.</td></tr>
<tr><td>3</td><td>System processes berth assignments using AI.</td><td>4</td><td>Optimal berth selected based on real-time and historical data.</td></tr>
<tr><td>5</td><td>System confirms berth allocation.</td><td>6</td><td>Berth assignment displayed on dashboard for review.</td></tr>
<tr><td colspan="4">Alternative Flow</td></tr>
<tr><td>A1</td><td>No berths available.</td><td>A1-1</td><td>Vessel placed in waiting queue; next berth suggested.</td></tr>
<tr><td>A2</td><td>High-priority vessel requires berthing.</td><td>A2-1</td><td>System reallocates berths, prioritizing urgent vessel.</td></tr>
<tr><td>A3</td><td>Operator overrides AI recommendation.</td><td>A3-1</td><td>Manual change logged; future AI predictions adjusted.</td></tr>
</table>

4.6.3 Manual Berth Assignment

<table>
<tr><td>Name</td><td colspan="3">Manual Berth Assignment</td></tr>
<tr><td>Actors</td><td colspan="3">Port Authority, Terminal Operator</td></tr>
<tr><td>Summary</td><td colspan="3">An operator manually assigns a berth to a vessel
instead of using the AI-generated recommendation.</td></tr>
<tr><td>Pre-Conditions</td><td colspan="3">The vessel must be registered in the system.
The berth must be available and meet the vessel’s
requirements.</td></tr>
<tr><td>Post-Conditions</td><td colspan="3">The vessel is assigned to the manually selected berth,
and the system logs the decision.</td></tr>
<tr><td>Special
Requirements</td><td colspan="3">Operators should have a manual override option with
real-time berth status updates.</td></tr>
<tr><td colspan="4">Basic Flow</td></tr>
<tr><td colspan="2">Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>1</td><td>Operator selects a vessel from the schedule.</td><td>2</td><td>System displays berth availability
and AI-recommended options.</td></tr>
<tr><td>2</td><td>Operator manually selects a berth.</td><td>3</td><td>System confirms the assignment and
updates the schedule.</td></tr>
<tr><td>3</td><td>System logs the manual decision.</td><td>4</td><td>Future AI berth assignments learn from
operator preferences.</td></tr>
<tr><td colspan="4">Alternative Flow</td></tr>
<tr><td>1-A</td><td>Selected berth is occupied.</td><td>2-A</td><td>System alerts the operator and suggests
alternative berths.</td></tr>
<tr><td>1-B</td><td>Vessel is delayed.</td><td>2-B</td><td>System allows the operator to reschedule
the berth assignment.</td></tr>
</table>

4.6.4 Berth Utilization Reports and Historical Analytics

Name	Berth Utilization Reports & Historical Analytics		
Actors	Port Operator, Terminal Manager, Port Authority		
Summary	The system provides historical reports on berth utilization, vessel turnaround times and congestion trends. Users can analyze past data to improve future berth scheduling strategies.		
Pre-Conditions	The system must have stored historical berth allocation data and a reporting module. The user must be logged in to access reports.		
Post-Conditions	The user views historical berth usage, vessel trends and efficiency insights. Reports can be filtered by date range, vessel type and berth occupancy rates.		
Special Requirements	Reports should be exportable (CSV, PDF) for external analysis. The system should provide graphical insights (charts, tables) for easy interpretation.		
Basic Flow			
Actor Action		**System Response**	
1	User accesses Reports section.	2	System loads historical berth usage data.
3	User selects date range and filters.	4	System retrieves and displays filtered berth utilization reports.
5	User views AI insights on efficiency.	6	System provides performance metrics, turnaround times and congestion patterns.
7	User downloads the report.	8	System generates exportable report (PDF, CSV, etc.).
Alternative Flow			
A1	No data for selected filters.	A1-1	System displays error: No records found for the selected date range.
A2	User requests custom visualization.	A2-1	System allows switching between bar charts, heatmaps and tables.

4.7 Hardware and Software Requirements

The hardware and software requirements necessary for the development and deployment of the project are listed below.

4.7.1 Hardware Requirements

- GPU for processing and training of our model.
- A server with a powerful CPU and sufficient RAM.
- An internet-capable system for real-time data updates and API interactions if needed.

4.7.2 Software Requirements

The software stack for PortFlow consists of AI development tools, backend frameworks, frontend technologies and database management systems to ensure a scalable, secure and efficient port management system.

- For Operating System we will be using Windows 10/11 or macOS
- Python (for AI model development)
- TensorFlow / PyTorch (for deep learning-based berth allocation)
- Node.js / Express.js (for backend API development)
- React.js / Vue.js (for building the interactive dashboard)
- PostgreSQL / MySQL (for structured berth allocation data)
- MongoDB (for storing unstructured logs and real-time events)
- JWT (JSON Web Tokens) for secure user authentication
- OAuth 2.0 for API access control
- SSL Encryption for secure data transmission
- GitHub / GitLab for source code management
- JIRA / Trello for agile development tracking

These hardware and software requirements ensure that PortFlow operates efficiently across development, deployment and real-time usage scenarios. The AI-driven system requires powerful computing resources, while the dashboard and backend services rely on scalable cloud-based infrastructure to deliver real-time berth allocation and analytics.

4.8 Graphical User Interface

This section gives GUI dumps of each screen. These are subject to change as the development process is ongoing but gives general insights of what the system will be looking like.

Users log in with their credentials and select their role (Operator or Manager). Role selection determines available features and permissions.

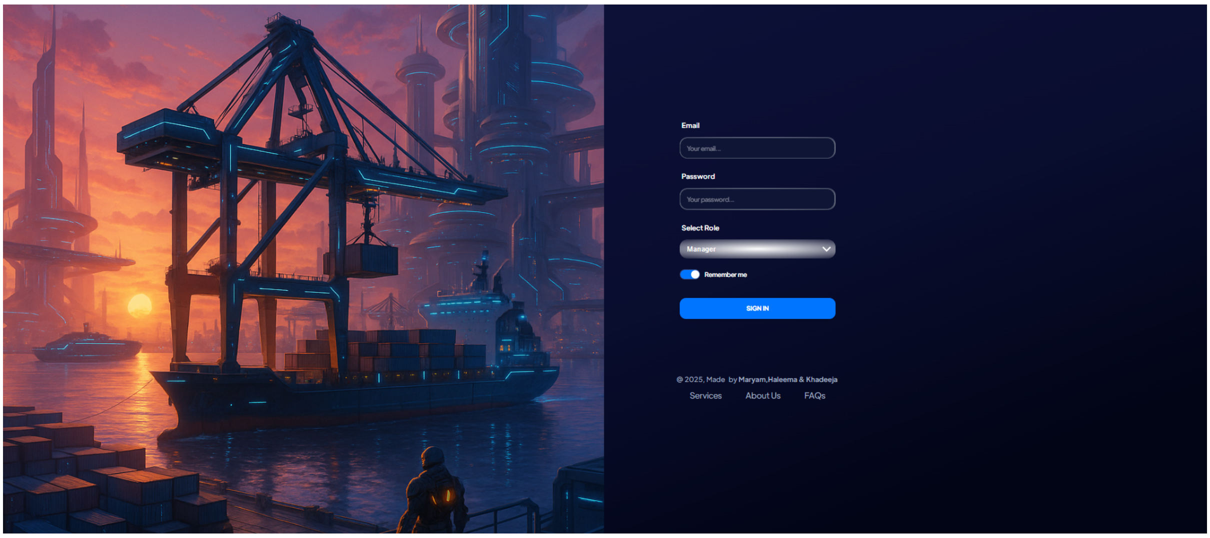

Figure 4.1: Landing Page - Login Screen

The central hub where users see an overview of system updates. Available options and functionalities depend on the user's role.

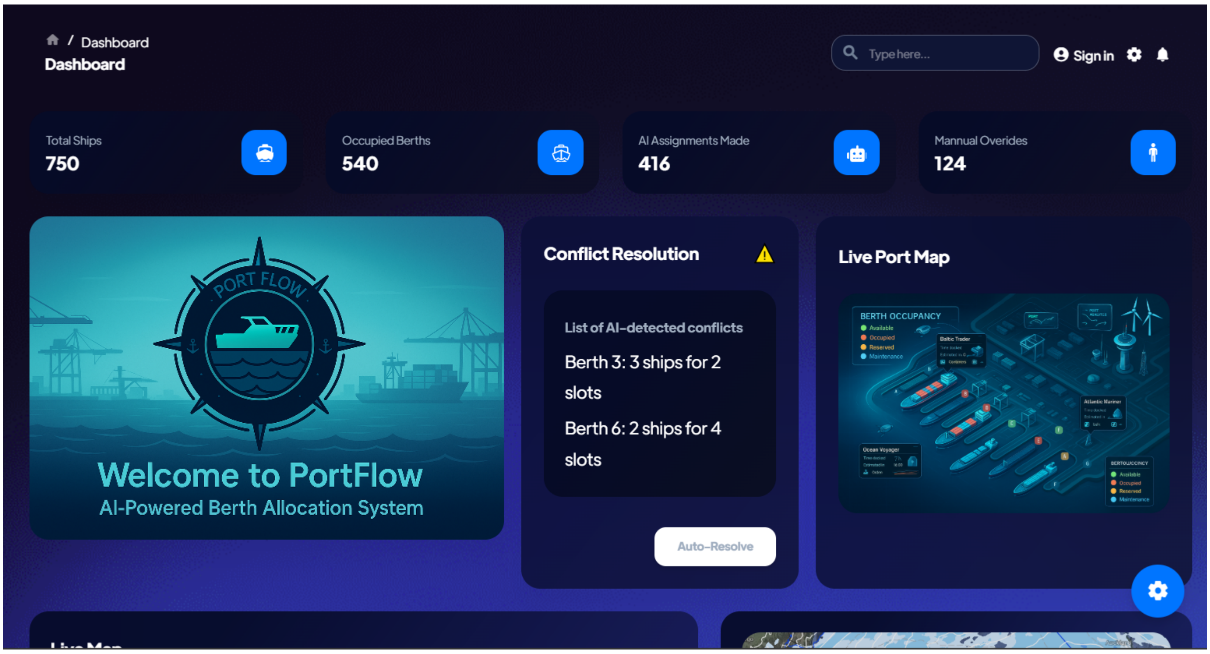

Figure 4.2: Port Operations – Live Monitoring and Management

AI suggests optimal berth assignments based on ship size, arrival time and port congestion on the following screen. Operators can accept or adjust assignments as needed.

Figure 4.3: Smart Berth Allocation – AI-Powered Recommendations and Berth status

The following screen shows more of the dashboard having live map and weather reports for insights

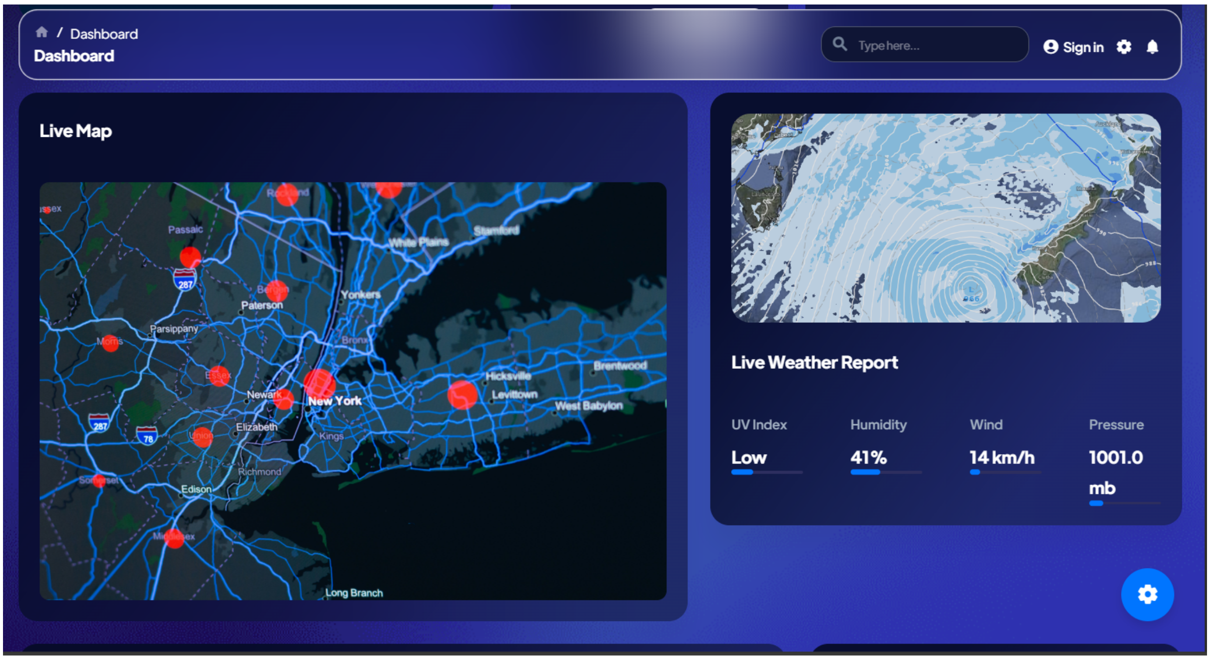

Figure 4.4: Live map and weather status

The following screen shows real-time ship schedules, delays and berth allocations that can be reviewed by relevant operators. It allows manual berth assignment when AI suggestions are not ideal.

Figure 4.5: Live Ship Schedule and Berth assignment overrides

The following screen shows Berth utilization metrics for relevant insights.

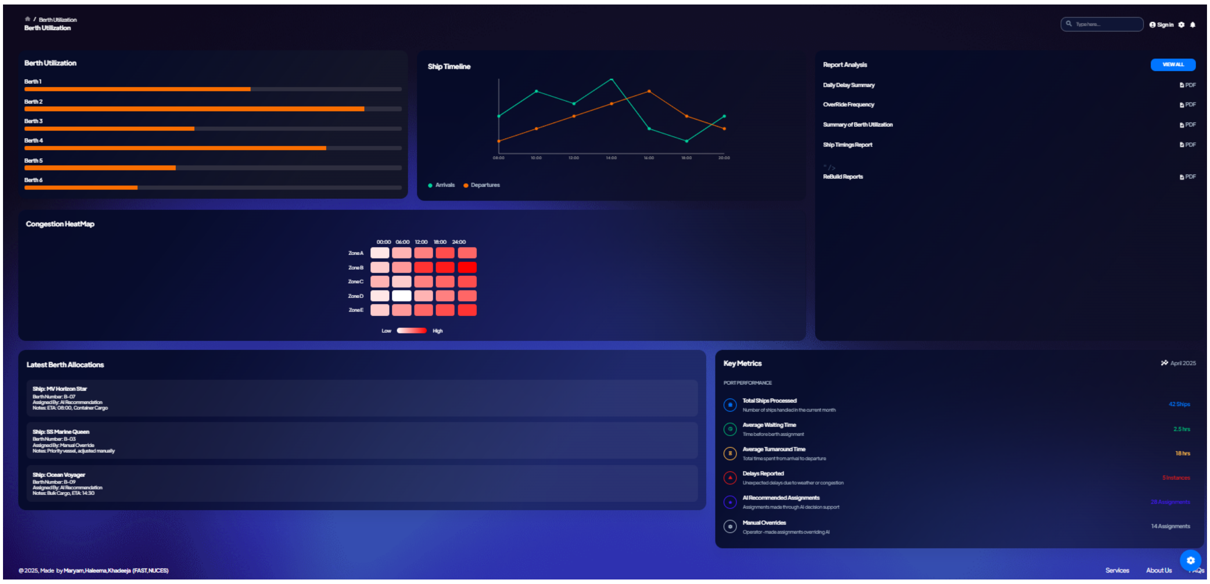

Figure 4.6: Berth Utilization – Track performance

The following screen shows a panel for adding managers or operators by the port authorities and admin control over ships and berths.

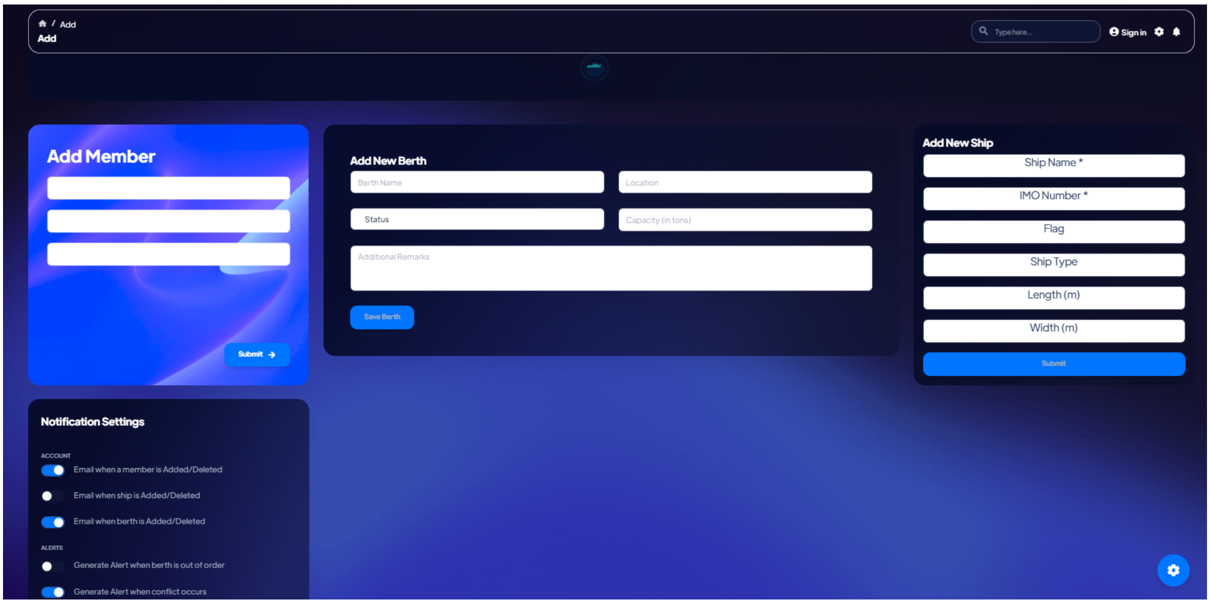

Figure 4.7: Add panel-For port admin use

4.9 Database Design

This section defines the database structure for managing users, ships, berths and schedules. It ensures efficient data storage and retrieval to support system operations.

4.9.1 ER Diagram

Below is the ER Diagram.

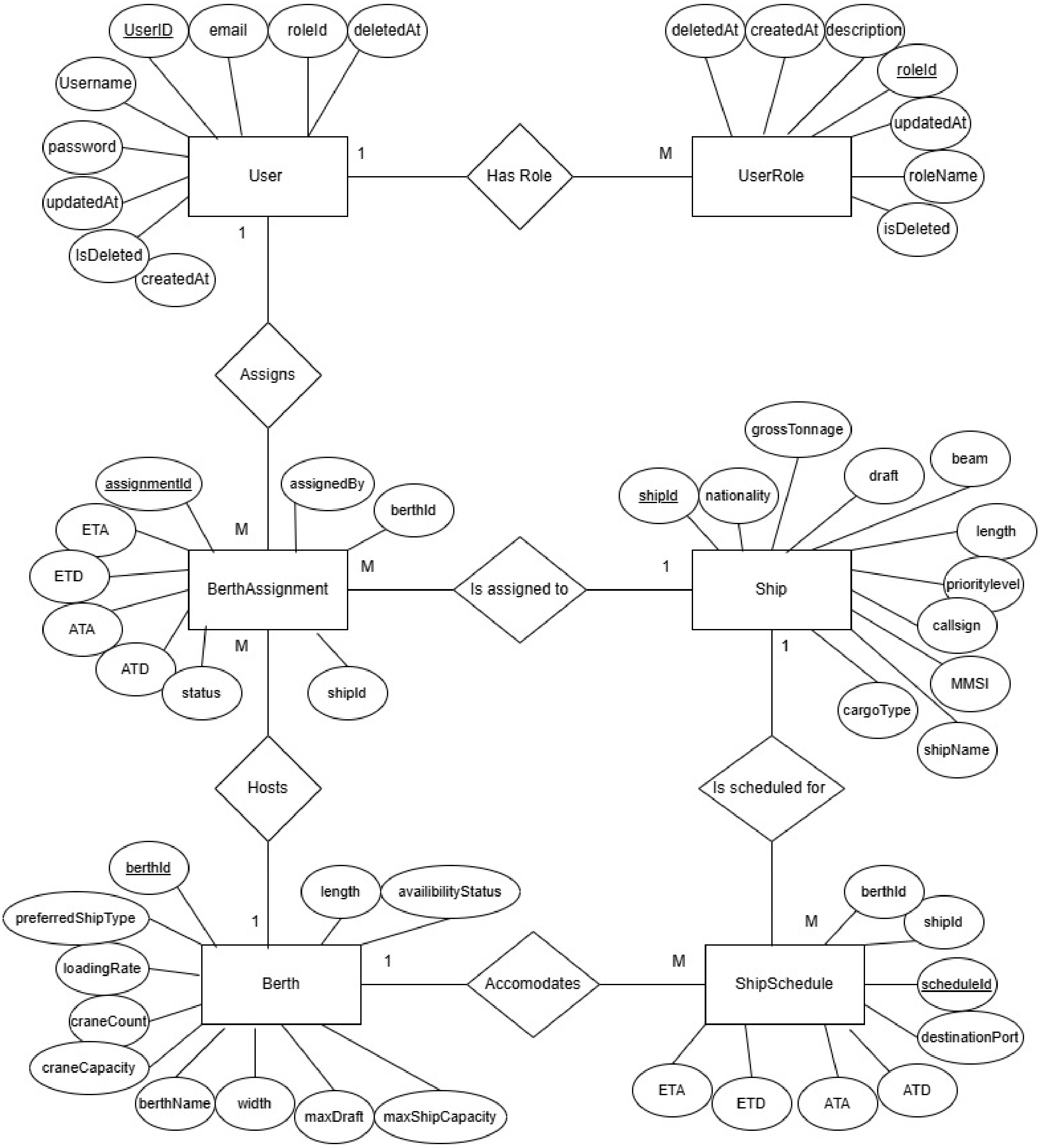

Figure 4.8: ER Diagram of PortFlow.

4.9.2 Data Dictionary

Table 4.1: User Attributes Table

Attribute	Data Type	Relation To	Description
`userId`	VARCHAR	-	Unique identifier for each user
`username`	VARCHAR	-	Name of the user
`email`	VARCHAR	-	Email address of the user
`passwordHash`	VARCHAR	-	Hashed password of the user
`roleId`	INT	`UserRole`	Role of the user in the system
`isDeleted`	BOOLEAN	-	Indicates if the user account is deleted
`createdAt`	TIMESTAMP	-	Time of user creation
`updatedAt`	TIMESTAMP	-	Time of user information update
`deletedAt`	TIMESTAMP	-	Time of user deletion

Table 4.2: UserRole Table

Attribute	Data Type	Relation To	Description
`roleId`	INT	-	Unique identifier for each role
`roleName`	VARCHAR	-	Unique name of the role
`description`	TEXT	-	Information about role permissions
`isDeleted`	BOOLEAN	-	Indicates if the role is deleted
`createdAt`	TIMESTAMP	-	Time of role creation
`updatedAt`	TIMESTAMP	-	Time of role information update
`deletedAt`	TIMESTAMP	-	Time of role deletion

Table 4.3: Ship Table

Attribute	Data Type	Relation To	Description
`shipId`	VARCHAR	-	Unique identifier for each ship
`shipName`	VARCHAR	-	Name of the ship
`MMSI`	VARCHAR	-	Maritime Mobile Service Identity
`callSign`	VARCHAR	-	Type of ship (Container, Tanker, etc.)
`length`	FLOAT	-	Ship length in meters
`draft`	FLOAT	-	Ship draft in meters
`grossTonnage`	FLOAT	-	Gross tonnage of the ship
`commodity`	VARCHAR	-	Type of cargo carried
`nationality`	VARCHAR	-	Ship's country flag

Table 4.4: Ship Schedule Table

Attribute	Data Type	Relation To	Description
`scheduleId`	VARCHAR	-	Unique identifier for each schedule
`shipId`	VARCHAR	Ship	ID of the ship
`berthId`	VARCHAR	Berth	ID of the assigned berth (if assigned)
`destinationPort`	VARCHAR	-	Destination port of the ship
`Anchorage time`	TIMESTAMP	-	Actual Time of Arrival
`Sailing time`	TIMESTAMP	-	Actual Time of Departure
`Berthing time`	TIMESTAMP	-	Actual Time of Berthing
`Waiting Time`	VARCHAR	-	Time waited till Berth Assignment

Table 4.5: Berth Table

Attribute	Data Type	Relation To	Description
`berthId`	VARCHAR	-	Unique identifier for each berth
`berthName`	VARCHAR	-	Name of the berth
`length`	FLOAT	-	Length of the berth in meters
`maxDraft`	FLOAT	-	Maximum draft allowed at the berth
`maxShipCapacity`	FLOAT	-	Maximum ship capacity (tonnage)
`availabilityStatus`	BOOLEAN	-	Status of berth (Occupied/Free)
`preferredCommodityType`	VARCHAR	-	Ship types preferred for this berth
`loadingRate`	FLOAT	-	Loading/unloading rate in tons per hour
`craneCount`	INT	-	Number of cranes assigned
`craneCapacity`	FLOAT	-	Total crane capacity

Table 4.6: BerthAssignment Table

Attribute	Data Type	Relation To	Description
`assignmentId`	VARCHAR	-	Unique identifier for each assignment
`shipId`	VARCHAR	Ship	ID of the assigned ship
`berthId`	VARCHAR	Berth	ID of the assigned berth
`assignedBy`	VARCHAR	User	User who assigned the berth
`Anchorage time`	TIMESTAMP	-	Actual Time of Arrival
`Sailing time`	TIMESTAMP	-	Actual Time of Departure
`status`	VARCHAR	-	Status of assignment (Scheduled, Completed, Canceled)

4.10 Risk Analysis

The following is a list of risks of the berth allocation system:

4.10.1 Technical Risk

Following are some technical risks, associated with the development and maintenance of system:

- Inaccurate vessel arrival time or berth availability data may cause inefficient scheduling.
- The AI system needs constant availability; hardware or software failures might interfere with

operations.

- As the size of the port increases, the system can suffer performance bottlenecks unless optimized suitably.
- The AI model might favor some vessels over others unfairly because of biased historical data.
- The system can have difficulty integrating with currently installed port management software and APIs.

4.10.2 Ethical and Legal Risk

Following are some Ethical risks, associated with the development and maintenance of system:

- Conflicts can occur if berth allocations prefer certain shipping companies inequitably.
- Sensitive shipping and port information needs to be protected from unauthorized use.
- The system needs to comply with international maritime and port authority laws.

4.10.3 Dataset-Related Risk

Following are some Dataset related risks, associated with the development and maintenance of system:

- An insufficient dataset will lead to poor berth predictions.
- Incorrect or missing data will cause AI to make bad decisions.

4.10.4 Business Risks

Here are some Business related risks, associated with our system:

- Certain ports might be reluctant to make a transition from manual berth allocation to an AI-based system.
- It can take substantial initial investment to implement AI and cloud infrastructure.

4.10.5 Operational Risks

Following are some Operational risks, associated with our system:

- In case PortFlow has unplanned outages, port operations could be affected.
- If port operators override AI suggestions incorrectly, berth scheduling might become inefficient.

4.11 Conclusion

This chapter described the Software Requirements Specifications (SRS) of the berth allocation system, including the functional requirements, design of the database, hardware and software requirements and possible risks. The use cases showcase how the system automates the allocation of berths, facilitates manual overriding and offers historical analysis in order to enhance decision-making. Moreover, risk analysis determines the technical, ethical, business and operational difficulties that need to be overcome in order to implement it smoothly. Such specifications guarantee that the system is effective, scalable and versatile across various port conditions.

Chapter 5 Proposed Approach and Methodology

Port berth allocation is a dynamic process that must be efficiently scheduled to minimize vessel waiting times, prevent congestion and maximize berth usage. Conventional heuristic-based scheduling cannot keep up with changing vessel arrival rates, mixed cargo types and variable delays. This research uses AI-driven berth allocation models developed from historical port data to mimic real-time action and optimize berth schedules. The method combines RL along with Evolutionary Optimization to achieve optimized and robustness in berth allocations.

This chapter explains the methodology used in the development and testing of the AI-based berth allocation system. The research utilizes reinforcement learning to make dynamic berth allocation decisions and evolutionary algorithms for optimization. Following figure provides a general overview of the proposed methodology.

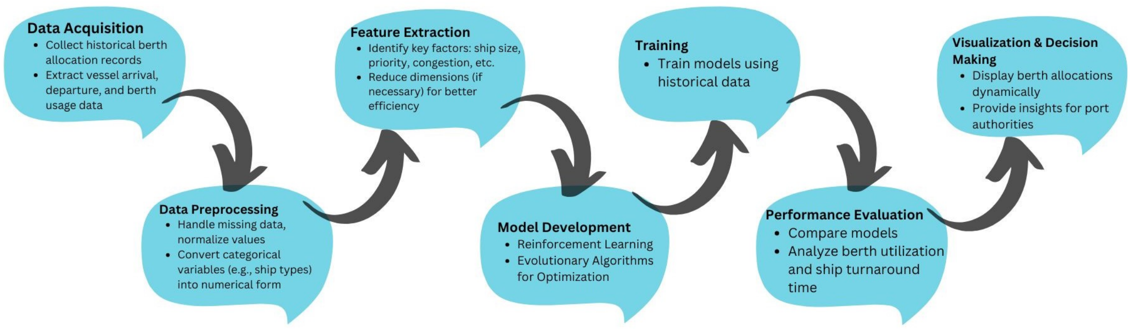

Figure 5.1: Proposed Methodology Layout

5.1 Data Preparation and Acquisition

Using historical data to construct a predictive berth allocation system requires preprocessing, analysis and organization to derive meaningful insights.

5.1.1 Dataset Selection

Our dataset is made up of historical data from Port Bin Qasim and Port of Los Angeles, including:

- Ship Arrival & Departure Data: Estimated Time of Arrival (ETA), Actual Time of Arrival (ATA), Estimated Time of Departure (ETD), Actual Time of Departure (ATD).
- Ship Attributes: Type, size, cargo type, priority level, nationality.
- Berth Information: Length, depth, crane capacity, occupancy status.

- Operational Data: Turnaround times, congestion levels, equipment availability.

5.1.2 Data Preprocessing

For ensuring data quality and fitness for AI modeling, the following preprocessing is done:

- Missing value handling and attribute normalization.
- Extracting extra features like congestion index, berth efficiency score and trends in handling capacity.
- Encoding categorical values (e.g., ship types, berth characteristics etc) for consumption by AI models.

5.2 Reinforcement Learning-Based Berth Allocation

With planned ship schedules, reinforcement learning (RL) is used to dynamically decide berth allocations. RL works by enabling an agent (here, the berth allocation system) to interact with an environment, learn from the past and optimize future action based on rewards. The model regularly updates its policy by analyzing various berth allocations, learning from past data and optimizing allocations in order to reduce vessel waiting time and increase berth utilization. By trial-and-error learning as well as policy optimization, RL becomes adaptable to dynamic port conditions such that efficient berth scheduling is ensured.

5.2.1 State Space (Model Inputs)

The AI model gets real-time input of states in the form of:

- Availability of berths (occupied or available).
- Ship characteristics, such as size, cargo type and priority.
- Current ship traffic and workload.

5.2.2 Action Space (Decisions Possible)

Possible actions for berth allocation are:

- Allocate ship to a vacant berth.
- Postpone berth allocation when congested.
- Re-allocating berths due to delays on schedules.

5.2.3 Reward Function (Decision Optimization)

The reinforcement learning model is learned from historical data and the reward function to maximize berth allocation efficiency is as follows:

- If the model predicts an historically efficient berth assignment : Positive Reward.
- If the model predicts an inefficient berth : Penalty Applied.

5.3 Hybrid AI-Heuristic Optimization

As reinforcement learning optimizes berth allocation decisions, heuristic optimization methods provide global optimization by exploring a wider solution space and refining berth assignments. The research investigates hybrid AI-heuristic models integrating reinforcement learning and evolutionary optimization methods.

To improve berth allocation efficiency, evolutionary algorithms enhance the reinforcement learning model:

- Reinforcement learning decides initial berth assignments.
- Evolutionary optimization investigates other berth assignments to increase efficiency.
- If there is a better allocation, assignments are reconfigured accordingly.
- Evolutionary optimization can also be used to tune the parameters of reinforcement model.

5.4 Comparative Analysis and Evaluation

Reinforcement learning-based berth allocation approaches are analyzed in this study and optimization methods with evolutionary algorithms are examined.

5.4.1 Comparative Analysis

The performance of reinforcement learning in berth allocation is evaluated and evolutionary algorithms for optimization are examined. The efficiency of these methods is compared using key performance indicators.

5.4.2 Evaluation Metrics

The evaluation criteria are:

- Waiting time before berthing.

- Occupancy percentage of the berth.
- Effect on total port traffic flow.

5.5 Deployment and Testing

The following is done to deploy and test the suggested berth allocation system:

- Training and testing reinforcement learning models for berth allocation.
- Incorporating evolutionary optimization for berth assignment optimization.
- Comparing results to identify the best approach.

5.6 Results and Optimization

The section discusses the performance analysis and optimization techniques for berth allocation models with efficient scheduling results.

5.6.1 Expected Performance

The expected performance for the algorithms under consideration are as follows:

- Reinforcement Learning-Based Model is suitable for dynamic situations but demands a high amount of computational power.
- Hybrid Model with Evolutionary Optimization strikes a balance between adaptability and efficiency to provide optimal berth allocations.

5.6.2 Model Refinement

Following steps can be taken to further refine our model:

- Tuning learning rate and exploration-exploitation trade-off.
- Prioritizing ship turnaround time vs. congestion management trade-offs.

5.7 Conclusion

This chapter presented an AI-based approach to berth allocation optimization, integrating reinforcement learning-based decision-making and evolutionary optimization methods. Through utilization of historical data, the suggested methodology guarantees efficient, adaptive and scalable berth scheduling.

Comparative studies evaluate the feasibility of reinforcement learning and evolutionary algorithms, with potential future enhancements to integrate real-time and explainability.

Chapter 6 High-Level and Low-Level Design

This chapter presents a comprehensive architectural design of our berth allocation system based on AI, its major components, data flow and communication among various modules. It discusses the high-level design, which defines the overall functional units of the system and the low-level design, which goes into the details of specific implementation. The system is based on a modular and scalable architecture to enable efficient berth allocation, dynamic rescheduling and real-time monitoring through an easy-to-use dashboard.

6.1 System Overview

The AI-based berth allocation system is aimed at enhancing port efficiency through berth assignment optimization through machine learning and heuristic-based techniques. The system follows a layered design and an iterative development approach, ensuring modularity, scalability and a clear separation of concern. It facilitates real-time decision-making, dynamic rescheduling and interactive monitoring for port operators. At a general level, the system comprises the following important components:

- Berth Allocation Engine which leverages reinforcement learning optimized by evolutionary algorithms to allocate berths dynamically as a function of vessel ETA, priority, cargo type and berth availability.
- Rescheduling and Conflict Resolution Module which identifies ship delays or unplanned congestion and reassigns berths in real-time through rule-based heuristics.
- User Dashboard which gives port operators a graphical view of berth allocations, ship schedules, congestion levels and manual override functionality.
- Database Management that stores an organized dataset of ship schedules, berth history, operational limitations and AI-derived berth assignments.

  The system is meant to be scalable and adaptable, enabling easy integration with future developments such as advanced prediction models, multi-port management and more optimization techniques. The subsequent sections will continue to elaborate on both the high-level and low-level structure of the system.

6.2 Design Considerations

The berth allocation system has to be developed with important considerations so that the system is efficient, scalable and applicable in reality. This section presents the assumptions, constraints, design objectives and development approaches that frame the architecture and implementation of the system.

6.2.1 Assumptions and Dependencies

The development and execution of the system are based on particular assumptions and external dependencies:

- Historical ship schedules, berth utilization records and port operational information are presumed to be organized and available. The system expects primary inputs like ETA, ETD, cargo type and ship details from port databases.
- The system relies on AI platforms (TensorFlow/PyTorch), databases (SQL/NoSQL) and web technologies for dashboard visualization.
- The training of models is presumed to have access to GPU-enabled computing for deep learning models. The system, if deployed, needs to be operating on port servers or cloud to process berth allocation requests in real-time.
- Port operators and logistics managers are the direct users and it is presumed that they will be using the system through an easy-to-use dashboard to verify and override berth allocations if required.
- The system is intended for operation in on-premises or cloud infrastructures, providing secure and efficient port operations.

6.2.2 General Constraints

To ensure efficiency and reliability, the system operates within certain constraints:

- The berth allocation system should make scheduling decisions within 5 seconds to avoid port operations' delays.
- Data encryption and role-based access control (RBAC) shall be enforced to allow secure access to berth schedules and avoid unauthorized changes.
- The system needs to be compatible with port management software in use to easily incorporate berth allocation proposals into current port operations.
- The system needs to scale efficiently with growth in the number of ships, berths and port extensions without loss of performance.
- AI computations need GPU acceleration for training and database queries need to be optimized for quick data access.

6.2.3 Goals and Guidelines

The system adheres to fundamental design principles to guarantee accuracy, efficiency and usability:

- The AI model should be of high accuracy in berth allocation in order to minimize ship waiting times and enhance port efficiency.
- The system needs to balance processing time and memory requirements, providing smooth scheduling without overloading computational resources.
- The dashboard should offer an easy and interactive interface through which port operators can easily view, override and analyze berth schedules.
- The system must be able to handle dynamic berth reassignment, making changes in the event of delays or surprise ship arrivals.

6.2.4 Development Methods

The system is designed with an Agile approach, allowing for flexibility and ongoing improvements from user feedback.

- Our project is using an iterative development approach. The project is constructed in several iterations, improving berth allocation models through real-world testing and feedback.
- Berth allocation and dashboard visualization i.e core modules are developed in parallel to bring faster implementation.
- Continuous testing and validation is performed. Scheduling algorithms and AI models are tested rigorously using historical port data to refine performance and resilience.
- The system should be able to integrate with the port management software if needed,ensuring minimal operational disruption.

Through these design considerations, the system provides robustness, flexibility and practicality in AI-based berth allocation.

6.3 System Architecture

The berth allocation system is architected as a modular, scalable and AI-based system that maximizes berth scheduling while providing human intervention through an interactive dashboard. The architecture is divided into four different layers, allowing effective management of data, AI-driven decision-making and real-time flexibility. This section describes the system's internal structure, such as its central subsystems, data flow and interfaces with external elements. The reason for this decomposition is that it ensures:

- The system is modular i.e. Each of the subsystems operates independently of others while contributing to overall system efficiency.
- It is designed to be scalable i.e the system will be able to process rising ship traffic and berth allocation requests.
- PortFlow is a machine learning and optimization-based intelligent berth scheduling system.
- A Human-Centric Interface is implemented to enable port operators to track berth usage and override AI-driven decisions.

The system is structured into four main components, with seamless interaction between AI-based berth allocation, data management and user interaction.

6.3.1 Subsystem Architecture

The system consists of four primary subsystems, each responsible for a distinct set of functionalities. These include data storage, AI-based berth allocation, business logic and user interaction.

6.3.1.1 Data Management Subsystem (Storage and Retrieval Layer)

This subsystem handles the storage and retrieval of ship, berth and schedule data. It ensures that AI models and the dashboard have access to accurate, up-to-date information.

- Database: Holds all ship schedule, berth assignment and port operational information.
- Historical Data: Employed for AI training and predictive analysis.
- API Data Retrieval: Provides real-time berth status updates for making decisions.

Key Functions:

- Stores ship, berth and schedule information.
- Dynamically fetches and updates berth status.
- Supplies AI models with necessary input data.

6.3.1.2 AI Processing and Optimization Subsystem (Berth Allocation Engine)

This subsystem is the central decision-making module tasked with intelligent berth allocation through AI and optimization.

- Applies Deep Reinforcement Learning for dynamic berth allocation.
- Hybrid Optimization with with Evolutionary Algorithms optimize berth allocations for global optimization.

Key Functions include:

- Learns optimal berth allocation policies.
- Dynamically adjusts berth assignments.

6.3.1.3 Application Logic Subsystem (Business Logic and Exception Handling)

This subsystem serves as the operating control layer, implementing business rules, exception handling and manual overrides.

- Allows a manual override system that enables port operators to override AI-determined berth assignments.
- It reassigns berths automatically in the event of unanticipated ship delays or berth overload.
- Enforces priority-based berthing for emergency cargo or priority vessels.

Key Functions:

- Ensures port-specific business rules compliance.
- Offers manual intervention capabilities.
- Manages congestion alarms and berth conflicts.

6.3.1.4 Presentation and Interaction Subsystem (Dashboard and API Layer)

This subsystem offers an intuitive interface for viewing berth assignments and making changes.

- A dashboard presents berth assignments, ship schedules and congestion alarms.
- API Services publish real-time berth occupation information for external port management system integration.
- User Authentication and Access Control restricts only the right users to make changes to berth assignments.

Key Functions:

- Enables port operators to visualize berth allocations.
- Offers alerts for scheduling conflicts.

- Supports integration with external port management systems through APIs.

6.3.2 External System Interactions

The system communicates with external systems to provide smooth operation and integration with the current port infrastructure.

- Port Management System (PMS) Synchronizes ship schedules and berth availability.
- Shipping Companies and Agents can view berth schedules through API for improved planning.

6.3.3 System Workflow (Step-by-Step Process)

The subsequent steps show the entire berth allocation process, from retrieving data to assignment of berths and user intervention:

- Ship schedules and berth availability are retrieved by the system from the database.
- Initial assignments of berths are made based on ship features and berth limitations. Optimization of the assignments is carried out by evolutionary algorithms, making global efficiency possible.
- The optimized berth schedule is enforced in real-time. Man-in-the-loop can override the decisions made by AI manually.
- In case of delay of a ship or berth congestion, berths are dynamically rebalanced by AI. Alerts inform operators of bottlenecks and conflicts.
- Operators can track berth usage through the dashboard. APIs enable data access for external port systems.

6.3.4 System Architecture Diagram

Below is the architecture diagram for PortFlow

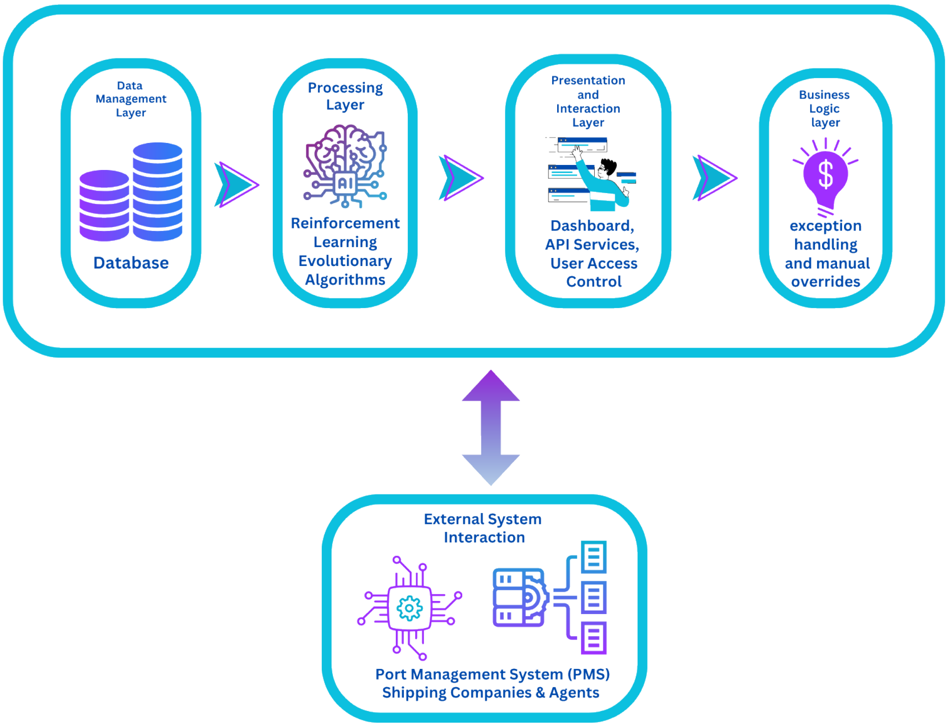

Figure 6.1: System Architecture Diagram

6.3.5 Rationale for System Decomposition

The suggested architecture is organized into four separate layers (Data Management, AI Processing, Business Logic and User Interaction) to allow for modularity, scalability and flexibility. This architecture was selected on the basis of the following major factors:

6.3.5.1 Separation of Concerns

Each subsystem only handles a distinct function—storage of data, AI-based berth allocation, business logic and user interaction, thus maintains a clean and understandable system.

6.3.5.2 Future Expansion

By decoupling the AI processing from the business logic, replacing or upgrading models is simple and straightforward.

6.3.5.3 Real-Time Overriding and Manual Monitoring

Unlike automated systems, our design includes a user dashboard for manual intervention, allowing port operators to adjust berth assignments in the event of unforeseen constraints.

6.3.5.4 Scalability and Performance Considerations

A monolithic design was rejected because it would lead to performance bottlenecks and poor adaptability for future AI model upgrades.A microservices approach was considered but was dismissed for now due to complexity. We focus on modularity within a single system instead of full microservices.

6.3.5.5 Handling Historical Data Instead of Real-Time Streaming

We did not select a purely predictive AI model that creates ETA/ETD itself since we are presuming that we will get these values from external sources. Instead, we aim for reactive decision-making and adaptive berth reallocation according to constraints in our dataset.

A few other designs were proposed but rejected due to various issues. A monolithic system was not suitable because it is not scalable, upgrading AI models in it is a problem and it has bad modularity. Although a microservices architecture provides improved modularity, it was not used since it introduces extra complexity without immediate gain, but it can be considered for scaling in the future. An end-to-end AI automated process was also ruled out since it entirely eliminates human decision-making, which is impractical in the context of actual port operations. Finally, predictive model-based calculation of ETA/ETD was not possible as it assumes availability of real-time data, which is missing in our system currently.

Thus, our hybrid AI + optimization model with modular layers ensures efficient berth allocation while allowing manual intervention when needed

6.4 Architectural Strategies

The architectural strategies for the berth allocation system aim at efficiency, scalability and flexibility in handling berth assignments dynamically. The architecture provides optimized berth scheduling, real-time monitoring and user-friendly interaction through an organized modular system. This section outlines the tools, design decisions and future plans to achieve an optimal system.

6.4.1 Tools and Software

The system is based on a number of core technologies to provide efficient data processing, optimization and visualization.

6.4.1.1 Python & PyTorch/TensorFlow

The AI models and optimization algorithms for berth allocation are developed using Python, leveraging deep learning frameworks.

- The primary programming language for data processing, training AI models and optimization is Python. These frameworks support GPU acceleration that is necessary in processing such a huge dataset.
- Python offers a rich ecosystem of libraries for the application of machine learning, data processing. The libraries are used in handling data pre-processing, extraction of characteristics and evaluation of the model.
- PyTorch/TensorFlow supports deep reinforcement learning and optimization for berth allocation.
- Major libraries are NumPy (numerical computations), Pandas (data handling) and OpenAI Gym (reinforcement learning environments).

6.4.1.2 SQL Server

A relational database stores and manages ship schedules, berth data and historical records for real-time access.

- A relational database (PostgreSQL) holds ship schedules, berth information and historical data for real-time access.
- Query optimization indexing and caching for rapid data access reduce AI decision-making latency.
- Supports ACID transactions to ensure data consistency and data loss prevention during berth re-allocation.

6.4.1.3 Django (Backend API)

Django REST Framework (DRF) is used to develop the backend API.

- Backend API development is handled using Django REST Framework (DRF) to provide secure and efficient communication between the AI engine, the database and the user dashboard.
- Real-time berth allocation information, schedule updates and congestion monitoring are provided through API endpoints.

6.4.1.4 React.js (Frontend Dashboard)

The interactive web dashboard for port operators is built using React.js.

- React.js is employed to create an interactive web dashboard for port operators.
- Port operators are able to view berth schedules, override AI-based decisions and monitor ship statuses.
- Chart.js and D3.js offer data visualizations for berth occupation and ship movements.

6.4.2 Client-Server Model

The system has a client-server architecture with divided responsibilities:

- The frontend (client) offers an easy-to-use interface for administrators and port operators.
- The backend (server) executes data processing, AI decision-making and berth scheduling optimizations.

6.4.2.1 Client-Side

The frontend provides a user-friendly interface for port operators.

- The React.js dashboard enables users to look at berth assignments, request reassignment and observe congestion trends.
- Users may override berth assignments manually when needed.

6.4.2.2 Server-Side

The backend has following functionalities:

- The backend executes real-time berth allocation requests, implements reinforcement learning with evolutionary optimization and updates berth statuses.
- Database queries facilitate effective retrieval of ship schedules and berth usage metrics.
- AI model inference dynamically allocates berths considering ship priority, congestion levels and availability.

This architecture ensures a clear separation of concerns, where backend AI and data processing are decoupled from frontend visualization, enhancing system scalability. It also incorporates asynchronous API handling, allowing for a smooth user experience even under high data loads. Additionally, the system is designed for scalability, enabling the server to be horizontally scaled to handle more requests while keeping the client lightweight.

6.4.3 System Enhancement Future Plans

In order to further enhance berth allocation efficiency and port operations, numerous enhancements are in the pipeline:

6.4.3.1 Additional Optimization Methods Integration

Berth assignment can be further enhanced by using Multi-Agent RL for improving coordination between multiple AI agents managing different berths.

6.4.3.2 Integration of Real-Time Data

Real-time port data will enable dynamic berth reassignment based on live analytics and IoT sensor inputs.

- If there is access to real-time port data, the system can be modified to dynamically reassign berths based on streaming data analytics.
- Real-time ship location and congestion information can be provided by IoT-enabled sensors to improve AI predictions.

6.4.3.3 Advanced Predictive Analytics

AI-powered predictive modeling will enhance ship arrival forecasts and detect congestion anomalies.

- Apply LSTM-based predictive modeling to predict ship arrivals and departure delays.
- Traffic heatmaps and anomaly detection can detect unusual port congestion trends.

6.4.3.4 3D Simulation for Berth Visualization

A real-time 3D simulation for berth monitoring and decision-making.

- A real-time 3D simulation will be incorporated into the dashboard, enabling port operators to dynamically visualize ship movements, berth usage and congestion levels.
- The simulation engine will create interactive, real-time models of ship docking, cargo unloading and berth reallocation scenarios.
- This virtual port environment will assist in decision-making, strategy planning and training for port operators by offering an immersive, data-driven experience.

These enhancements provide greater flexibility, enabling the system to handle unexpected port conditions efficiently. They also improve automation by minimizing human interaction while ensuring effective berth scheduling. Additionally, they enhance scalability, future-proofing the system to manage increasing port traffic effectively.

6.5 Domain Model/Class Diagram

Below is the class diagram.

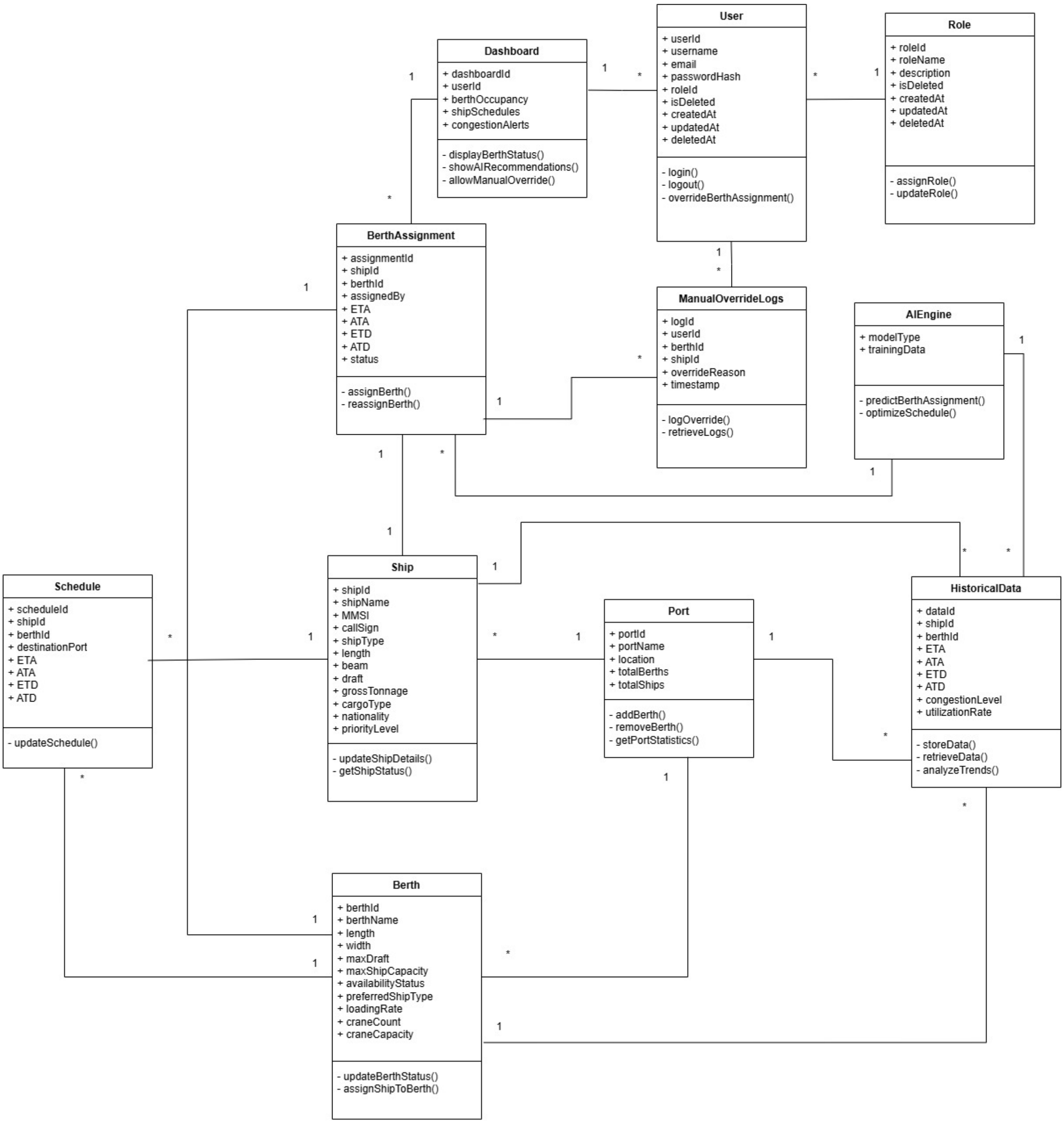

Figure 6.2: Class Diagram of PortFlow.

6.6 Policies and Tactics

This chapter details the design policies and tactical options that determine how the AI-based berth allocation system is implemented. These decisions influence the software architecture, development strategy, security and maintainability without significantly affecting the overall high-level design.

6.6.1 Selection of Technologies and Software Components

The choice of frameworks and tools depends on scalability, performance and ease of maintenance. The backend is implemented using.NET (C#) because of its stability in handling API calls and database operations, whereas the AI model is created with Python (PyTorch) for optimization of deep learning. The frontend is implemented with React.js for delivering a responsive and dynamic user interface.

For database management, PostgreSQL is selected for its ACID support, providing data integrity and speedy query execution of berth allocation information. Redis is also incorporated to cache highly retrieved ship schedule data to minimize API response time. Other options like Node.js for the backend and MongoDB for the database were entertained but discarded owing to performance bottlenecks in processing complex berth scheduling queries and transactional consistency problems.

6.6.2 Engineering Trade-Offs

The system emphasizes real-time flexibility, efficiency and scalability in berth scheduling. To this end, we investigate various optimization methods, such as reinforcement learning, hybrid AI-heuristic optimization and conventional exact methods. Although RL provides dynamic decision-making through learning from historical berth assignments, it can be challenged by convergence speed and global optimization. In contrast, conventional optimization techniques such as Mixed-Integer Linear Programming (MILP) and Genetic Algorithms (GA) may offer near-optimal berth allocations but are computationally costly and not real-time-friendly. The hybrid technique (RL + Evolutionary Optimization) balances these trade-offs by having RL model generate initial berth allocations and Evolutionary Algorithm refine the allocations to reduce waiting time and congestion. This blend balances flexibility and global optimization, making it the most effective solution for our system against isolated AI or purely heuristic approaches.

6.6.3 Coding and Implementation Standards

By adhering to modular design principles, the development makes sure that distinct elements like vessel tracking, berth assignment and real-time monitoring remain loosely coupled for simpler testing, debugging and future enhancements.To guarantee interoperability between the AI model, backend and frontend, the code is organized in accordance with RESTful API standards, using defined response formats and uniform naming conventions. To facilitate easy developer cooperation, Git (GitHub/GitLab) is used to manage version control with branching techniques (feature branches, development, main).The following rules are upheld in order to preserve code security and quality:

- Tools such as Pylint (for Python) and ESLint (for JavaScript) are used for linting and static code analysis.
- Jest (React.js), PyTest (Python) and NUnit (C#) are used for unit and integration testing.
- CI/CD pipelines for deployment and automated testing.

6.6.4 Security Policies and Data Protection

Security procedures include the following to guarantee data integrity and confidentiality:

- Role-Based Access Control (RBAC) is implemented i.e. only those with permission (admins, port operators) can change berth assignments.
- Berth scheduling data is protected by AES-256 encryption.
- OAuth 2.0 authentication secures API endpoints, ensuring only authenticated users access vital port data.
- To keep track of unauthorized alterations, audit logging records every update to the berth assignment.

SSL/TLS protocols are used to encrypt all communications between the AI model, backend and frontend in order to prevent data breaches.

6.6.5 Testing and Debugging Strategy

System robustness and dependability are guaranteed by a multi-tiered testing methodology:

- Unit Testing for database queries, AI inference models and individual API endpoints are all tested separately.
- Integration testing makes sure that the backend, dashboard user interface and AI-driven berth allocation all work together seamlessly.
- Performance testing makes sure that berth assignments scale well with growing ship traffic by assessing how the system behaves under heavy traffic loads.
- Security testing finds flaws in database access rules, authentication procedures and API endpoints.

Prometheus and Grafana are used to implement continuous monitoring, which enables real-time performance tracking of the berth allocation system and AI model.

6.6.6 Maintainability and Future Enhancements

Future enhancements are made possible by the system's scalable and extensible design:

- By simulating berth assignments in real-time using Three.js/WebGL, a visualization module will give port operators an interactive perspective on congestion trends.
- Using fresh berth allocation data, the model is continuously trained to increase its accuracy over time.
- To manage berth scheduling across several terminals at once, multi-agent reinforcement learning is being used.

The berth allocation system guarantees long-term stability, scalability and flexibility to future port management requirements by adhering to structured code, security best practices and rigorous testing techniques.

6.7 Conclusion

In Chapter 6, PortFlow system's modular architecture, which seamlessly integrates AI-driven berth allocation, real-time decision-making and user interaction was examined along with its high-level and low-level design. The system is composed of four main layers i.e. Data Management, AI Processing and Optimization, Application Logic and Presentation and Interaction. These layers are essential to ensuring seamless and effective port operations. At its core, PortFlow uses RL model in conjunction with hybrid optimization techniques to deliver adaptive and optimized berth scheduling. The user-friendly dashboard provides port operators with real-time insights and manual override capabilities, balancing automation and human control.

The system, which is built for future expansion, scalability and flexibility, may incorporate predictive analytics, real-time data streams and even 3D berth simulations. PortFlow makes berth allocation a more intelligent, sustainable and efficient process by fusing human experience with AI-driven automation, laying the groundwork for future smarter port operations.

Chapter 7 Implementation and Test Cases

In this chapter, we provide detailed implementation of our proposed system, which combines Deep Q-Learning with Bayesian Optimization to solve a navigation task within a simulated environment. Our approach focuses on optimizing agent learning efficiency by automating hyperparameter tuning through Bayesian Optimization and Genetic algorithm.

We will outline how we structured the environment, designed the reward function, implemented the DQN agent and integrated Bayesian Optimization and Genetic algorithm to enhance training performance. Furthermore, we highlight the training pipeline, including the data flow, algorithmic logic and interaction between various components.

7.1 Implementation

We developed an intelligent berth allocation system using a Dueling Double Deep Q-Network (D3QN) agent trained within a custom OpenAI Gym environment. The goal was to dynamically assign berths to incoming ships in a way that minimizes port congestion and maximizes berth utilization.

To optimize the agent's performance, we employed Bayesian hyperparameter tuning using the Optuna framework and the genetic algorithm, with the possibility of replacing them with other algorithms like Particle swarm optimization and Cuckoo search algorithm for comparative analysis.

This section outlines the major implementation components, which are further elaborated in the upcoming subsections

The overall system architecture for the berth allocation optimization problem was structured as shown in the diagram below. The process began with data preprocessing, where raw berth occupancy and ship arrival data were cleaned and transformed into a format compatible with the custom Gym environment. This environment simulated real-time berth assignment using domain-specific parameters such as commodity codes, berth availability and current occupancy. Within the environment, the reward function computed a score based on berth congestion and utilization, guiding the agent's learning process. A Dueling Double Deep Q-Network (D3QN) agent interacted with the environment and learned optimal policies over time through trial and error. To further enhance the agent's performance, Bayesian optimization using Optuna and Genetic algorithm were employed to tune key hyper parameters such as learning rate and discount factor. The optimized agent was then trained across multiple episodes, with the goal of maximizing cumulative rewards, reducing congestion and improving berth utilization. Each component in the pipeline was modular and adaptable, allowing for future extensions for other optimizations and heuristics.

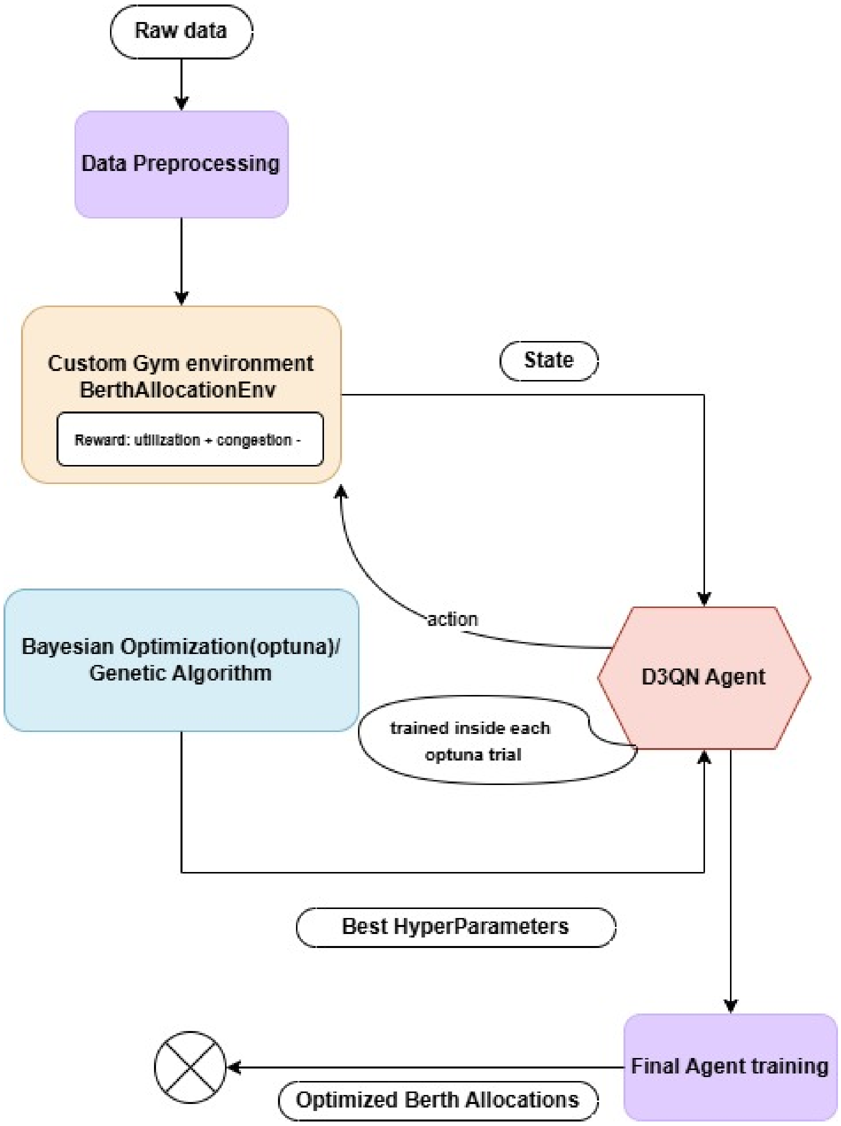

Figure 7.1: Overall view of implementation.

7.1.1 Dataset preprocessing and preparation

The initial raw dataset contained ship-level operational data, including details such as ship name, nationality, GRT, LOA, dead weight and identifiers like IMO and MMSI numbers. It also included timestamped records for anchorage, berthing and sailing events, each recorded in separate date and time fields. Additional columns captured the berth and terminal used, the associated shipping agent, last port of call and commodity details including import and export volumes.

To make the dataset suitable for modeling berth allocation, several preprocessing steps were required. The first step was the integration of berth-specific information. This involved joining the raw vessel dataset with external data containing detailed attributes of each berth, such as annual capacity (e.g., 4 million tons), vessel size and draught limits (e.g., 210m/12m). Another early task was understanding and incorporating terminal-specific context. Each terminal in the dataset had a distinct structure. some had only one berth, while others had two or three. Also each terminal was designed to handle specific commodities, such as coal for power plants or containerized goods. Based on this, we created mappings that linked individual berths to their respective terminals and separately mapped commodities to the terminals that were capable of handling them. These mappings were essential for structuring the environment and ensuring that our agent only made valid allocations.

After these mappings were established, we derived several useful features from the existing data. The separate date and time fields for anchorage, berthing and sailing were merged into complete datetime fields. This allowed us to calculate key operational metrics, including waiting time (berthing time minus anchorage time), stay duration (sailing minus berthing) and total port time (sailing minus anchorage). We also extracted hour and day components from anchorage and berthing times to capture temporal patterns that could affect port operations.

We then encoded various categorical fields to prepare the dataset for use in a reinforcement learning environment. Categorical columns such as ship name, shipping agent, nationality, last port, commodity, terminal and ship type were encoded using label encoding, resulting in numerical representations like Ship NameCode, TerminalCode and CommodityCode. These encodings ensured consistency and enabled the agent to process the data numerically.

In the final cleaning phase, we addressed missing values in fields such as Dead Weight, LOA, IMO NO and MMSI NO using standard imputation techniques like mean or mode replacement. Where critical values were missing and could not be reasonably inferred, rows were dropped to preserve dataset quality. Additional formatting steps included standardizing date formats and ensuring categorical fields were uniformly represented.

The result was a fully preprocessed dataset with over 30+ features, combining original ship operation data, derived time-based attributes, encoded categorical fields and structured terminal-berth-commodity mappings.Lastly we visualized the trends in the data to understand it better. This refined dataset was used within our custom-built environment, forming the core input for our AI-based agent.

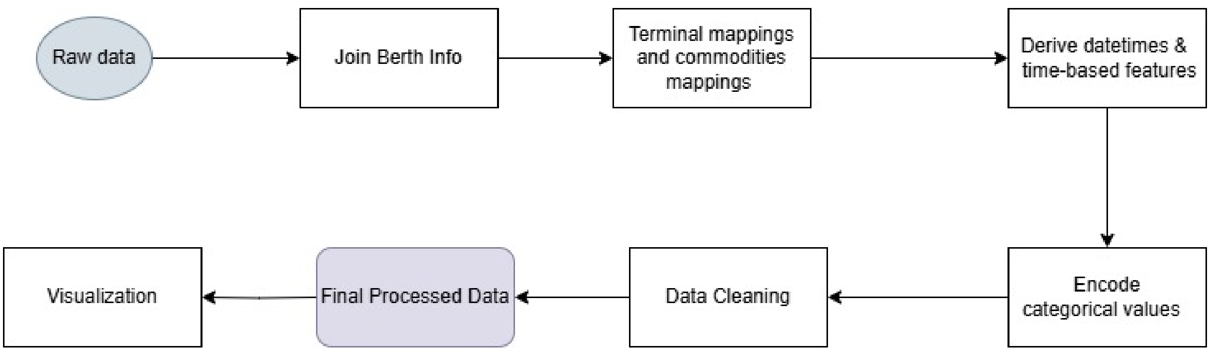

Figure 7.2: Dataset Preprocessing and preperation.

7.1.2 Environment Setup

To model the berth allocation problem realistically and enable reinforcement learning-based optimization, a custom OpenAI Gym environment named BerthAllocationEnv was implemented. The environment simulated the sequential arrival and assignment of ships to terminal berths while tracking terminal occupancy and environmental metrics such as congestion and berth utilization.

The environment received a dataset containing ship-specific features (e.g., commodity type, stay duration) and was initialized with a maximum queue length of 50 ships per commodity type. It supported 12 different commodity types and 13 terminals, with each terminal having a configurable number of berths. Each berth was modeled with key attributes such as its terminal ID, occupancy status, occupying ship details, time until which it remained occupied, total occupied time, idle step counter, and consecutive occupied step counter.

A nested data structure (terminaltoberths) was used to map each terminal to its corresponding berth IDs. Berths were initialized during environment setup, and a mapping from commodities to terminals (commoditiestoterminals) was used to track compatibility.

The observation space was a continuous vector of size:

$$2 \times \text{number of berths} + \text{number of commodities}$$

Each berth contributed two values (occupancy status and normalized occupation time), while each commodity contributed a normalized queue length value. This vector was used to capture the current system state at each decision step.

The action space was defined as a discrete set of berth indices, plus one extra action (no-op), giving an action space size of numberths + 1. During runtime, actions were interpreted based on berth compatibility and availability.

At each time step, the environment progressed by either assigning a ship to a berth or skipping allocation based on the agent's action. If a berth was unoccupied and compatible with the ship's commodity, the ship was dequeued and assigned to the berth. The berth's occupancy attributes were updated, and a positive reward was given.

After every valid assignment, the environment computed two critical metrics. These were defined numerous times and are subject to more improvement as they as the crux of our problem. The Congestion was calculated as:

$$\text{Congestion} = \min\left(\frac{\text{Total Queued Ships}}{\text{Max Queue Length}}, 1.0\right)$$

Where:

- **Total Queued Ships** is the total number of ships waiting in all commodity-specific queues.
- **Max Queue Length** is a normalization constant (default = 50) defining the maximum queue capacity.

This metric is normalized to the range $[0,1]$ and reflects port congestion based on queue occupancy. A congestion value of 0 indicates no ships are waiting, while a value of 1 implies queues are at or over capacity.

The second metric was berth utilization which was calculated for each commodity c as:

$$\text{Utilization}_c = \min\left(\frac{\text{Total Occupied Time}_c}{\text{Total Available Berth Time}_c}, 1.0\right)$$

Average berth utilization across all active commodities:

$$\text{Average Utilization} = \frac{\sum_{\substack{c\in\text{Active}\\ \text{Commodities}}} \text{Utilization}_c \cdot \text{Berth Count}_c}{\text{Total Active Berths}}$$

Where:

- **Total Occupied Time**$_c$: Cumulative time (in hours) that berths compatible with commodity c are occupied.
- **Total Available Berth Time**$_c$: Total time available for berths compatible with commodity c, computed as the number of compatible berths multiplied by the simulation time.
- **Berth Count**$_c$: Number of berths that support commodity c.
- **Active Commodities**: Commodities with non-empty queues at the time of computation.

Utilization is capped at 1.0 to prevent overweighting. The weighted average accounts for how many berths each commodity can use, ensuring a balanced assessment of berth usage efficiency.

The final reward signal was a weighted average of utilization and congestion (inverted):

$$\text{Reward} = \alpha \cdot \text{Utilization} + \beta \cdot (1 - \text{Congestion})$$

where α and β are scaling factors that determine the relative importance of berth utilization and congestion minimization, respectively. These factors can be adjusted based on the optimization objectives, and are often chosen such that $\alpha + \beta = 1$.

The environment introduced some action-specific bonuses and penalties:

- $+1.0$ for assigning a ship to a compatible berth.
- $+0.1\times$ consecutive occupied steps to encourage sustained use.
- $+0.5$ for assigning a ship to an idle berth after 3 idle steps.
- -0.02 for failing to assign a ship to an idle berth.
- $+0.15$ (or $+0.075$ normalized) if congestion < 0.2.

These are normalized and combined as follows:

$$\text{Normalized Base} = \text{Base Reward} \cdot 0.5$$

$$\text{Normalized Action} = \left(\frac{\text{Action Reward} - \text{Action Min}}{\text{Action Max} - \text{Action Min} + 10^{-6}} \right) \cdot 0.5$$

$$\text{Total Reward} = \text{clip}(\text{Normalized Base} + \text{Normalized Action},\ 0,\ 1)$$

The episode advanced ship-by-ship. At each time step, new ships were probabilistically added to queues (50 percent chance), and berth status was updated based on completion times. The environment terminated once the simulation time exceeded 50 steps or all ships from the dataset had been processed

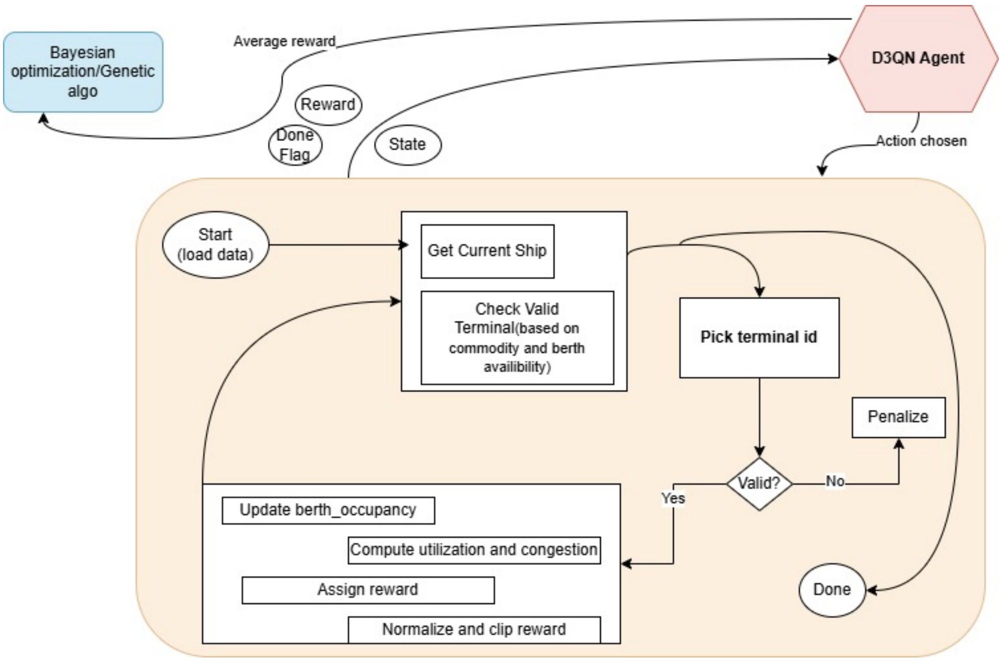

Figure 7.3: BerthAllocation Environment.

7.1.3 Hyperparameter Optimization

While aiming to enhance the berth allocation, we tuned our hyperpaarmeters of D3QN using various optimization algorithms, the detail is given as follows://

7.1.3.1 Bayesian Optimization

To improve the performance of the berth allocation agent, this project used Optuna, a hyperparameter optimization framework based on Bayesian optimization. The goal was to find the best values for four key hyperparameters of the Dueling Double Deep Q-Network (D3QN) agent: learning rate (lr), discount factor (gamma), epsilon decay (epsilondecay) and minimum epsilon (epsilonmin). Optuna ran multiple trials and in each one, it sampled a unique set of these parameters and passed them to a newly initialized D3QN agent.

Each trial involved training the agent within the berth allocation environment for a number of episode. During training, the agent received a state from the environment, selected a valid terminal berth as an action and received a reward based on how well it balanced berth utilization and congestion. These values were recorded but not used directly for optimization. Only the average reward across the 20 episodes was returned to Optuna, which used it to evaluate the performance of that hyperparameter set.

This module functioned as a loop: Optuna proposed a configuration, the agent trained using that configuration and the average reward was returned to guide future trials. Once all trials were completed, Optuna provided the best-performing hyperparameters, which were then used to train the final version of the agent.

In terms of data flow, Optuna sent the sampled hyperparameters into the environment-agent setup. The environment responded by simulating berth allocations and returning rewards to the agent, which learned from its experience. After training, the environment indirectly contributed to Optuna's decision-making by influencing the final reward value that Optuna used to judge trial performance.

7.1.3.2 Genetic Algorithm Optimization

In order to enhance the performance of the berth allocation agent, the project also tested Genetic Algorithm (GA), which is a population-based optimization algorithm based on natural selection. It aimed to identify the optimal values for four essential hyperparameters of the Dueling Double Deep Q-Network (D3QN) agent: learning rate (lr), discount factor (gamma), epsilon decay and minimum epsilon. GA has a set of candidate solutions (sets of hyperparameters), iteratively evolves them across generations and chooses the top-performing configurations.

Every member of the population is a distinct combination of the four hyperparameters. For every generation, the fitness of every member is assessed by training a D3QN agent in the berth allocation setting with those hyperparameters. The agent acts on the environment by observing a state, choosing a valid terminal berth as an action and observing a reward depending on how well it balances congestion and berth utilization. The average reward over multiple episodes was taken as the individual's fitness score.

Parents were chosen as the best-performing half of the population. New individuals (children) were generated by the crossover of hyperparameters between two randomly selected parents. Random mutations were applied to some of the hyperparameters at a fixed mutation rate to preserve diversity and prevent local optima. This continued for several generations.

After all the generations were done, the one with the best average reward was chosen as the optimal hyperparameter setting. These best hyperparameters were then employed to train the final agent.

With regard to data exchange, GA created a population of hyperparameter sets, each of which was fed into the environment-agent configuration. The agent was trained on the given parameters and the environment provided rewards to inform selection of the best individuals. Crossover and mutation allowed the GA to adjust the hyperparameters over time towards better agent performance.

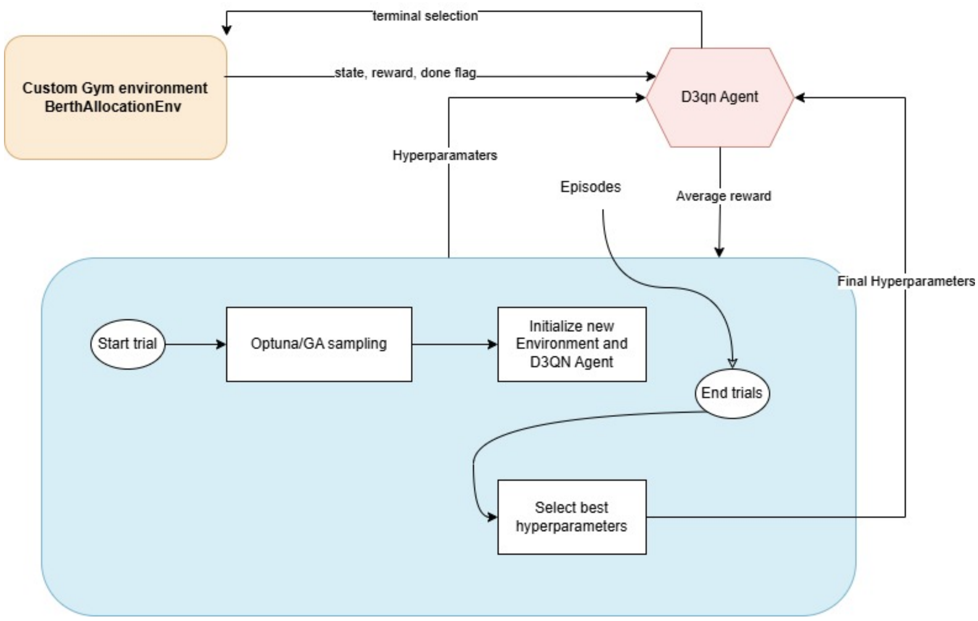

Figure 7.4: Bayesian optimization optuna or Genetic Algorithm

7.1.4 DQN and Training Process

The Dueling Double Deep Q-Network (D3QN) agent was implemented using PyTorch to solve the berth allocation optimization problem. The primary goal of this agent was to assign ships to available berths efficiently, aiming to maximize port utilization while minimizing congestion. The architecture of D3QN built on the standard Deep Q-Network (DQN) by separating the estimation of state values and advantages. This allowed for more stable learning by computing Q-values as the sum of a value stream and an advantage stream, as described by the equation:

$$Q(s,a) = V(s) + A(s,a) - \frac{1}{|\mathscr{A}|}\sum_{a'} A(s,a')$$

The agent's behavior was governed by a training loop that followed the standard reinforcement learning cycle: the environment was reset at the start of each episode and at each step, the agent selected an action based on an ε-greedy policy. This policy balanced exploration and exploitation, where the agent initially explored more and gradually shifted to exploitation as it learned.

The agent's interaction with the environment resulted in the storage of transitions in a replay buffer, which were later sampled during training. The training process involved performing gradient descent on the loss between the predicted Q-values and the target Q-values, which were computed using the Double DQN algorithm. Specifically, the target network was updated periodically to maintain stability. The reward signal for the agent was designed to encourage high berth utilization and low congestion. The reward was structured to incentivize efficient use of berths while avoiding congestion caused by multiple ships competing for the same berth.

The exploration strategy was based on an ε-greedy policy, where the epsilon value decayed over time, starting with a high exploration rate that gradually shifted towards exploitation. The decay was governed by the formula:

$$\text{epsilon} = \max(\text{epsilon}_{\text{min}}, \text{epsilon} \times \text{epsilon}_{\text{decay}})$$

ensuring that the agent explored sufficiently early on and exploited its knowledge in later episodes. The loss function used for training was the Mean Squared Error (MSE) between the predicted Q-values and the target Q-values, with the optimizer being Adam. Hyperparameters, including the learning rate, were optimized using Optuna for improved performance.

To avoid overfitting, the agent employed an early stopping mechanism. If the agent failed to improve its performance over a set number of episodes, training was halted and the best-performing model was restored. This ensured that the agent did not continue training unnecessarily, saving computational resources. The performance of the agent was tracked through various metrics, including total reward per episode, berth utilization and congestion levels. These metrics were plotted and saved for visualization, providing insights into the agent's learning process and its effectiveness in solving the berth allocation problem. After training, the best model was saved for future use or testing.

This methodology and the resulting agent offered an optimized solution for the berth allocation problem, demonstrating how reinforcement learning, specifically the D3QN architecture, could be effectively applied to complex real-world problems like port management.

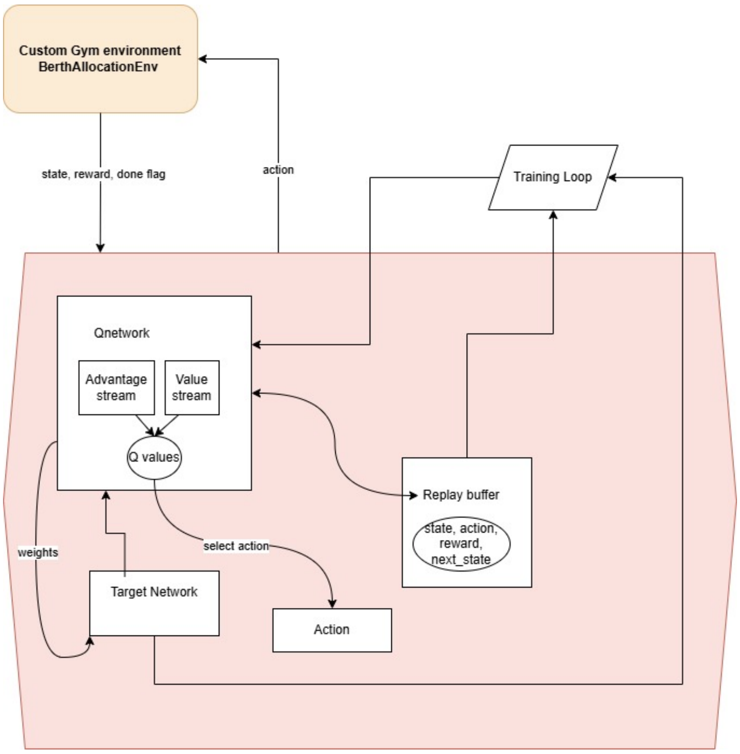

Figure 7.5: D3qn agent.

7.2 Conclusion

In this chapter, we presented the full implementation of our intelligent berth allocation system, which leveraged a Dueling Double Deep Q-Network (D3QN) agent and Bayesian hyperparameter optimization via Optuna and GA. Starting from data preprocessing, we transformed raw port operation records into a structured and enriched dataset, enabling realistic simulations within a custom OpenAI Gym environment. This environment modeled real-world constraints, such as terminal-specific berth limitations and commodity handling requirements.

Through careful design of the reward function, the agent was trained to optimize berth utilization while minimizing port congestion. The integration of Bayesian optimization and GA significantly improved training efficiency by automating the tuning of key hyperparameters. Additionally, the modular and scalable nature of our system architecture enables future experimentation with alternative algorithms such as Genetic Algorithms, Particle Swarm Optimization, or Cuckoo Search.

This implementation lays the groundwork for a robust, data-driven approach to berth allocation, demonstrating the effectiveness of reinforcement learning in solving complex, constraint-bound optimization problems in port operations.

Chapter 8 Experimental Results and Discussion

This chapter presents and discusses the experimental results obtained during the development of the PortFlow system, a berth allocation optimizer powered by a Dueling Double Deep Q-Network (D3QN) with Bayesian Optimization. The aim was to minimize port congestion, reduce vessel waiting time and enhance berth utilization. The experiments conducted focus primarily on hyperparameter optimization and post-training performance evaluation. The agent was trained across multiple episodes, with each episode representing a sequential series of ship arrivals and berth assignments. The reward function used during training incorporated two core metrics, berth utilization and port congestion, to guide the agent's decision-making process.

8.1 Hyperparameter Optimization Results

The hyperparameters are tunes using BO and GA, the experiments and results for both are as discussed in this section.

8.1.1 Bayesian Optimization

Bayesian Optimization was employed using Optuna to fine-tune the hyperparameters of the D3QN agent. Bayesian optimization via Optuna's TPE sampler explored 10 trials over the following search space.

Table 8.1: Hyperparameter Search Space and Optimal Values BO

Hyperparameter	Search Range	Optimal Value
Learning Rate	$[1\times10^{-3},\ 7.4\times10^{-3}]$ (log scale)	0.006131 ± 0.0001
Gamma (γ)	$[0.906,\ 0.995]$	0.993 ± 0.01
Epsilon Decay	$[0.954,\ 0.997]$	0.969 ± 0.01
Epsilon Min	$[0.02,\ 0.09]$	0.07 ± 0.01

The objective function maximized the average episodic reward, indirectly optimizing for berth utilization and minimizing congestion.The following tables summarize the results.

Table 8.2: Summary of Bayesian Optimization Trials

Trial	Avg Reward	Learning Rate (lr)	Gamma	Epsilon Decay
0	28.63	0.005969	0.984	0.969
1	28.12	0.003046	0.906	0.974
2	26.21	0.003989	0.983	0.997
3	27.05	0.001489	0.986	0.983
4	28.40	0.007234	0.958	0.973
5	27.97	0.005885	0.995	0.978
6	27.46	0.004597	0.940	0.954
7	27.02	0.006131	0.993	0.969
8	23.64	0.002641	0.981	0.996
9	26.91	0.007351	0.983	0.969

Table 8.3: Best Hyperparameters Selected

Hyperparameter	Value
Learning Rate	0.006131151761593549
Gamma	0.993
Epsilon Decay	0.969
Epsilon Min	0.07

These values provided the best balance between learning speed (via learning rate), long-term value estimation (gamma) and exploration-exploitation tradeoff (epsilon decay). Although some trials showed slightly better congestion or utilization, they lacked in reward consistency or robustness.

This result confirms that the D3QN agent benefits significantly from careful hyperparameter tuning and that reward-maximizing strategies can coexist with acceptable levels of congestion and berth usage in realistic port scenarios.

8.1.2 Genetic Algorithm Optimization

Genetic Algorithm (GA) was employed as an alternative evolutionary method to optimize the hyperparameters of the D3QN agent. The GA iteratively evolved populations of candidate solutions across 10 generations, evaluating their fitness based on average episodic reward.

The following table describes the hyperparameter search domain and the best configuration discovered by the GA:

Table 8.4: Hyperparameter Search Space and Optimal Values via GA

Hyperparameter	Search Range	Optimal Value
Learning Rate	$[2\times10^{-4},\ 7.7\times10^{-3}]$ (log scale)	0.007632 ± 0.0001
Gamma (γ)	$[0.933,\ 0.983]$	0.959 ± 0.01
Epsilon Decay	$[0.935,\ 0.998]$	0.991 ± 0.01
Epsilon Min	$[0.02,\ 0.08]$	0.03 ± 0.01

The objective function in GA optimization also maximized average episodic reward, with generations competing based on performance. the following table presents the results across 10 generations:

Table 8.5: Summary of Genetic Algorithm Generations

Trial	Best Score	Learning Rate (lr)	Gamma	Epsilon Decay
1	25.05	0.007739	0.980	0.988
2	27.13	0.007632	0.959	0.991
3	25.71	0.001427	0.983	0.998
4	26.47	0.000205	0.947	0.991
5	27.13	0.005093	0.936	0.951
6	21.34	0.007468	0.933	1.007
7	23.60	0.002170	0.951	0.935
8	21.10	0.000205	0.947	0.967
9	25.86	0.000205	0.947	0.991

Table 8.6: Best Hyperparameters Selected via GA

Hyperparameter	Value
Learning Rate	0.007632137479434160**5**
Gamma	0.959
Epsilon Decay	0.991
Epsilon Min	0.03

These results reinforce the effectiveness of evolutionary algorithms such as GA in hyperparameter tuning, particularly when exploring wide and potentially nonlinear search spaces where gradient-based methods may underperform.

However, for better convergence and performance stability, more generations or training episodes need to be conducted. The current outcomes represent a summary from our limited setup, and the results, while insightful, do not yet reflect optimal convergence.

8.2 Performance Evaluation

We plotted the graphs for utilization, congestion and rewards over the episodes for both the GA and Bayesian version.

8.2.1 Bayesian Optimization Version

Our training from the Dueling Double Deep Q-Network (D3QN) with hyperparameters optimized through Bayesian Optimization demonstrates a stable and informative learning path for the performance metrices i.e. congestion, reward and utilization

8.2.1.1 Reward Progression

The training rewards plot highlights a strong initial surge, showing the agent quickly picks up strategies to boost rewards early on. After this rise, the rewards show some natural variability, with the moving average reflecting a dynamic learning process as the agent adapts across episodes. This suggests the agent is actively engaging with the environment and learning over time.

Figure 8.1: Reward progression plot for BO.

8.2.1.2 Berth Utilization

The plot for berth utilization displays a promising upward trend in the initial episodes, indicating that the agent effectively begins to optimize resource use early on. Throughout the training, the utilization fluctuates, with the moving average showing a generally stable range around 0.75 to 0.85, reflecting consistent engagement with the task. This variability suggests the agent is actively adapting its strategy across episodes, laying a solid foundation for further refinement in resource allocation.

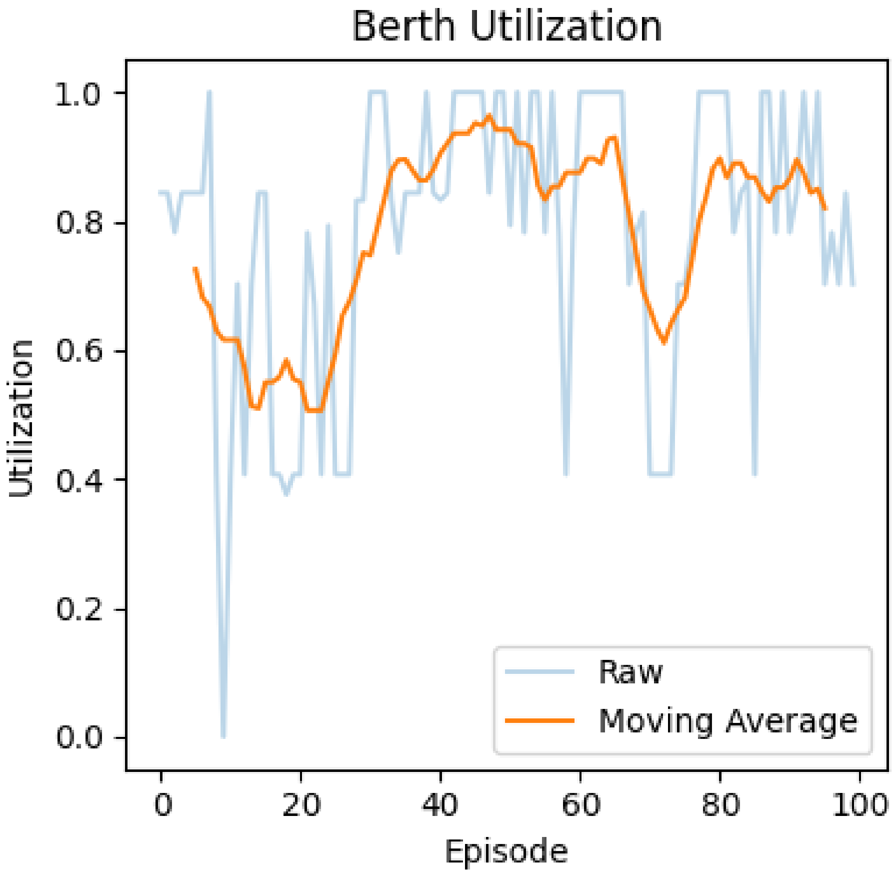

Figure 8.2: Berth utilization plot for BO.

8.2.1.3 Port Congestion

The plot for queue congestion illustrates a moderately stable yet fluctuating trend across episodes. While the raw values exhibit significant variance, the moving average hovers consistently between 0.38 and 0.45 for the majority of the training, indicating that the agent maintains a reasonable level of congestion control. The lack of a strong downward trend suggests that while the policy is stabilizing, it has not yet fully optimized queue reduction. This behavior implies that the agent is in the process of refining its decision-making, and extended training or enhanced reward shaping may be necessary to drive congestion even lower

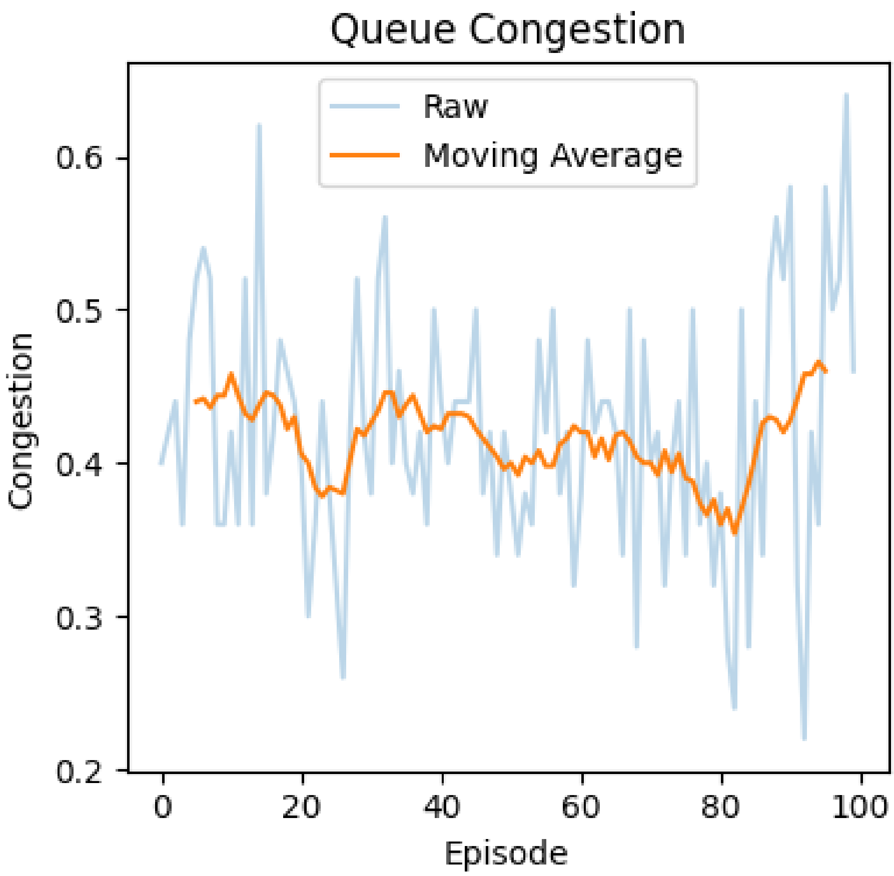

Figure 8.3: Port congestion plot for BO.

8.2.2 GA Optimization Version

The result interpretations for the D3QN model, optimized by GA algorithm are as follows:

8.2.2.1 Reward Progression

The Episode vs Reward graph shows that the agent begins with relatively high but fluctuating rewards, suggesting it is initially exploring the environment with some effectiveness. As training progresses, the fluctuations remain present in the raw reward line, but the moving average shows a clear upward trend and gradually stabilizes. This indicates that the agent is learning a more consistent berth allocation policy. While there are occasional performance drops, the overall trajectory reflects steady improvement and convergence toward a stable, high-reward policy.

Figure 8.4: Reward progression plot for GA.

8.2.2.2 Berth Utilization

The Berth Utilization plot shows that the agent maintains consistently high utilization throughout training, with the raw values frequently touching the upper limit of 1.0. The moving average begins slightly lower but steadily rises and stabilizes above 0.9, indicating that the agent learns to allocate berths efficiently. Although a few sharp drops are observed—likely due to exploration or suboptimal decisions—the overall trend reflects strong and sustained utilization performance.

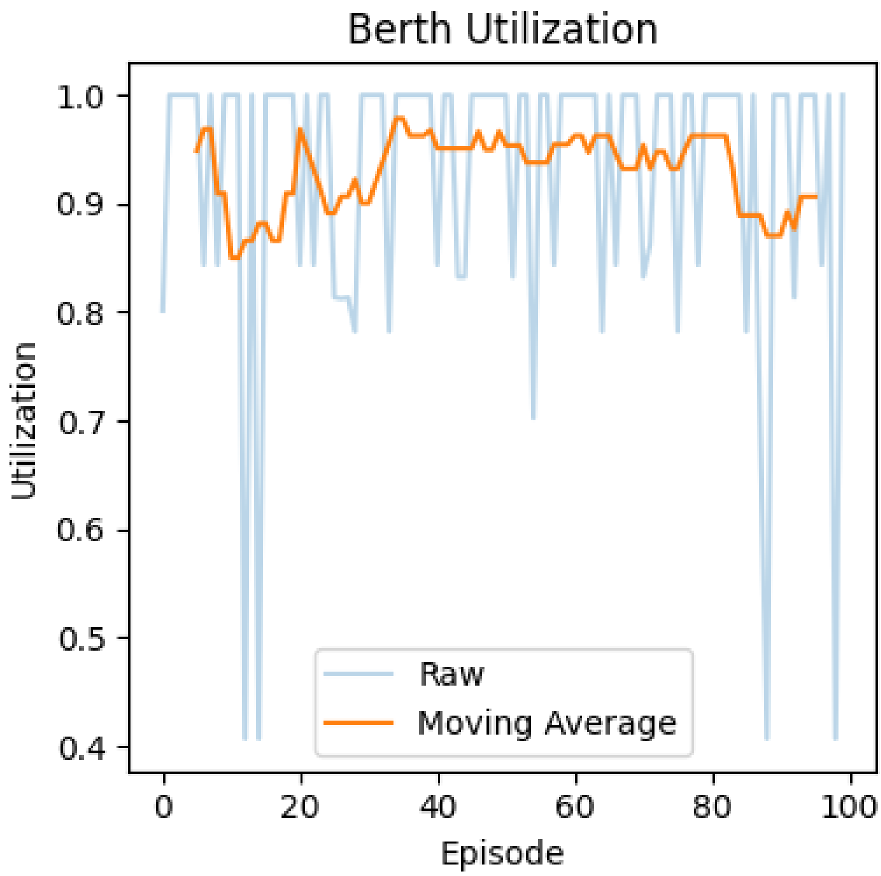

Figure 8.5: Berth utilization plot for GA.

8.2.2.3 Port Congestion

In the Queue Congestion graph, the raw values fluctuate significantly, especially in early episodes. However, the moving average shows a gradual decline, indicating that the agent is learning to reduce congestion over time. While some variability persists, the trend suggests improved scheduling behavior, leading to fewer vessels waiting in queue as training progresses.

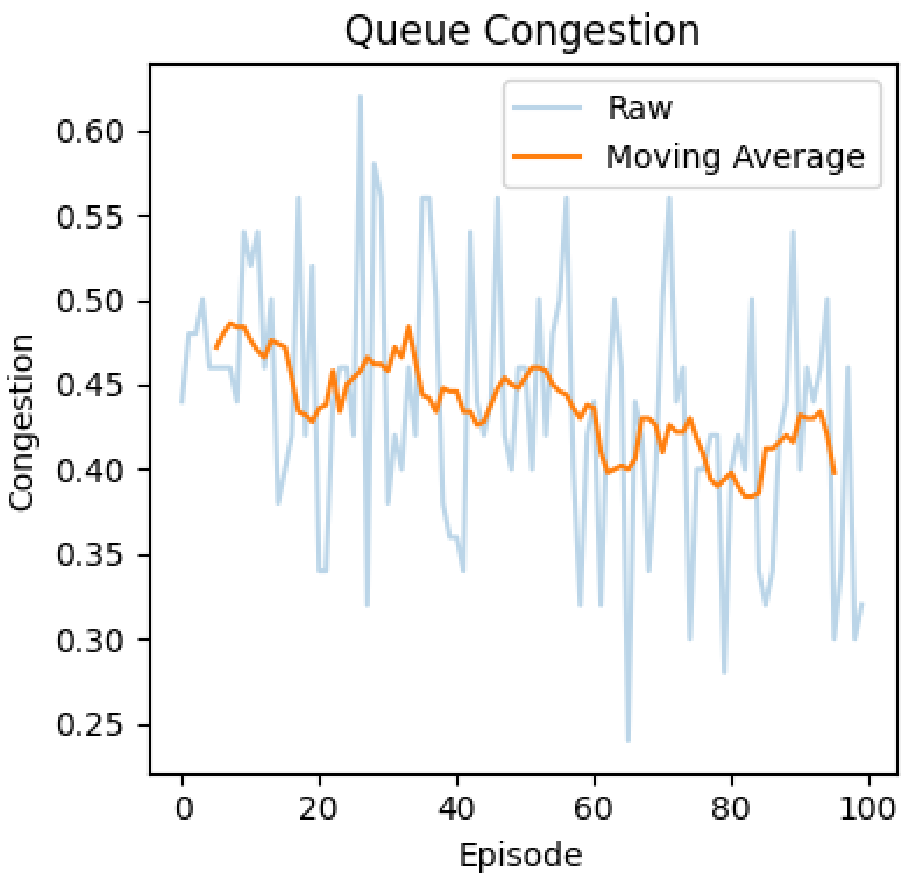

Figure 8.6: Port congestion plot for GA.

8.3 Conclusion

In conclusion, the PortFlow system, driven by a Dueling Double Deep Q-Network (D3QN) with hyperparameter tuning via Bayesian Optimization and Genetic Algorithms (GA), demonstrates strong potential in optimizing berth allocation by minimizing congestion and maximizing utilization. While Bayesian Optimization is generally known for faster convergence, our experimental setup showed better performance with GA, likely due to limited training episodes and ongoing refinements in metric definitions. These results highlight the importance of further fine-tuning and extended training to fully leverage the strengths of different optimization methods. Overall, the system shows promise for real-world deployment and merits deeper exploration and scalability testing in dynamic port environments.

Chapter 9 Conclusion and Future Work

In this chapter, we shall conclude the work done in FYP-1 of the project in detail and describe the roadmap for FYP-2. The final chapter summarizes current progress, methodologies used and challenges faced so far. In the future work section, we elaborate upon these and a few more milestones that need to be accomplished, providing a roadmap for implementing and optimizing the remaining work. This includes improvements such as dataset augmentation, dashboard integration and the integration of a simulated environment to iterate and validate the hybrid AI-based berth allocation system.

9.1 Conclusion

In modern port operations, berth allocation strategies reflect the practice of traditional methods, but there are key limitations that must be overcome, which is why the PortFlow system was proposed. The project focuses on dynamic scheduling under uncertain vessel arrival, congestion and constraints on resources. Dueling Double Deep Q-Networks (D3QN) form the backbone of reinforcement learning in this system, which adopts hybrid optimization based techniques (like Genetic Algorithm (GA), Bayesian Optimization (BO), Particle Swarm Optimization (PSO) and Cuckoo Search Algorithm (CSA)) for adaptive and scalable optimization for improving berth utilization and reducing vessel waiting periods.

In FYP-1, the project centered on literature review, establishing the architecture and methodology, mathematically defining the problem and designing a reward function that encourages berth usage while penalizing congestion and delay. A comparative strategy was developed between different combinations of AI heuristics. Even with successful theoretical foundation and extensive system planning, various implementation aspects are still in progress. The interfacing of AI modules with the backend system, simulation-based testing and dashboard integration have been delayed to FYP-2 because of resource and time limitations.

Substantial progress was achieved in conceptual development, design of model architecture and formulation of algorithms. Formulation of the BerthAllocationEnv was one of the most difficult and draining aspects of this phase. The field of real-time berth allocation is complex by nature, with dynamic scheduling, constrained spatial-temporal resources and competing optimization objectives. Converting that into a workable and realistic setting that captures operational conditions without compromising compatibility with deep reinforcement learning frameworks was a huge task. It involved hours of debugging, redefining transitions, verifying metrics such as congestion and utilization and making the behavior of the environment correspond to port operations. The burden of this challenge was substantial. Despite this, conquering these challenges provided a firm foundation for subsequent implementation and fine-tuning, merging heuristic optimization with deep reinforcement learning to solve an actual-world logistics challenge.

9.2 Future Work

To fully realize the potential of the PortFlow system, several key areas will be addressed in FYP-2. These include dataset augmentation, full backend development, system evaluation under real-time constraints and improved visualization and interactivity for end users.

9.2.1 Dataset Expansion and Augmentation

The current dataset, while functional for initial modeling, is limited in scale and diversity. The next phase will focus on Synthesizing new training samples through data augmentation, generating diverse scenarios using simulation techniques and acquiring larger and more representative datasets, potentially from more port authorities.

9.2.2 Model Refinement and Comparisons

In FYP-2, we will continue refining our current approach by improving the implementation of the hybrid models and expanding the evaluation process and compare them. The plan includes enhancing the dataset, testing in a more realistic environment and conducting broader comparisons to better understand each model's performance. This phase will also focus on improving system integration and making the overall solution more robust and deployment-ready.

9.2.3 Backend Integration and System Development

The backend infrastructure, including APIs and database connectivity, will be developed to support data ingestion and storage, enable dynamic communication between the AI engine and dashboard and to ensure secure, scalable and modular service deployment.

9.2.4 Simulated Environment for Model Testing

To emulate a realistic port scenario, a simulated environment will be developed. This simulation will allow for: Stress testing of algorithms under varying vessel traffic, observing real-time rescheduling and berth adaptation and also judging the effectiveness of different optimization methods in dynamic conditions.

9.2.5 Interactive Dashboard Completion

A central aspect of PortFlow is the operator dashboard. In FYP-2, this component will be fully realized with features including: Live berth allocation visualization, AI-recommended berth suggestions and override mechanisms and historical analytics for manual control and insight. Also a 3d visualization for vessal tracking can be considered if there is sufficient time.

9.2.6 Model Optimization and Refinement

Performance enhancements will be pursued through Hyperparameter tuning of reinforcement learning agents, Exploring attention mechanisms or actor-critic models as potential upgrades and Incorporating feedback loops for adaptive learning over time.

9.2.7 Scalability and Cloud Deployment

The final version of PortFlow if time constraints allow it, will potentially be designed for deployment on cloud platforms to allow for distributed data processing and real-time analytics and ensure robustness, security and scalability for use in different port environments.

The work accomplished in FYP-1 demonstrates the strong potential of hybrid AI-based solutions for solving complex real-world logistical problems. PortFlow represents a novel approach that combines intelligent decision-making with practical usability for port operators. The project has already established a firm research and design foundation and the planned future enhancements will significantly elevate the performance and applicability of the system.

By incorporating simulation, deeper model comparisons, interactive interfaces and larger datasets, the next phase aims to transform the PortFlow prototype into a fully functional, deployable solution capable of revolutionizing port berth allocation.

Bibliography

[1] T. Ai, L. Huang, R. Song, H. Huang, F. Jiao, and W. Ma, "An improved deep reinforcement learning approach: A case study for optimisation of berth and yard scheduling for bulk cargo terminal," *Advances in Production Engineering Management*, vol. 18, pp. 303–316, September 2023.

[2] Y. Dai, Z. Li, and B. Wang, "Optimizing berth allocation in maritime transportation with quay crane setup times using reinforcement learning," *Journal of Marine Science and Engineering*, vol. 11, p. 1025, May 2023.

[3] P. Wang, J. Li, and X. Cao, "Discrete dynamic berth allocation optimization in container terminal based on deep q-network," *Mathematics*, vol. 12, p. 3742, November 2024.

[4] B. Li, C. Yang, and Z. Yang, "Multiple container terminal berth allocation and joint operation based on dueling double deep q-network," *Journal of Marine Science and Engineering*, vol. 11, p. 2240, November 2023.

[5] M. A. Dulebenets, M. Kavoosi, O. Abioye, and J. Pasha, "A self-adaptive evolutionary algorithm for the berth scheduling problem: Towards efficient parameter control," *Algorithms*, vol. 11, p. 100, July 2018.

[6] C. Arango, P. Cortés, A. Escudero, and L. Onieva, "Genetic algorithm for the dynamic berth allocation problem in real time," in *Swarm Intelligence and Bio-Inspired Computation* (X.-S. Yang, Z. Cui, R. Xiao, A. H. Gandomi, and M. Karamanoglu, eds.), ch. 17, pp. 367–383, Elsevier, 2013.

[7] B. A. Akinnuwesi, O. B. Yama, A. M. Rahman, and S. G. Fashoto, "Berth allocation model for container terminal using genetic algorithm technique: Case of apapa wharf, lagos, nigeria," *Journal of Computer Science and Its Application*, vol. 28, June 2021.

[8] S. Aslam, M. Michaelides, and H. Herodotou, "Dynamic and continuous berth allocation using cuckoo search optimization," in *Proceedings of the 7th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS2021)*, April 2021.

[9] N. Umang, M. Bierlaire, and I. Vacca, "The berth allocation problem in bulk ports," in *Proceedings of Swiss Transport Research Conference (STRC)*, April 2011.

[10] S. Aslam, M. P. Michaelides, and H. Herodotou, "Enhanced berth allocation using the cuckoo search algorithm," *SN Computer Science*, vol. 3, p. 325, June 2022.

[11] S. Aslam, M. Michaelides, and H. Herodotou, "Optimizing multi-quay berth allocation using the cuckoo search algorithm," in *Proceedings of the 8th International Conference on Vehicle Technology and Intelligent Transport Systems*, March 2022.

[12] E. Ursavas, "Priority control of berth allocation problem in container terminals," *Annals of Operations Research*, vol. 317, pp. 805–824, 2022.

[13] B. Li, Z. Elmi, A. Manske, E. Jacobs, Y. yip Lau, Q. Chen, and M. A. Dulebenets, "Berth allocation and scheduling at marine container terminals: A state-of-the-art review of solution approaches and relevant scheduling attributes," *Journal of Computational Design and Engineering*, vol. 10, pp. 1707–1735, June 2023.

[14] A. Agra and F. Rodrigues, "Distributionally robust optimization for the berth allocation problem under uncertainty," *Transportation Research Part B*, vol. 164, pp. 1–24, August 2022.

[15] M. Yu, Y. Lv, Y. Wang, and X. Ji, "Enhanced ant colony algorithm for discrete dynamic berth allocation in a case container terminal," *Journal of Marine Science and Engineering*, vol. 11, p. 1931, October 2023.

[16] P. Razzaghi, A. Tabrizian, W. Guo, S. Chen, A. Taye, E. Thompson, A. Bregeon, A. Baheri, and P. Wei, "A survey on reinforcement learning in aviation applications," *Engineering Applications of Artificial Intelligence*, vol. 136, p. 108911, October 2024.

[17] A. Budipriyanto, B. Wirjodirdjo, I. N. Pujawan, and S. Gurning, "A simulation study of collaborative approach to berth allocation problem under uncertainty," *The Asian Journal of Shipping and Logistics*, vol. 33, pp. 127–139, September 2017.

[18] L. Kolley, N. Rückert, M. Kastner, C. Jahn, and K. Fischer, "Robust berth scheduling using machine learning for vessel arrival time prediction," *Flexible Services and Manufacturing Journal*, vol. 35, pp. 29–69, September 2023.

[19] M. Jahangard, Y. Xie, and Y. Feng, "Leveraging machine learning and optimization models for enhanced seaport efficiency," *Maritime Economics Logistics*, February 2025.

[20] M. K. Yadav, "Transforming the shipping industry: Integrating ai-powered virtual port opera-

tors for end-to-end optimization," *Journal of Advanced Research Engineering and Technology (JARET)*, vol. 3, pp. 245–257, June 2024.

[21] X. Chen, D. Ma, and R. W. Liu, "Application of artificial intelligence in maritime transportation," *Journal of Marine Science and Engineering*, vol. 12, p. 439, March 2024.

[22] J. Kosiek, A. Kaizer, A. Salomon, and A. Sacharko, "Analysis of modern port technologies based on literature review," *Gdynia Maritime University Journal*, vol. 15, p. 22, September 2021.

[23] M. M. Karim, A. S. M. A. Buhyian, B. T. Costa, and M. A. Rahman, "Automated handling of port containers using machine learning," in *Proceedings of MARTEC 2022, The International Conference on Marine Technology*, December 2022.

[24] A. Carboni, F. Deflorio, C. Caballini, and S. Cangelosi, "Advances in terminal management: simulation of vehicle traffic in container terminals," *Maritime Economics Logistics*, August 2024.

[25] Z. Yue and J. Mangan, "A framework for understanding reliability in container shipping networks," *Maritime Economics Logistics*, vol. 26, pp. 523–544, October 2024.

[26] S. Yin, M. Jin, H. Lu, G. Gong, W. Mao, G. Chen, and W. Li, "Reinforcement-learning-based parameter adaptation method for particle swarm optimization," *Complex Intelligent Systems*, vol. 9, pp. 5585–5609, March 2023.

[27] K. Azizzadenesheli, E. Brunskill, and A. Anandkumar, "Efficient exploration through bayesian deep q-networks," in *2018 Information Theory and Applications Workshop (ITA)*, pp. 1–9, 2018.