ANTONIO ROSA

A DATA ANALYSIS AND VISUALIZATION FRAMEWORK TO SUPPORT THE DESIGN OF DIGITAL SOLUTION FOR THE SEA & DRY PORTS

Relatório de Dissertação/projeto de investigação do Mestrado em Engenharia de Software

JÚRI

Presidente: (Prof. Dr., José António Moinhos Cordeiro, Professor Coordenador DEI, Instituto Politécnico de Setúbal)

Orientador: (Prof. Dr., Miguel Angel Guevara López, Professor adjunto DEI, Instituto Politécnico de Setúbal)

Vogal: (Prof. Dr. Osvaldo Rocha Pacheco, Professor Auxiliar DETI, Universidade de Aveiro)

(December 2025)

Acknowledgement

First and foremost to God for giving me life, to my family for unwavering support, to APS (Administração dos Portos de Sines e do Algarve) for proving us with the dataset and an expert who aided us in achieving our objective, to my colleagues of work package 3.7 with whom we spent times in meetings and conferences, finally my advisor Prof. Dr. Miguel Lopéz who believed in me and helped me to navigate through the whole process, and of course to the scientific community of IPS specifically the Systems and Information Technology Department of Escola Superior de Tecnologia de Setúbal for providing me an outstanding advisor and allowing me to present this research work.

Resumo

O setor global de transporte marítimo de contentores, que se provou ser um pilar do comércio internacional, gera quantidades consideráveis de dados operacionais. No entanto, os portos marítimos frequentemente enfrentam dificuldades para transformar esses dados brutos em informações úteis devido a sistemas isolados e à falta de ferramentas integradas de monitoramento em tempo real. Esta pesquisa aborda a lacuna crítica na tomada de decisões baseada em dados para operadores portuários, projetando a implementação e validação de um pipeline totalmente automatizado para o monitoramento de Indicadores-Chave de Desempenho (KPIs).

A metodologia envolveu um processo estruturado de curadoria de dados aplicado a um conjunto de dados de 341.408 registos brutos de movimentação de contentores de um terminal importante. Esse processo incluiu a remoção de dados duplicado, o tratamento de valores ausentes preenchidos como ("Desconhecido"), a padronização de data e hora e a filtragem, resultando em um conjunto de dados limpo e validado de 319.614 registos. Um banco de dados PostgreSQL serviu como data warehouse otimizado, com visualizações materializadas pré-computando os principais KPIs. Os dados processados foram visualizados por meio de um painel interativo construído no Metabase, apresentando 12 cartões distintos que monitoram métricas como movimentação total de contentores, duração média de armazenamento, uso do modo de transporte e atividade do terminal.

ETL (Extração, Transformação e Carregar) baseado em Python reduziu o conjunto de dados em 6,4% por meio da limpeza, a camada de banco de dados possibilitou consultas eficientes e o painel do Metabase forneceu aos stakeholders uma visão intuitiva e em tempo real das operações. O sistema foi conteinerizado usando Docker para implantação portátil e automatizado via agendamento cron. Este trabalho contribui com uma estrutura prática e escalável para análise logística portuária, indo além de modelos teóricos para uma solução implementada. Ele comprova que a curadoria e a visualização automatizadas de dados podem aprimorar significativamente a visibilidade operacional. Trabalhos futuros se concentrarão na integração de fluxos de dados em tempo real e análises preditivas para fazer a transição do monitoramento descritivo para o prescritivo.

Palavras-chave: Pipeline de dados, Monitoramento de KPIs, Logística portuária, Curadoria de dados, Painel de controle, Metabase,
PostgreSQL, Operações de terminais de contentores, Automação.

Abstract

The global container shipping industry, has proven to be a cornerstone of international trade, generates sizable amounts of operational data. However, seaports often struggle to transform this raw data into actionable insights due to siloed systems and a lack of integrated, real-time monitoring tools. This research addresses the critical gap in data-driven decision-making for port operators by designing, implementing, and validating a fully automated pipeline for Key Performance Indicator (KPI) monitoring.

The methodology involved a structured data curation process applied to a dataset of 341,408 raw container movement records from a major terminal. This process included deduplication, handling of missing values (filled with Unknown), datetime standardization, and filtering, resulting in a cleaned and validated dataset of 319,614 records. A PostgreSQL database served as the optimized data warehouse, with materialized views pre-computing core KPIs. The processed data was visualized through an interactive dashboard built in Metabase, featuring 12 distinct cards that monitor metrics such as total movements, average storage duration, transport mode usage, and terminal activity.

The results demonstrate a successful end-to-end integration: the Python-based ETL (Extract, Transform, Load) pipeline reduced the dataset by 6.4% through cleaning, the database layer enabled efficient querying, and the Metabase dashboard provided stakeholders with an intuitive, real-time view of operations. The system was containerized using Docker for portable deployment and automated via cron scheduling. This work contributes a practical, scalable framework for port logistics analytics, moving beyond theoretical models to a deployed solution. It proves that automated data curation and visualization can significantly enhance operational visibility. Future work will focus on integrating real-time data streams and predictive analytics to transition from descriptive to prescriptive monitoring.

Keywords: Data Pipeline, KPI Monitoring, Seaport Logistics, Data Curation, Dashboard, Metabase, PostgreSQL, Container Terminal Operations, Automation.

Contents

Acknowledgement i

Resumo ii

Abstract iii

List of Figures viii

List of Tables ix

List of Acronyms x

1 1

1.1 Introduction 1

1.2 Problem Statement 2

1.3 Research Question 2

1.4 Main Objective 3

1.5 Specific objectives 3

1.5.1 Scientific Objectives 4

1.5.2 Practical Objectives 4

1.6 Contributions 4

1.6.1 Methodological Contribution 4

1.6.2 Technical Contribution 5

1.7 Structure of the Thesis 6

2 7

2.1 Introduction 7

2.2 Port Digitalization and Data-Driven Operations 7

2.3 Data Curation and Data Quality in Operational Systems 9

2.4 ETL Pipelines and Data Processing Architectures 11

2.5 KPI Monitoring and Business Intelligence in Port Operations 13

2.6 Development Segments of the Project 14

2.6.1 Usability and User-Centric Design of the Pipeline 15
2.7 Usability Heuristics 15
2.7.1 Principal Components of the Pipeline's Usability. 15
2.7.2 Multi-Stakeholder Perspectives 16
2.7.3 Usability and User-Centric Design of the Pipeline 16
2.7.4 Usability and User-Centric Design of the Pipeline 16
2.8 Existing Solutions for Port Data Management 17
2.8.1 Terminal Operating Systems (TOS) 17
2.8.2 Port Community Systems (PCS) 17
2.8.3 Academic Prototypes and Pilot Systems (APPS) 17
2.9 Identified Gap 19
2.9.1 Research Gap and Motivation 19
2.10 Point of Departure: How This Research Differs. 19
3 20
3.1 Introduction 20
3.2 Data Quality Framework 20
3.3 Methodological Approach Overview 20
3.4 Data Acquisition 21
3.5 Data Cleaning and Curation 23
3.5.1 Cleaning procedures included: 23
3.6 Preprocessing and Feature Engineering 24
3.6.1 Key preprocessing operations 24
3.7 Exploratory Data Analysis (EDA) Strategy 25
3.8 Temporal Modelling 28
3.8.1 Modelling approach 28
3.9 Database Design and KPI Computation 28
3.9.1 Database schema 29
3.9.2 KPI Computation 29
3.10 Automation and Reproducibility 34

3.10.1 Key automation components 34
3.11 Conclusion 35
4 36
4.1 Introduction 36
4.2 Requirements and Design Principles 36
4.2.1 Architecture Overview 36
4.2.2 Functional Requirements 37
4.2.3 Non-Functional Requirements 37
4.3 High-Level Architecture 38
4.3.1 Architecture Rationale 39
4.4 ETL Pipeline Architecture 39
4.4.1 Data Acquisition Module 39
4.4.2 Data Cleaning Module 39
4.4.3 Preprocessing & Feature Engineering Module 40
4.4.4 Loader Module 40
4.5 Database Layer 40
4.5.1 Schema Design 40
4.5.2 Materialized Views for KPI Computation 41
4.6 Visualization Layer (Metabase) 41
4.6.1 Dashboard Structure 42
4.6.2 Automated Metabase Integration 42
4.7 Containerization and Orchestration 42
4.7.1 Docker Architecture 42
4.7.2 Scheduling with Cron 43
4.8 Execution Environment 44
5 45
5.1 Introduction 45
5.2 Data Quality Improvements 45
5.2.1 46

5.2.2 Duplicate Removal 46
5.2.3 Missing Value Standardization 46
5.2.4 Timestamp Correction 46
5.2.5 structural Reduction of Dataset 47
5.3 KPI Results 47
5.3.1 Throughput and Container Stays 47
5.3.2 Storage Duration 48
5.3.3 Transport Mode Distribution 48
5.3.4 Yearly Growth Patterns 49
5.3.5 Activity Over Time 50
5.4 System Performance Evaluation 50
5.4.1 ETL Pipeline Runtime 50
5.4.2 Database Query Performance 50
5.4.3 Dashboard Responsiveness 50
5.5 Discussion of Findings 51
5.5.1 Operational Insights 51
5.5.2 Methodological Insights 51
5.5.3 Comparison to Literature 51
5.6 System Validation 51
5.6.1 Expert Review 52
5.6.2 Manual Verification 52
5.6.3 Software Reliability 52
5.7 Limitations 52
6 54
6.1 Conclusion 54
6.2 Future Work 54
7 Bibliography. 55

List of Figures

Figure 1.1 Project workflow, (Author) 5
Figure 2.1 Module Segment process, (Author) 15
Figure 3.1 Pipeline Orchestration sequence, (Author) 35
Figure 4.1 Architecture Overview, (Author) 37
Figure 4.2 Automated pipeline System High-Level Architecture process, (Author) 38
Figure 4.3 Working Containers in the Docker compose, (Author) 43
Figure 5.1 Avarage Movement per Container, (Author) 48
Figure 5.2 Most used mean of transport, (Author) 49
Figure 5.3 Container Yearly Behavior, (Author) 49

List of Tables

Table 2.1 Comparison of existing Port Solutions, (Author) 18
Table 3.1 Dataset information about nexus_container_movement_psa, (Author) 22
Table 3.2 Logg of dataset information about nexus_container_movement_psa, (Author) 23
Table 3.3 Numerical Column Basic Statistics, (Author) 25
Table 3.4 Automated Calculation for each Categorical Count, (Author) 27
Table 3.5 Implemented Key Performance Indicators (KPIs), (Author) 30
Table 4.1 Cron process sequence lifecycle, (Author) 44
Table 5.1 Automated quality Report of processed dataset, (Author) 46
Table 5.2 Throughput and container Stay, (Author) 47
Table 5.3 Avarage Storage Duration, (Author) 48

List of Acronyms

ACID	**Atomicity Consistency Isolation Durability**
API	**Application Programming Interface**
APPS	**Academic Prototypes and Pilot Systems**
APS	**Administração dos Portos de Sines e do Algarve**
BI	**Business Intelligence**
CDT	**Container Dwell Time**
CRISP-DM	**Cross Industry Standard Process for Data Mining**
CSV	**Comma Separated Value**
CTS	**Categorical time series**
DAQ	**Data Acquisition**
DBMS	**Database Management System**
EDA	**Exploratory Data Analysis**
ETD	**Estimated Times of Departure**
ETL	**Extract Transform Load**
GIGO	**Garbage In Garbage Out**
IoT	**Internet of Things**
KPIs	**key performance indicators**
ML	Machine Learning
OLAP	**Online Analytical Processing**
OSS	**Open-Source Software**
PCS	**Port Community Systems**
SQL	**Structured Query Language**
SQuaRE	**Systems and software Quality Requirements and Evaluation**
TOS	**Terminal Operating Systems**
TPACK	**Technological Pedagogical Content knowledge**
UN/CEFACT	**United Nations Centre for Trade Facilitation and Electronic Business**

1

Chapter 1 Introduction

1.1 Introduction

Seaport terminals are critical nodes in global logistics networks, serving as intermodal platforms where containers are received, processed, stored, and dispatched. As international trade volumes increase and vessel sizes continue to grow, port operators face mounting pressure to improve operational efficiency, reduce congestion, and maintain service reliability. This evolution has intensified the demand for data-driven decision-making solutions, where operational data must be transformed into timely, reliable, and actionable performance indicators.

Despite the availability of large datasets generated by daily operations in ports' terminals, many ports continue to struggle with fragmented information systems, inconsistent data formats, and manual reporting practices. Raw materials (operational) data are often dispersed across heterogeneous sources, limiting the ability of port authorities and terminal operators to obtain a consistent and up-to-date view of key metrics such as container dwell time, equipment usage, transport mode distribution, and overall activity. As a result, performance monitoring becomes reactive, error-prone, and unable to support strategic planning or real-time operational adjustments.

This challenge is particularly relevant in the context of digitalization initiatives such as the Sines NEXUS Agenda (project), which aims to strengthen technological capabilities across Portuguese port infrastructures. Within Work Package 3.7, a key objective is to enhance performance management practices for sea and dry ports through improved data integration, automation, and analytical capabilities. In this context, the development of an automated pipeline capable of transforming raw operational data into validated Key Performance Indicators (KPIs) constitutes an essential step toward enabling smarter, more efficient port operations.

This thesis addresses this need by designing, implementing, and validating an automated data curation and visualization pipeline for container movement data. The proposed solution integrates data cleaning, preprocessing, KPI computation, and dashboard deployment within a modular and

reproducible architecture built entirely on open-source technologies. The pipeline aims to overcome the persistent issues of data fragmentation, inconsistent data quality, and manual reporting, offering a scalable framework for real-time operational monitoring.

1.2 Problem Statement

Although ports increasingly rely on digital systems, the underlying operational data remain difficult to exploit due to several persistent challenges:

- **Data Fragmentation:** Container movement records originate from multiple systems and lack a unified schema.
- **Inconsistent Data Quality:** Duplicates, missing values, incorrect timestamps, and non-standardized categories undermine analytical reliability.
- **Lack of Automation:** Many terminals still depend on manually updated spreadsheets or ad-hoc queries to compute KPIs.
- **Limited Analytical Capability:** Static reports and non-interactive tools prevent stakeholders from performing drilldowns or exploring patterns.

[1] NEXUS Pacto de Inovação – Transição Verde e Digital para Transportes, Logística e Mobilidade. Código de operação: 02-C05-i01.01-2022.PC645112083-00000059 (projeto 53).
https://ips.pt/wp-content/uploads/2025/05/NEXUS-Ficha-de-projeto.pdf

These limitations reduce situational awareness, hinder response time, and restrict the potential for optimization. A structured, automated, and reproducible data pipeline is required to transform raw operational data into accurate and accessible KPIs.

1.3 Research Question

This thesis is guided by the following research question / hypothesis:

How can an automated Extract Transform and Load (ETL) pipeline be designed and implemented to transform raw container movement data into reliable KPIs that support interactive and near real time performance monitoring in seaport operations?

1.4 Main Objective

The main objective of this work is to design, implement, and validate an automated data curation and dashboarding pipeline capable of transforming raw operational data from seaport terminals into structured datasets and KPIs suitable for interactive monitoring.

Such dashboards not only enable monitoring of operational efficiency but also facilitate benchmarking across terminals, predictive analytics, and policy evaluation. This thesis is motivated by the need to develop an automated, and robust pipeline capable of acquiring, cleaning, processing, and visualizing port container data in an integrated manner. We hope to improve the gape of identified problem of unprocessed or in some cases not well processed data which leads to delays in terminal management, with the use of dashboards for a real-time and dynamic monitoring tool for port users and stakeholders to reinforce the implementation of Smart Gates and Smart Terminal concept.The use of key performance indicators is very important in benchmarking any type of operations. Only by measuring the performance of operations it is possible to manage and improve. [1]. Performance management theory is that “if you cannot measure it, you cannot improve it’’ Designing and selecting appropriate KPIs is vital for measuring and monitoring performance [2].The dashboard that will be produced aims to verify and validate the data that results from data integration from several sources into a single cohesive view, offering flexible display options to satisfy different user profile needs, enable drill-down capabilities for comprehensive analysis, and support proactive decision-making in real-time. For example, with a well cleaned dataset, the dashboard can display metrics such as average container storage duration, Transport Usage Duration, Container Yearly Behaviour with accuracy and ttrustworthiness.

1.5 Specific objectives

We will be exploring data analysis and visualization algorithms and methods. Specifically, we will be using python library for designing data Curation, analysis and visualization framework that will be using the main key performance indicators information collected in the various objects inserted into a dataset to provide relevant KPI data, allowing a more efficient and precise ports' management solution enhancements according to pilot results dissemination and Promotion. The approach is used to foster and manage long-term growth and competitive advantage within the organization through people, systems, and procedures. Each perspective reveals a unique set of indicators that function as dashboards. These indicators provide critical knowledge factors that support the monitoring of business strategies and the design and schedule of organizational processes [3]. Therefore, the objectives of our research can be divided into scientific and practical goals:

1.5.1 Scientific Objectives

1. Develop a reproducible data curation methodology for operational port datasets.
2. Define and formalize KPIs relevant to seaport container operations, which can be supported by frequently collected data sources.
3. Investigate appropriate data quality assessment techniques for industrial datasets.
4. Design a modular ETL architecture adaptable to different port contexts.

1.5.2 Practical Objectives

1. Clean, preprocess, and structure the container movement dataset provided by Administração dos Portos de Sines e do Algarve - APS.
2. Implement a PostgreSQL analytical database with materialized views for KPIs computation.
3. Construct an interactive dashboard in Metabase to visualize operational metrics.
4. Automate the entire pipeline using Docker and scheduled execution mechanisms.

---

[2] Beyond BI: other problems you can solve with Metabase 2025 Accessed: Nov. 23, 2025. [Online]. Available: https://www.metabase.com/learn/metabase-basics/overview/beyond-bi

1.6 Contributions

We produced two primary contributions:

1.6.1 Methodological Contribution

A structured and reproducible approach for transforming raw container movement data into validated KPIs, integrating cleaning, preprocessing, temporal modelling, and analytical computation.

The choice ensures realistic applicability in real port operations while adhering to scientific precision. This research differentiates itself from state-of-the-art by addressing the identified limitations through the following contributions:

Holistic KPI Integration; unlike Terminal Operating Systems (TOS) (siloed) or Port Community Systems (PCS) (document-focused), our pipeline integrates operational KPIs (e.g., equipment operation, storage duration) with logistical and temporal KPIs (e.g. Avg Storage Hours per Stay, total stay duration) into a single consistent view to create digital representations of physical and functional characteristics of the terminal area [3].

The pipeline is built on a stack of open-source technologies (Python, PostgreSQL, Metabase), presenting

it as a transparent, customizable, and cost-effective choice to commercial solutions. The whole architecture is documented for reproducibility, a contribution to the academic community.

Features like automated anomaly detection can significantly reduce the manual effort required to ensure data consistency and quality The methodology includes precise data quality checks and validation frameworks [4]. To ensure the reliability of the KPIs presented, a step often assumed but not explicitly detailed in commercial systems. Integrated system lets governance tasks become automated through schema validation and data profiling and quality grading operations that were formerly manual and error prone [5]. The end-to-end automation from raw Comma Separated Value (CSV) to a production-ready dashboard demonstrates a scalable engineering approach that surpasses the manual, report-generation processes still common in many ports.

1.6.2 Technical Contribution

A fully functional framework prototype composed of:

- Modular Python scripts for data acquisition, cleaning, and transformation.
- A PostgreSQL-based analytical layer with optimized materialized views.
- An automated Metabase dashboard generation process.
- A containerized deployment enabling reproducibility and scalability.

The resulting system demonstrates that open-source technologies can be effectively combined to support operational intelligence in seaport environments. This integration ensures that Raw data is curated, and KPIs are updated automatically and presented interactively to stakeholders. The whole process will follow a workflow as shown in Figure 1.1

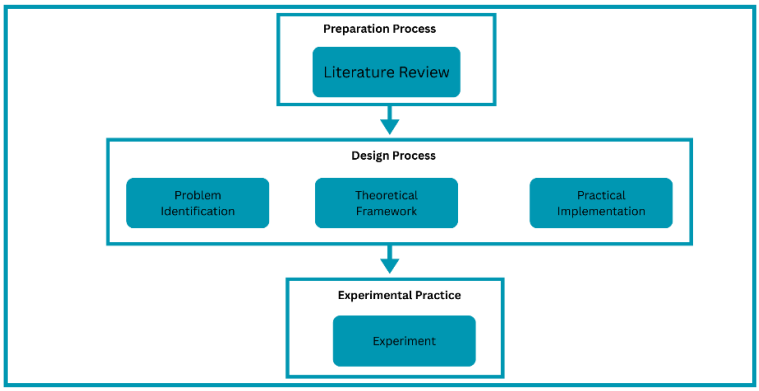


Figure 1.1 Project workflow

1.7 Structure of the Thesis

The thesis is organized as follows:

Chapter 1

- presents the theoretical foundations and literature review on port digitalization, data curation, ETL pipelines, KPIs, and existing solutions.

Chapter 2

- describes the methodology used to design the data pipeline, including data cleaning, preprocessing, modelling, and KPI computation.

Chapter 3

- details the system architecture and implementation, covering the ETL pipeline, database design, dashboard construction, and deployment.

Chapter 4

- presents and discusses the results, including data quality improvements, KPI outcomes, system performance, and validation.

Chapter 5

- concludes the thesis, summarizes the main contributions, outlines limitations, and proposes directions for future work.

2

Chapter 2 Theoretical Framework and Literature review

2.1 Introduction

In this chapter, we present the theoretical foundations and existing research relevant to the design of an automated data curation and visualization pipeline for seaport container operations. It synthesizes literature on port digitalization, data curation, ETL processes, KPI frameworks, and business intelligence systems, and identifies the research gap addressed by this thesis. The purpose is to position the proposed solution within established knowledge and justify its methodological and technical choices.

A data-driven, graphical user interface that combines and visualizes real-time and historical Key Performance Indicators (KPIs) derived within a maritime logistics ecosystem. It employs information visualization principles and human-computer interaction (HCI) methodologies to provide stakeholders with an at-a-glance understanding of complex operational performance, enabling proactive decision-making, anomaly detection, and strategic planning through features like drill-down capabilities, filtering, and alert mechanisms.

2.2 Port Digitalization and Data-Driven Operations

Seaport terminals have undergone a rapid digital transformation driven by increasing trade volumes, operational complexity, and the need to improve efficiency. Modern ports are expected to operate with high reliability and minimal downtime, requiring accurate, timely, and granular information about container flows and terminal performance with facilitated Just-in-Time manufacturing and offshoring key drivers in Container shipping industry[6]. The concept of *Smart Ports* encompasses the integration of digital technologies, such as data analytics, IoT, automation, and decision-support systems to create efficient, transparent, and interconnected port operations. Looking feather on BI concept which clarifies the approach to data collection and transformation into persuasive information to enhance organizational decision-making; the concept is a slogan in this digital era of business when aiming to achieve organizational excellence and competitive advantages. Organizations are investing vast resources to develop their tailored version of BI by utilizing Artificial Intelligence (AI) technology using big data and the Internet of Things (IoT). [7]

Despite these technological advancements, many terminals continue to rely on siloed information systems where operational data are spread across multiple platforms with limited interoperability. This fragmentation reduces the ability to perform integrated operational analysis and restricts real-time monitoring capabilities, special projects like Belt and Road Initiative[8]. Consequently, the availability of high-quality and harmonized data becomes crucial for effective port governance, optimization, and strategic planning. In this sense, monitoring port performance is a key component of digitalization efforts. KPIs, such as container dwell time, vessel turnaround time, and yard utilization, form the foundation for evaluating operational efficiency [9]. The growing emphasis on data-driven decision-making platforms underscores the need for robust data processing pipelines capable of transforming raw operational data into reliable performance insights.

The convergence of several established scientific fields such as Information Visualization (InfoVis): which uses interactive, visual representations of abstract data to amplify reasoning. Almost half of human neural tissue is directly and indirectly associated with vision, enabling sophisticated abilities related to pattern recognition, particularly when visually Stimulated [10]. Therefore, visualization emerges as a natural strategy for representing complex data. Dashboards are an application of InfoVis principles to amplify situational awareness in a specific domain. From a Human-Computer Interaction (HCI) perspective, data visualizations are visual representations that improve users' cognitive capabilities during a task [11].

The employment of robust analytics is not a technological comfort but an operational necessity for modern seaports, driven by economic Pressure; ports operate in a highly competitive global market. Analytics directly adjust resource allocation (cranes, labor, space), decrease vessel turnaround times, and minimize costly delays, directly affecting the port's economic viability and appeal to shipping lines. Scheduling various equipment is fundamental as it ensures optimal operational efficacy within container terminals and effectively manages the coordination of key equipment [12]

Supply Chain Resilience can help in responding and recovering from unexpected events and disruptions to keep supply chain operations on track by preserving enough connectedness and supervising its structure and function [13]. As highlighted by the same author, ports are critical choke points in the post COVID-19 era, changed freight patterns and trade routes, intensified anti-globalization situation, and deteriorating geopolitical conflicts exert great pressure on the maritime supply chain [13]. Analytics provide the visibility and predictive capability needed to foresee disruptions and redirect resources, enhancing the entire supply chain's resilience, the concept has become a prevalent business practice, as industrial struggle has intensified to be no longer just between individual enterprises, but among supply chains [14]

Regulatory pressure to decrease emissions is increasing. The adoption of the green port concept became possible through pilot programs to objectively estimate and implement new technologies (IoT and Smart port), alternative fuels, substitute energy sources, and waste administration [15]. Smart port infrastructures, such as IoT and AI, can enhance the sustainability and efficiency of port operations through progressed logistics and operational procedures [16]. Analytics can improve equipment routes to minimize

fuel consumption, decrease truck idling times at gates, and support the transition to more sustainable port operations.

Concepts such as data cleaning and preprocessing are crucial stages in the data preparation, as they clearly involve detecting and correcting (or removing) corrupt, incomplete, or irrelevant records from a dataset. As we aim to create a well-structured dataset that can be used for modeling, and to improve data quality by addressing most common issues identified by Joenssen et al, such as missing values, noisy data (errors, outliers), and inconsistencies [17]. Data preprocessing and handling transforms data into truly useful information for organizations, letting them to analyze trends, optimize metrics and detect patterns; the process includes techniques like parsing, transformation, normalization, standardization, and aggregation to convert raw, unstructured, or inconsistent data into a clean, structured format suitable for analysis and modeling [18]. High-quality data cleaning is a prerequisite for producing reliable and valid analytical results, as the principle of "garbage in, garbage out" (GIGO) is paramount in data science [19].

The Analytics Layer is a component in a data architecture that contains computational logic and algorithms for processing data to generate insights, predictions, and knowledge. This layer, also known as semantic layers, improves data visualization by integrating impeccably with BI tools. Such integration allows the creation of clear, interactive reports and dashboards that are easy for users to understand and interact with [20]. Modern visualization platforms must support "real-time monitoring and observability" with minimal refresh latency to enable timely marketing decisions [21].

The Visualization Layer translates compound information and metrics into visual objects (e.g., points, lines, bars, maps) contained in charts, graphs, and dashboards [22]. The primary objective of this layer is to communicate information clearly and efficiently to end-users, enabling them to understand trends, outliers, and patterns in the data. Effective visualization follows a multi-dimensional data analysis platform (MuDAP) for researchers in cognitive science principles and graphic design [23].

2.3 Data Curation and Data Quality in Operational Systems

Data curation refers to the processes through which raw data are cleaned, organized, validated, and prepared for analysis [24]. According to the same author and available literature, data curation activity involves data collection, selection, erroneous data identification and correction, heterogeneity treatment, categorization and classification, metadata creation, and preservation. In a generic sense, data curation is an active data management activity which harnesses the data for further use in analysis. The concept is about work and actions taken by curators of a data repository to provide meaningful and enduring access to data, this encompasses a range of actions undertaken to ensure that research data are fit for purpose and available for discovery and reuse [25]. Active curation of these resources with accurate and flexible descriptions to check their availability, reliability and general quality of service is required. A well-curated resource would potentially enable reuse by improving knowledge of and about processes and hence avoid

wasteful reinvention; improve reliability by pooling operational histories, reputations, and validation by promoting best practice, verified procedures and popular processes [26]. Data curation tasks and procedures are commonly described with a research data lifecycle model. In this model, the decisions involved in a set of data curation are divided into abstracted steps. By performing data curation according to a lifecycle model, the data provider can perform each data curation task and procedure with accuracy and the data reuser can understand in detail the methodology and workflow used [27].

We also looked in literature available for other relevant subjects that aggregate value in our research. As the economy continues to develop around the globe, there is a growing demand for data acquisition speed and other indicators [28]. Component-based Regression Tests - CORTS captures runtime module-level reliance that are related to reflection and dynamic class filling mechanisms. This is possible because such dependencies are clearly defined inside the module descriptor [29]. Jensen et al, describes Data Acquisition (DAQ) as being the process of sampling signals from the physical world such as image, data which could apply to text as well and converting the resulting samples into a digital numeric format that can be manipulated by a computer and software, were segments are collected regularly using periodic sampling and a determined schedule for transmissions [30]. The process is critical as it establishes the foundational step in any data-driven pipeline, and the fidelity, accuracy, and frequency of acquisition directly constrain all subsequent analysis. In the context of information systems, data acquisition also extends to the automated collection of digital data from sources such as web APIs, transaction logs, and network streams. In this research case we followed the following process in modular form called data_loading.py which reads raw CSVs, validates schema, and prepares staging datasets for feather processing.

In statistics, Exploratory Data Analysis (EDA) is a method of analyzing datasets, summarizing their core characteristics through graphics and other visualization methods Exploratory Data Analysis is a philosophy and set of techniques pioneered by John Tukey for analyzing datasets to summarize their main characteristics, often employing visual methods [31]. The primary goal of EDA is to examine the data without strict expectations to discover fundamental patterns, spot anomalies, test hypotheses, and verify assumptions. It relies heavily on graphical representations (e.g., histograms, scatter plots, box plots etc.) and quantitative summaries (e.g., measures of central tendency, dispersion, and correlation). EDA is inherently an iterative process that emphasizes the role of visualization and detective work in identifying what the data can reveal beyond formal modeling or hypothesis testing. It is suggested that it is important to see EDA as an integral part of statistical inference [32].

On a conceptual view, a database management system (DBMS) is described as a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications; which means specifying the data types, structures, and constraints of the data to be stored in the database [33]. The Database Layer, also known as the data access layer or persistence layer, is a theoretical tier in a multi-tier software architecture responsible for

storage, retrieval, management, and integrity of data. It abstracts the underlying data storage technology (e.g., relational databases like PostgreSQL, NoSQL systems like MongoDB, or data warehouses) from the business logic and presentation layers [34]. Its core functions include providing a structured schema, supporting query languages (e.g., SQL), ensuring data consistency through ACID (Atomicity, Consistency, Isolation, Durability). This layer acts as the single source of veracity for an application's data.

Deployment orchestration depends on automated decision-making processes to streamline the software lifecycle, from development to deployment [35]. Orchestration is the automated configuration, organization, and management of complex computer systems, middleware, and services; the concept has become a commonly adopted standard for application deployment among medium to large-scale organizations [36]. In modern data infrastructure, this involves using containerization tools (e.g., Docker) and orchestration platforms (e.g., Rest API, Cron/Linux Scheduler) to define, run, and scale workflows as a collection of interdependent tasks.

In operational environments such as seaport terminals, data often originates from diverse sources and may contain inconsistencies, missing values, duplicates, or incompatible formats. These issues undermine trust in analytical outputs and hinder the production of accurate KPIs.

Data quality is typically characterized by several dimensions that include, but not limited to:

- Completeness: Absence of missing or null values in required fields
- Consistency: Uniform formatting and classification across datasets
- Accuracy: Correctness and reliability of recorded values
- Timeliness: Data reflects the appropriate time of events
- Uniqueness: Absence of duplicate records

Seaport datasets frequently violate these principles due to manual inputs, asynchronous system updates, and a lack of standardized data management practices. As a result, an effective data curation strategy including systematic cleaning, normalization, timestamp standardization, and error handling is essential for ensuring reliable downstream analytics. This is why the research in industrial data management emphasizes the importance of reproducible curation workflows, automated validation rules, and transparent procedures that allow datasets to be consistently prepared for analysis. These principles underpinned the design of the data channel that we proposed in this thesis.

2.4 ETL Pipelines and Data Processing Architectures

To embark on an automated framework, the concept of Extract, Transform, Load (ETL) pipeline brings a notion of a structured process where data is extracted from raw sources, transformed through cleaning and preprocessing, and loaded into the aimed system (often a database or visualization layer). In this project,

we looked at what can be the best ETL process that can be implemented in modular Python scripts (utils/) and integrated with PostgreSQL and Metabase. According to the scientific community, the ETL workflow deals with the gathered data to comply with extraction, processing, cleaning, integration and loading of the data [37]. ETL is organized by a workflow with several tasks, which are partitioned in three phases; the first phase is called Extraction, the main purpose of this phase is to gather data from operational sources. Wrappers, triggers, or log files are used to perform the process. For this purpose, communication to operational data sources is established, which is in our case the dataset provided by APS. The data source has different access paths and data format; other phase is named Transformation, This phase performs data cleaning, data integration, data filtering, data transformation, and data processing. In the second phase, there are removed conflicts of values, semantic, and structural conflicts, such as inconsistent values, synonyms, and duplicates. The transformed and cleaned data are stored in the data staging area. The last phase is called Loading; data of interest is loaded to the target data warehouse for further processing. The According to Assis Vilela “ETL workflow concept aims to store operational data in a consistent and integrated way [37]. Together, these concepts enable the creation of a data-driven decision support system that is both scientifically grounded and operationally relevant.

ETL pipelines are core components of modern data engineering, supporting the transition from raw data to analytical-ready datasets. As mentioned before, ETL processes typically involve:

1. Extraction: Importing data from source systems (e.g., CSV files, operational databases).
2. Transformation: Cleaning, filtering, standardizing, and enriching data.
3. Loading: Persisting processed data in a structured database or data warehouse.

Modular ETL architectures enable maintainable and scalable workflows by dividing processing tasks into discrete components. Automation plays a crucial role in ensuring that data pipelines remain consistent and up to date, particularly in operational contexts where fresh data are generated continuously.

Research highlights the importance of:

- Modular, reusable transformation functions
- Logging mechanisms for traceability
- Data schemas to enforce structure
- Scheduling mechanisms for periodic updates
- Containerization for environmental reproducibility

We use a methodology that follows these principles closely, integrating modular Python components, PostgreSQL materialized views, and Docker-based automation to achieve a robust and reproducible pipeline.

2.5 KPI Monitoring and Business Intelligence in Port Operations

Meanwhile Key Performance Indicators (KPIs) are quantifiable measures used to evaluate the success of an organization in achieving operational or strategic goals that quantify and enable information on a complex phenomenon, such as environmental impact, to be simplified into a form that is relatively easy to use and understand. The three main functions of indicators are quantification, simplification and communication [38]. Defining and selecting appropriate KPIs is vital for measuring and monitoring performance although there is a degree of subjectivity in what kind of KPIs are relevant for each problem.[39] In seaports, KPIs typically include container throughput, berth occupancy, dwell time, turnaround time, and yard utilization. These indicators provide actionable insights for stakeholders ranging from terminal operators to customs agencies as mentioned by Hinkka et al. [38] KPIs are essential tools for evaluating and improving the operational performance of ports and shipping fleets. They serve as quantifiable metrics that enable stakeholders to assess efficiency, reliability, safety, and sustainability [39].

The implementation of data analytics in container terminals is a rapidly evolving field, moving from traditional descriptive reporting to predictive and prescriptive analytics. The process encompasses the design and construction of one or more artifacts such as models, diagrams, dashboards, software, or other tangible elements; such a solution that uses the information collected in the various objects (sensing, equipment, integrated systems, operators' inputs, etc.) to provide relevant KPI data to be further investigated. collaborative activities, comprising augmented reality (AR) artifact creation and VT-based discussion, in reinforcing the technological pedagogical content knowledge (TPACK) [40]. This allows creativity. In our research case, these artifacts aim to solve the identified problem.

Business Intelligence (BI) systems transform structured data into charts, tables, and dashboards to support managerial decision-making.[7] In port environments, BI tools enable stakeholders to:

- Monitor operational activity
- Detect anomalies or delays
- Compare performance over time
- Perform drill-down analysis on cargo flows
- Support capacity planning and resource allocation

Effective KPIs provide simplified, quantifiable metrics that capture complex operational behavior. In seaport terminals, commonly used KPIs include:

- Container Yearly Behavior
- Movements per Container
- Movements by Transport Mode
- Transport Usage Duration

- Total Unique Stays
- Avg Gross Weight by ISO Type
- Terminal activity intensity

To be useful, KPIs must rely on **clean, validated, and consistently processed data**, highlighting the necessity of an integrated data curation pipeline.

Dashboards are a critical part of BI systems, providing visual interfaces that support intuitive interpretation of KPIs. Modern BI tools such as Metabase, Power BI, and Tableau emphasize usability, interactivity, filter controls, and real-time synchronization with backend databases. In operational environments, dashboards need to support both high-level overviews and detailed drill-down capabilities to meet the needs of different stakeholders (i.e., users' profiles: managers, planners, analysts). The prototype dashboarding framework implemented in this thesis follows widely accepted visualization principles: clarity, simplicity, consistency, and minimal cognitive load.

The dashboards concept alluded to interactive visual interfaces that aggregate and present data in real time; this process transforms raw data into meaningful patterns that support decision-making. According to Simran Sethi, with well-defined KPI strategies, dashboards can convert big data analytics to useful information [41]. In logistics, dashboards facilitate continuous monitoring, anomaly detection, and predictive analytics. Dashboards can provide both an overview and detailed views at the same time and might create transparency and can be used for accountability and stimulating engagement. The usage of data on dashboards does not result in transparency and accountability per se. Benefits can only be gained if dashboards are properly designed [42]. A dashboard system can be effective in providing adaptive support to address the difficulties that arise with analysis of big data. Typically, dashboards have a well-organized display consisting of visual elements such as graphs, charts, and alert mech-anisms with color codes, which enable users to intuitively capture critical information required in decision-making processes [43].

2.6 Development Segments of the Project

The modular design of the current project mirrors the development segments of modern data-driven systems. The reimplementation determination required to move to a real system is greatly augmented by the number of distinct data sources and protocols for assembling automation. A modular approach allows developers to reuse core components while interfacing with increasingly diverse external services or data processing amenities [44]. Figure 2.1 represents the modular segment process from beginning till the end.

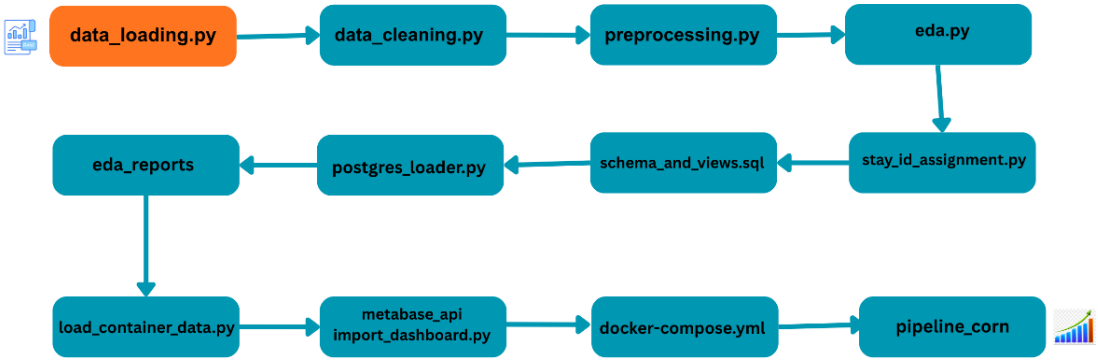


Figure 2.1 Module Segment process

2.6.1 Usability and User-Centric Design of the Pipeline

An evident usability of any automated system is the cornerstone of its effectiveness. A technically perfect pipeline is worthless if its output is not usable by the intended stakeholders. High usability ensures that stakeholders can effectively achieve their goals whether it's identifying a bottleneck, validating a hypothesis, or preparing a report. Poor usability leads to misunderstanding of data, rejection of the tool, and ultimately, failure to realize the intended return on investment. In contrast, involving stakeholders input through meetings and interviews would increase understanding of what KPIs are important at the employee level and make sure that those metrics are business drivers, not meaningless numbers [41].

2.7 Usability Heuristics

The dashboard design is informed by Nielsen's usability heuristics for user interface design [45]. Specifically, visibility of system status. The dashboard clearly shows the data's last refresh time. Match among system and the real world. KPIs use business terms (e.g., stay duration, turnaround time). User control and freedom give filters and drill-down capabilities. Consistency and standards adhere to common visualization conventions (e.g., time on the x-axis). Aesthetic and restrained design avoids overload of information; presents only the most relevant data.

2.7.1 Principal Components of the Pipeline's Usability.

Data accuracy and Trustworthiness which means the pipeline's automated data validation steps (Section 3.2) ensure users can trust the metrics described. Intuitive visual encoding Using right chart types for the data (e.g., line charts for trends over time, bar charts for comparisons). Interactive exploration, allowing users to filter by time range, Terminal Activity, avg_duration, etc., to answer their own specific questions.

2.7.2 Multi-Stakeholder Perspectives

The value proposition of the dashboard differs across user groups. There is a user perspective (Port Operators, Managers): the focus here is real-time operational awareness, execution benchmarking, rapid anomaly detection. Well-designed dashboards significantly support managers in visualizing data as well as spotting trends, which in turn aids them in overcoming information overload [41]. Developers' Perspective is how maintainable, scalable, and documented codebase is. The other concerned stakeholder is Regulatory bodies and Port authorities they are more interested in transparency, safety compliance, environmental monitoring, and overall port performance metrics for reporting and strategic planning. Usability and User-Centric Design of the Pipeline

Relevant Standards and Compliance

To ensure interoperability, data quality, and professional rigor, the pipeline aligns with several international standards where applicable such as ISO/IEC 25010: Systems and software Quality Requirements and Evaluation (SQuaRE) [46]. This standard provides a framework for evaluating software product quality.

In our pipeline we specifically address functional suitability: Accuracy and completeness of KPI calculations. Performance efficiency: Response time and resource utilization. Usability: Correctness recognizable features to use. Maintainability: The modularity of the pipeline and reusable possibilities that it provides. W3C Standards, for any web components, observance to web standards ensures accessibility and cross-browser compatibility. Data standards, While not explicitly implemented in the prototype, the design considers the importance of data standards like IMO Compendium on Facilitation and Electronic Business and UN/CEFACT standards for potential future integration with Port Community Systems [47].

2.7.3 Usability and User-Centric Design of the Pipeline

Practical Implementation.

A vital part of this process is the practical implementation of the artifacts in real situations to test the effectiveness and applicability of solutions proposals as in the case of Sines Container terminal. As we are aware, the artifact is part of a bigger project with an impact on stakeholders in deciding which part of the prototype to use and where to implement it. The process will involve collecting, curating and processing of data from port operations and solution enhancements according to pilot results.

2.7.4 Usability and User-Centric Design of the Pipeline

Assessment and Validation.

After implementation, artifacts are rigorously evaluated. This may include comparison with existing

solutions, performance testing and other methods to determine effectiveness, innovation and usefulness of the proposed solutions, dissemination and promotion. Quantifying the exact number of systems is challenging due to proprietary commercial solutions. However, market and academic research can be categorized into three main spectrums of application as mentioned by [48], [49], [50].

2.8 Existing Solutions for Port Data Management

Existing digital systems in ports typically fall into three categories:

2.8.1 Terminal Operating Systems (TOS)

Platforms such as NAVIS N4 or TERMINAL OPERATING SYSTEMS used by large terminals manage container movements, equipment control, and yard planning. They offer real-time operational data but often provide limited customization or transparency regarding KPI definitions. Furthermore, TOS platforms usually operate as closed systems with proprietary interfaces. Commercial Terminal Operating Systems (TOS) It manages the transfer of containers between trucks and the yard and trucks, trucks and vessels, and vessels and trucks, utilizing heavy lifting equipment. Major TOS like NAVIS N4, SOLAS, and COSMOS have integrated dashboard modules. Their focus is often on operational throughput (e.g., moves per hour, crane productivity) and vessel turnaround times [48]. They are strong on real-time data but can be siloed and less flexible for cross-functional KPI analysis

2.8.2 Port Community Systems (PCS)

PCSs aim to integrate data across port stakeholders (shipping lines, customs, truckers). Although they facilitate document exchange and visibility across the supply chain, they generally do not provide deep operational analytics or customizable KPI tools. Port Community Systems (PCS): is the technological platform that enables networking between the public and private agents and entities involved in the ship and cargo services offered by ports Systems like Portico (Barcelona), DA-Desk (Rotterdam), and others in Europe aim to integrate data across multiple stakeholders (terminals, haulers, customs, shipping lines). Their dashboards focus on visibility across the logistics chain,

tracking cargo status, and estimating truck turnaround times at gates [49].

2.8.3 Academic Prototypes and Pilot Systems (APPS)

Academic research frequently focuses on optimization algorithms berth allocation, yard planning, crane scheduling or simulation models. While these studies contribute valuable insights, most do not address the full data lifecycle from ingestion to dashboard deployment. Real-world validation is often limited due to data accessibility challenges. Academic and Open-Source Research Prototypes: Academic

prototyping is an attempt to reify an idea to a sufficient degree of fidelity where knowledge gained from prototyping can be applied back to the idea [50]. In academia, research institutions, especially in Europe where patents on software still do not exist, Open Source Software (OSS) is considered as a good mean for technology transfer [51]. Numerous research institutions have developed prototypes. For example: The PORTAL project in Portugal focuses on data sharing and interoperability for port logistics. Research often focuses on optimizing specific areas using simulation such as berth allocation, yard planning, or gate congestion prediction, rather than on comprehensive, interactive KPI dashboards. Table 2.1 shows the existing systems.

Table 2.1 Comparison of existing Port Solutions)

System Type	Primary Focus	Strengths	Limitations (Research Gaps)
Commercial TOS	**Operational throughput, Equipment utilization**	**Real-time data, High reliability, Vendor support**	**Siloed data, Limited customization, High cost, Proprietary formats**
Port Community Systems (PCS)	**Cross-stakeholder visibility, Document flow**	**Supply chain integration, Standardized data exchange**	**Often lack deep operational KPIs, Focus on documentation over analytics**
Research Prototypes	**Algorithmic optimization (e.g., AI, simulation)**	**Innovative, addresses specific optimization problems**	**Often not integrated into live operations, limited scalability, narrow scope**
Data Curation Prototype	**Automated cleaning of raw, noisy, and unprepared data**	**Provides data integrity, optimiaze data, Standardize data Acurate information usage Can be used in various business need.**	**Limited test on to one terminal, Limited functionalities of open-source software usage**

2.9 Identified Gap

Across all categories, a consistent gap exists:

There is not widely documented, open, fully automated, reproducible pipeline for transforming raw operational data into validated KPIs and dashboards.

This gap has been the motivation for the development of this thesis.

2.9.1 Research Gap and Motivation

The literature demonstrates a strong need for structured, automated data processing workflows in port environments. Existing systems either: (**TOS**) provide raw operational data without cleaning or standardization; (**PCS**) offer proprietary analytics with limited transparency; or Academic Prototypes and Pilot Systems (**APPS**) focus only on isolated analytical tasks without addressing the end-to-end data lifecycle. The combination of persistent data quality issues, fragmented information systems, and the need for near–real-time performance monitoring has created a clear gap in both practice and academic research. In this work we address that gap by designing a reproducible, open-source, and modular pipeline that integrates data curation, KPI computation, and dashboard visualization within a single automated framework.

2.10 Point of Departure: How This Research Differs.

The point of departure for this research is the recognition that existing approaches are too manual (traditional reports) and too infrastructure-heavy (big data systems requiring large-scale clusters).

In this research thesis we focus on the following approach:

- Lightweight, reproducible Python-based pipelines.
- Open-source tools (PostgreSQL, Metabase, Docker).
- Automation for continuous KPI monitoring.

3

Chapter 3 Methodology

3.1 Introduction

In this chapter we aim to present the methodological approach adopted to design and implement an automated ETL pipeline for transforming raw container movement data into reliable KPIs for port operations. The methodology we adopted in this research is structured around a data lifecycle model tailored for maritime container operations. The methodology begins with raw data acquisition, followed by systematic cleaning, preprocessing, and exploration analysis, culminating in the development of KPI-driven dashboards. The methodology follows a structured data engineering workflow inspired by the CRISP-DM framework and adapted to the requirements of seaport operational datasets [52]. It encompasses data acquisition, cleaning, preprocessing, temporal modelling, database design, KPI computation, and pipeline automation.

3.2 Data Quality Framework

Data quality is a cornerstone of reliable analytics. In our research contest the raw dataset (nexus_container_movement1.csv) comprised of 341,408 rows and 21 columns.

To ensure data integrity, the ISO 8000 Data Quality Standard [53]; was adopted as a guiding framework, emphasizing:

- Completeness: All required fields must be populated or explicitly marked as Unknown.
- Accuracy: Removal of invalid entries (e.g., malformed timestamps).
- Consistency: Standardization of categorical labels (e.g., Movement Report, Document Type).
- Uniqueness: Removal of fully identical duplicates, preserving unique operational records.
- Timeliness: Ensuring chronological ordering of movements.

3.3 Methodological Approach Overview

The methodological approach consists of five sequential stages:

1. **Data Acquisition:** Collection and initial inspection of raw data provided by the port authority.

2. **Data Cleaning and Curation:** Identification and correction of data quality issues including duplicates, missing values, and inconsistent formats.
3. **Preprocessing and Feature Engineering:** Transformation of curated data into analysis-ready structures, including the computation of temporal attributes.
4. **Exploratory Data Analysis Strategy:** Validation of the curated dataset through descriptive and structural checks.
5. **KPI Computation and Pipeline Automation:** Implementation of database logic, materialized views, and automated dashboard updates.

Each stage is executed through modular Python components containerized to ensure reproducibility.

3.4 Data Acquisition

Two datasets were provided by APS: *PR_JUL* and *nexus_container_movement_psa*. After consultation with domain experts, the second dataset was selected due to its detailed representation of container movements within Terminal XXI.

With the research work development, the stakeholders requested for different focus on the type of operational KPI that we should concentrate on, considering their need, and looking at the research emphasis which is to provide a practical concept that is scientifically proven to serve the port industry. With focus in seaport industry, using nexus_container_movement_psa from Port of Sines. Priore to data curation process we were able to have a glimpse of the structure of the information within the dataset. The table below gives detailed information found in the dataset before the curation process. There are 34141 entries (rows), and 21 columns, where the first 20 columns are object data type and 1 column which is a float64 data type.

The dataset was supplied in CSV format containing:

- Unique container identifiers
- Timestamps for operational events
- Document and movement types
- Transport modes
- Weight attributes
- Container status (full/empty)

Table 3.1 shows the outcome of the information in more detail.

Table 3.1 Dataset information about nexus_container_movement_psa.

```
Data columns (total 21 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   Dataset                      34141 non-null  object
 1   Document Type                34141 non-null  object
 2   Movement Report              34141 non-null  object
 3   Container Plate              34141 non-null  object
 4   Full Container?              34141 non-null  object
 5   ISO Type                     34141 non-null  object
 6   Gross Weight                 34141 non-null  float64
 7   Date Time of Movement        21082 non-null  object
 8   Stay Id                      34141 non-null  object
 9   Container Action             34141 non-null  object
 10  In Park?                     34141 non-null  object
 11  Open Stay?                   34141 non-null  object
 12  Inbound Date Time            31285 non-null  object
 13  Inbound Mean of Transport    34137 non-null  object
 14  Inbound Process Reference    34141 non-null  object
 15  Outbound Date Time           31068 non-null  object
 16  Outbound Mean of Transport   33517 non-null  object
 17  Outbound Process Reference   33517 non-null  object
 18  Terminal                     34141 non-null  object
 19  Terminal Entity              34141 non-null  object
 20  Application                  34141 non-null  object
dtypes: float64(1), object(20)
```

This raw dataset served as the sole input to the ETL pipeline. Initial inspection revealed typical data quality issues for operational systems, including inconsistent timestamp formats, missing values in categorical columns, and duplicated entries.

Datasets containing several sources of data are the foundation (raw materials) of any smart gate system. In the perspective of port operations, data can be classified as structured and unstructured. Structured data includes information such as container identifiers, vessel arrival and departure times, and cargo manifests. On the other hand, unstructured data encompasses video streams, sensor data, and weather reports among others [54].

As denoted by Zhang et al., Smart Port efforts include the use of new-generation digital information technologies such as the Internet of Things (IoT), cloud computing, big data, and intelligent sensing.[55] Datasets for real-time container tracking in port operations employing a big quantity of records may establish a piece of information which points out the detailed logic operations [56].

3.5 Data Cleaning and Curation

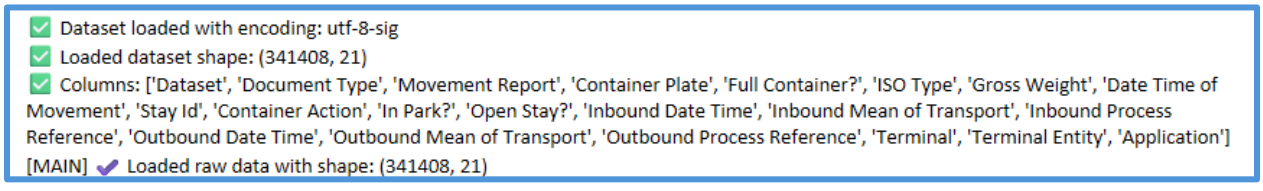
Data curation aimed to ensure completeness, consistency, accuracy, and structural validity. Cleaning processes were implemented in the Python script *data_cleaning.py* using the Pandas library.

3.5.1 Cleaning procedures included:

- **Duplicate Removal:**
  Exact duplicate rows were identified and removed, eliminating redundant entries caused by repeated data exports.
- **Missing Value Handling:**
  Missing entries in categorical fields were replaced with the explicit placeholder "Unknown" to preserve structural completeness.
- **Timestamp Standardization:**
  All timestamp fields were converted into ISO 8601 format. Invalid or malformed timestamps were corrected or removed according to predefined rules.
- **Categorical Normalization:**
  Document types, movement types, and container actions were cleaned by removing trailing spaces, harmonizing labels, and ensuring consistent spelling.
- **Column Pruning:**
  Non-essential fields were removed to reduce dimensionality, based on domain expert feedback.

These cleaning steps produced a curated dataset free of duplicates, structurally consistent, and ready for preprocessing. After curation, the cleaned dataset contained 319,614 records, representing a 6.4% reduction in volume due to deduplication and filtering. Key data quality metrics were documented, showing 0% missing values in categorical fields after cleaning and only 13 missing values in the numerical Gross Weight field as shown in table 3.2

Table 3.2 Logg of dataset information about nexus_container_movement_psa.



3.6 Preprocessing and Feature Engineering

Preprocessing transforms curated data into a structured format suitable for KPIs computation. This stage was implemented in *preprocessing.py* and included several transformations. Data pre-processing involves a series of steps to convert raw data derived from data extraction into a clean and tidy dataset preceding to statistical analysis [57]. The process can be iterative and may imply repeating this series of steps until the data is adequately consolidated for the purpose of statistical analysis which means converting raw but clean data into analysis-ready datasets. Such steps also involve concepts like feature engineering which aims on creating, selecting, and optimizing features to improve the predictive power of machine learning models. The step involves domain knowledge and creativity, as it includes transforming raw data into significant representations that capture core patterns and relationships [58]. In our case we used feature engineering for Creation of a new Stay IDs to uniquely identify container stays across multiple movements. Subdivision of movements into Inbound, Storage, and Outbound phases.

We used a stay_id_assignment.py to standardize numerical field which is Gross Weight for statistical analysis, translating categorical variables into consistent formats. Making use of preprocessing.py module to do calculation of durations between events like Inbound Arrival, Storage, Outbound Departure, anomalies identification such as negative or overlapping timestamps.

3.6.1 Key preprocessing operations

- **Event Ordering:**

  Records for each container were sorted chronologically based on the timestamp field.
- **Stay Identification:**

  A new attribute, *Stay_ID*, was generated to group all movements related to the same container stay. This enabled the reconstruction of inbound–storage–outbound sequences.
- **Duration Computation:**

  For each stay, temporal durations were computed using timestamp differences using core duration formular which gives a T_start, T_end concept as when time starts and when it ends. [59]

Three duration metrics were derived:

- Inbound duration
- Storage duration
- Outbound duration

- **Standardization of Fields**
  Weight attributes, container statuses, and transport modes were standardized for consistency.

The resulting preprocessed dataset contains enriched temporal values that are essential for KPI computation.

3.7 Exploratory Data Analysis (EDA) Strategy

Exploratory Data Analysis (EDA) was conducted to understand dataset distributions, detect outliers, and validate preprocessing results. Following Tukey's exploration framework which implies that data analysis is significantly visual, and he has several suggestions for graphical displays, such as graphs are used for many different purposes, they can be used to store quantitative data, to communicate conclusions, or to learn new information. Some types of plots are better for one purpose, and some are better for another [60]. We feather looked into the univariate analysis concept which is a minimal form of data analysis, where one variable is at a time analyzed in isolation [61]. The goal of this concept is to describe the distribution (how values are spread), central tendency (mean, median, mode) [62], dispersion (variance, standard deviation, interquartile range, range), shape (skewness, kurtosis), and to identify outliers. It's completely descriptive; it doesn't examine relationships between variables.

Meanwhile categorical response data seemed to be a universal form of data confronted by the applied statistics. Using statistical terms like Categorical time series (CTS) which are considered by taking values on a qualitative assortment consisting of a finite number of categories, which is referred to as ordinal range, if the categories show a natural ordering, or nominal range [63]. The analysis moves beyond simple counts to understand the arrangement of the dataset and the relationships between different categories or between a category and a numerical outcome as shown below in table 3.3 by our only numerical column.

Table 3.3 Numerical Column Basic Statistics.

Basic Statistics

Numerical Columns Statistics:

	count	mean	std	min	25%	50%	75%	max
Gross Weight	**319601.0**	**19522.757741**	**9348.250223**	**3.0**	**11226.0**	**22800.0**	**27668.0**	**204301.0**

Exploratory Data Analysis was performed to verify the correctness of the cleaning and preprocessing stages. This was not used to generate results, but to confirm:

- Logical validity of event sequences
- Distribution of key categorical fields
- Identification of anomalies (e.g., negative durations)
- Consistency across container stays
- Correctness of transport mode classification

Automated scripts generated descriptive statistics, summary tables, and consistency checks, allowing systematic validation of the curated dataset before loading it into the database.

The data quality evaluation and initial pattern discovery we used, were conducted through an automated Exploratory Data Analysis (EDA) pipeline we implemented in Python. Such a systematic approach ensured consistent, reproducible analysis and adhered to established data science methodologies for initial data characterization [60].

Percentage Distribution Computation. We analyzed categorical variables using the formula below.

$$P = \frac{\text{Category Count}}{\text{Total Records}} \times 100$$

Formular explanation.
P: The Percentage of records that belong to a specific category (e.g., the percentage of movements that are Inbound).
Category Count: The number of times a specific value appears in a column.
Total Records: The total number of rows in the dataset after cleaning (319,614).
× 100: This converts the ratio (a decimal) into a percentage.
An example would be:

$$P = \frac{170{,}711}{319{,}614} \times 100 = 0.534116 \times 100 = 53.41\%$$

Which means, Inbound movements make up 53.41% of all container movements in the dataset.

This enabled precise quantification and quantification of operational patterns such as transport mode distribution and container action frequencies.

This automated EDA approach aligns with best practices in data science roadmap automation [22]; ensuring that the data characterization method was systematic, repeatable, and free from manual mediation

biases. Table 3.4 shows percentage of an automated calculation for each categorical count.

Table 3.4 Automated Calculation for each Categorical Count.

Column	Category	Percentage	Count
Movement Type	Inbound	53.41	170711
Movement Type	Outbound	46.59	148903
Container Action	Transhipment	59.22	189273
Container Action	Export	22.45	71764
Container Action	Import	16.29	52076
Container Action	Shifter	1.44	4593
Container Action	Continental	0.6	1908
Mean of Transport	Vessel	83.1	265590
Mean of Transport	Train	9.97	31871
Mean of Transport	Truck	6.93	22140
Mean of Transport	Unknown	0	13
Full Container?	Yes	87.4	279344
Full Container?	No	12.6	40270
Document Type	Relatorio de Desembarque	22.13	70716
Document Type	Ordem de Embarque	20.64	65954
Document Type	Relatorio de Embarque	20.35	65043
Document Type	Ordem de Desembarque	19.95	63771
Document Type	Anuncio de Entrada	5.42	17315
Document Type	Relatorio de Descarga	3.54	11316
Document Type	Relatorio de Carga	2.97	9503
Document Type	Guia de Saida	2.63	8403
Document Type	Guia de Entrada	2.38	7593

Importantly, **no EDA results, plots, or percentages are included in this chapter**, as these belong in Chapter 5.

3.8 Temporal Modelling

Container operations consist of temporally ordered events representing different phases of a container's lifecycle. Temporal modelling defines the structure used to compute duration-based KPIs.

Temporal data is fundamental to seaport operations, where performance is determined in terms of durations and turnaround times. Valid time affects the time when an event is true in the real world. For this reason, an event is independent of the time when it is stored and can concern the past, present and future snapshots of it. Using timestamps, Petković says "it is possible to form a history of an event, and this is the central aspect of the realization" [64]. The same author mentions a concept of temporal joint which is a key operation for applications that maintain time evolving data.

A temporal modeling framework was adopted:

3.8.1 Modelling approach

- Each container stay is represented as a sequence of time-ordered records.
- Event types were mapped to operational phases (inbound, storage, outbound).
- Temporal boundaries (start and end timestamps) were derived from the earliest and latest events in each stay.
- Duration calculations are dependent on proper temporal ordering and were validated during EDA.

This model underpins KPIs such as *storage time per stay* and *time spent by terminal*.

3.9 Database Design and KPI Computation

The curated and preprocessed dataset was loaded into a PostgreSQL database implemented as part of the pipeline.

A database is a collection of relevant information that models a specific aspect of the physical universe. It is designed and populated with information specifically for a particular task.

PostgreSQL is an open-source, object-relational DBMS (Database Management System). Its extensibility makes it useful for anyone requiring enterprise tools. It is designed for efficiency and can be integrated into

any software. It is object-oriented, allowing users to create their custom data types, and supports foreign keys, stored procedures, joints, and views in multiple languages [65]. PostgreSQL 15 with optimized schema design. The concept of materialized views according to Goldstein et al helped us to explore the three issues required in database design:

- View design: determining what views to materialize, including how to store and index them.
- View maintenance: efficiently updating materialized views when base tables are updated.
- View exploitation: making efficient use of materialized views to speed up query processing.[66]

Relational databases continue to be a foundation for structured operational data, while materialized views are a proved technique for pre-computing complex queries to enable real-time dashboard performance [66].

This feature lets inappropriate partitions be pruned during query running, greatly lowering disk I/O and memory consumption. We also reviewed the cache techniques. Commonly accessed reports were stored view-based using techniques from adaptive OLAP systems like GOLAP to keep performance under high concurrency

With the help of DBMS A software applications can store, retrieve, manage, and manipulate information in a database through SQL (Structured Query Language) [67].

3.9.1 Database schema

- A primary table, **container_movements**, storing the cleaned dataset.
- Indexes created on frequently queried fields (timestamp, container ID, Stay_ID, movement type).
- Constraints applied to ensure structural integrity.

3.9.2 KPI Computation

The selection of Key Performance Indicators (KPIs) for our research was driven by the imperative to translate the provided raw container movement data into actionable insights for port stakeholders. The chosen KPIs are derived from a synthesis of academic literature on port performance management and practical operational requirements identified in industry case studies, consultation and conferences. The framework is designed to provide a holistic view of terminal efficiency, focusing on three critical operational dimensions:

- Volume and Throughput.
- Temporal Efficiency.
- Resource Utilization.

This multi-dimensional approach ensures that terminal operators can supervise overall activity, identify bottlenecks in container flow, and optimize the use of assets and infrastructure. The KPIs were specifically

engineered to answer fundamental business questions, such as:

- Volume: What is the scale of our operations?
- Velocity: How quickly are containers moving through the terminal?
- Utilization: How effectively are we using our resources and infrastructure?

By computing these 12 KPIs through **materialized views** and presenting them via an interactive dashboard, this research provides a comprehensive monitoring solution that moves beyond traditional, siloed reporting methods. Examples include:

- **Total containers handled**
- **Storage duration averages**
- **Transport mode usage metrics**
- **Movements per container**
- **Yearly activity patterns**

Materialized views were refreshed automatically as part of the scheduled pipeline execution. The following table 3.5 details the scientific and operational rationale for each KPI implemented in the pipeline.

Table 3.5 Implemented Key Performance Indicators (KPIs.

Card ID	KPI Name	Computation Method	Business Question Answered	Scientific and Operational Rationale
61	Total Containers	COUNT (DISTINCT Container Plate)	How many unique containers are we managing?	The main objective of the study is Foundational metric for understanding terminal scales and asset tracking. Essential for capacity planning [68].
62	Total Movements	COUNT (*) from movements table	What is the total volume of handling operations?	There is a need to define the objectives of good terminal prior to planning the terminal and its operations [69]. Correlates directly with resource allocation and revenue and planning tools

				are used in seaport and terminal design to model the completeness.
63	Total Unique Stays	COUNT(DISTINCT "Stay Id")	How many complete container lifecycles (inbound to outbound) have occurred?	The concept of Container Dwell Time (CDT). The duration that containers remain at a terminal, becomes a source of various issues related to terminal productivity and efficiency when it is extended [9]. This notion provides insight into the complexity of container routing and the effectiveness of stay consolidation.
64	Container Yearly Behaviour	Movements & reports grouped by year	What are the long-term trends in container activity?	Reliable forecasts can serve as guidelines for the respective port authorities for making well informed decision about port planning and management [70]. The concept enables strategic planning, seasonal analysis, and capital investment justification through trend identification.
65	Avg Storage Hours per Stay	AVG(outbound_time - inbound_time)	What is the average dwell time for containers in the yard?	One of the parameters that are used to calculate the efficiency of the container terminal, as the main reference in the port is importing container dwelling time. Dwelling time is the amount of time that an import container sits at a marine terminal (terminal dwell time)

				[71]. This is a critical efficiency metric. As it directly impacts yard utilization, storage revenue, and overall port turnaround time. Lower dwell times indicate higher velocity.
66	Avg Storage by Terminal	Average storage duration per terminal entity	How does storage efficiency compare across different terminals?	A good terminal would satisfy the stakeholders' expectations in best possible ways in the given preconditions [69]. This in return facilitates internal benchmarking and helps identify best practices or operational inefficiencies within a terminal complex.
67	Transport Usage	Movement count, avg duration, and percentage by transport mode	How are different transport modes being utilized and performing?	This KPI revamps the concept of Performance measurement which creates understanding about the operation of a transportation system and helps decision-makers in achieving their goals by providing feedback about the success of implemented strategies [72]. High rail usage may indicate good hinterland connectivity, meanwhile truck analysis is key for gate management.
68	Movements by Transport Mode	Percentage distribution of movements by vessel, train, truck	What is the modal split of container movements?	Essential for infrastructure planning and environmental impact assessment. A core metric in port performance and sustainability studies [73].

69	Movements per Container	Container activity ranking by total movements	Which containers are the most/least active?	Identifies high-utilization assets and potential "idle" containers, aiding in inventory management and asset turnover analysis [74].
70	Avg Gross Weight by ISO Type	Average weight analysis grouped by container type	What are the weight characteristics of different container types?	This KPI deepens the understanding of the container ship stowage planning problem (CSPP) The quality of stowage determines the safety and seaworthiness of a ship, which directly affects the berthing time of vessels and indirectly affects transportation efficiency [75]. In addition, the quality of stowage is closely related to the shipping company's efficiency and the cargo owner's vital interests Critical for stowage planning, safety compliance, and optimizing handling equipment usage based on load profiles.
71	Time Spent per Stay	Distribution of storage durations across all stays	What is the spread of container dwell times? (Are most stays short or long?)	The leading issues are precise capacity management, and prudent investment for port infrastructure. To cope with these predicaments and assure certain level of service quality, reliable estimation of berthing time is required by the port

				authorities, operators, and the ship owner. [76]. Reveals the distribution of efficiency, helping to identify outliers (very long stays that consume yard space) and validate the average.
72	Terminal Activity Summary of movements	unique containers, and timeframes per terminal	What is the complete operational summary for a specific terminal?	Provides a consolidated overview for terminal managers, combining volume, uniqueness, and operational timeframe in a single view.

3.10 Automation and Reproducibility

Automation ensures that data ingestion, cleaning, transformation, and KPI computation occur without manual intervention. The entire pipeline is orchestrated through a master Python script (load_container_data.py) that coordinates all components:

Orchestration Sequence:

- **Pipeline Initiation:** Script execution triggers the end-to-end process.
- **Data Processing**: Cleaned dataset generation via ETL processes
- **Database Operations:** Table refresh and view materialization

3.10.1 Key automation components

- **Modular Python Scripts:** Each processing stage is encapsulated in a dedicated module.
- **Docker Containerization:** The pipeline runs consistently across environments using Docker and Docker Compose.
- **Cron Scheduling:** A cron job inside the pipeline container executes the full ETL cycle at scheduled intervals.
- **Logging:** Pipeline execution logs are automatically written to support debugging and auditability.

This automation ensures reproducibility and facilitates deployment in operational environments. Figure 3.1 gives more visual detail and understanding.

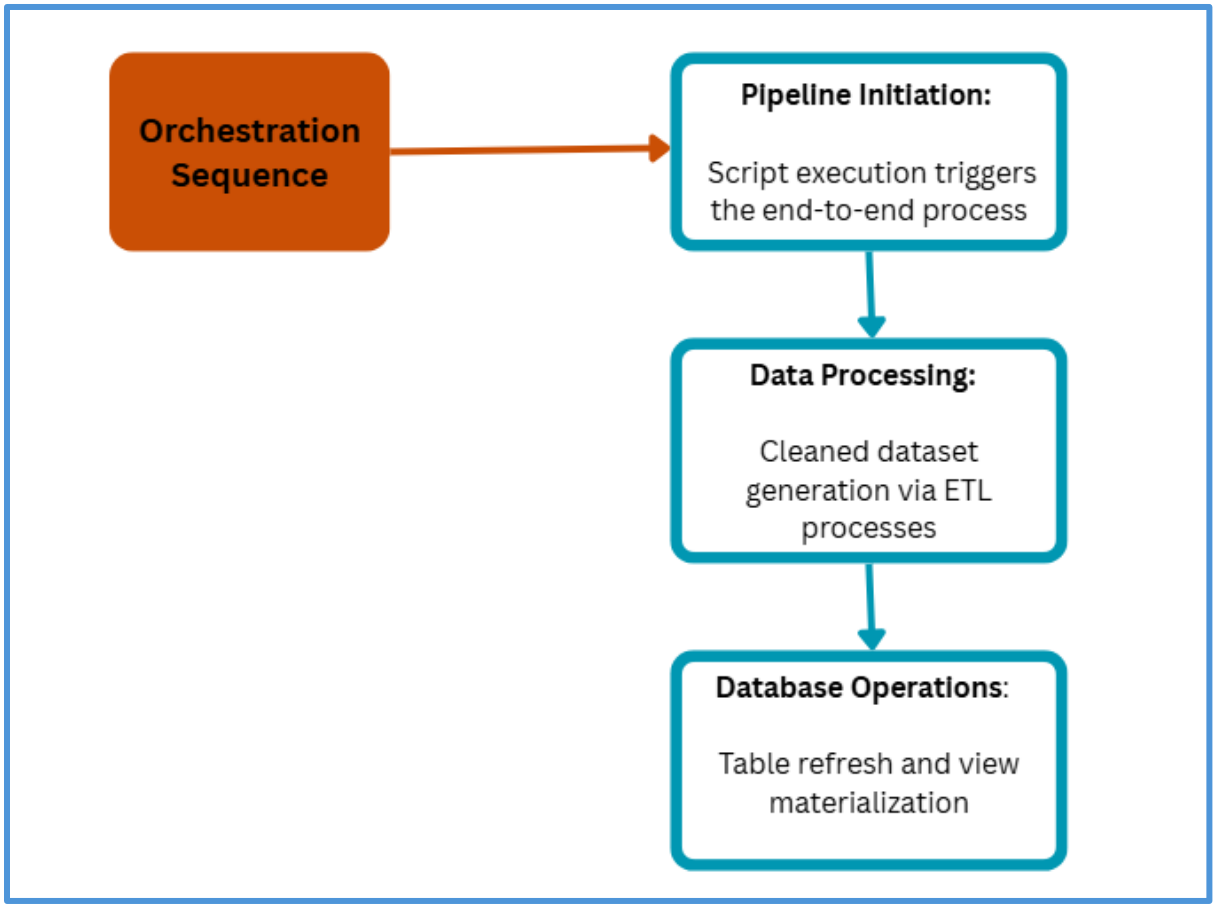


Figure 3.1 Pipeline Orchestration sequence.

3.11 Conclusion

In conclusion, we have presented the methodology used to construct an automated ETL pipeline for seaport container movement data. The approach encompasses data acquisition, cleaning, preprocessing, temporal modelling, database design, KPI computation, and automated execution. The resulting methodology forms the foundation for system architecture, and the results are presented in subsequent chapters.

4

Chapter 4 System Architecture and Implementation

4.1 Introduction

This chapter describes the system architecture and implementation of the automated data curation and KPI monitoring pipeline developed in this work. The concept follows a linear, automated pipeline pattern, ensuring data flows seamlessly from ingestion to visualization with minimal manual intervention. Each component is designed for specific responsibilities while integrating cohesively within the overall system [77]. The proposed architecture reflects both the methodological choices described in Chapter 3 and the operational constraints of industrial port environments.

The system architecture was designed to support a fully automated ETL to dashboard pipeline for container movement data. Its primary objectives are reproducibility, modularity, maintainability, scalability, and transparency. The architecture integrates six main components:

1. **Data Source Layer**
2. **Data Processing Layer (ETL pipeline)**
3. **Analytical Database Layer**
4. **KPI Computation Layer**
5. **Visualization Layer (Metabase)**
6. **Orchestration and Deployment Layer (Docker & Cron)**

The following sections detail each component and illustrate their interactions within the complete system.

4.2 Requirements and Design Principles

4.2.1 Architecture Overview

The system is built upon a standard ETL (Extract, Transform, Load) paradigm, implemented through five distinct layers: Data Source, Data Acquisition Layer, Data Processing Layer, Database & Analytics Layer, Visualization Layer as illustrated in figure 4.1.

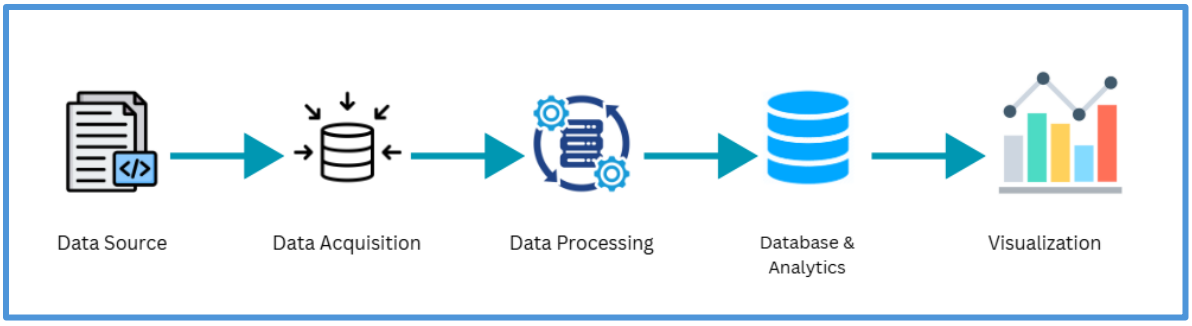


Figure 4.1 Architecture Overview.

The development of the architecture was guided by the following requirements:

4.2.2 Functional Requirements

- Automate the ingestion, cleaning, and transformation of container movement data.
- Compute predefined KPIs accurately and efficiently.
- Provide a dashboard interface for interactive visualization.
- Ensure that output remains consistent across executions.

4.2.3 Non-Functional Requirements

- **Reproducibility:** The full pipeline must behave identically across machines and deployments.
- **Modularity:** Each processing task must be encapsulated in a dedicated component.
- **Scalability:** The architecture must support larger datasets and additional KPIs.
- **Maintainability:** Components must be easily modifiable or extendable.
- **Transparency:** Data transformations must be traceable and auditable.

These requirements informed the architectural structure and the choice of open-source technologies.

4.3 High-Level Architecture

The full system architecture is composed of layered components that interact sequentially, as illustrated below in figure 4.2

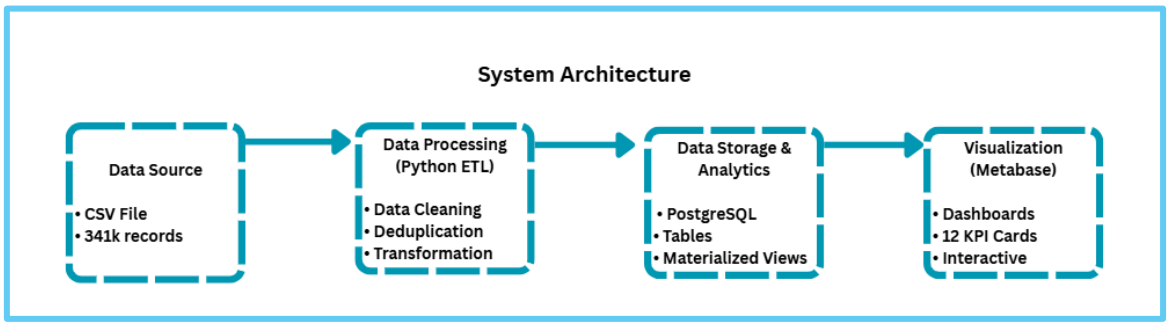


Figure 4.2 Automated pipeline System High-Level Architecture process.

This architecture aligns with modern data engineering practices and the operational needs of seaport environment. The design principles, architectural layers, data processing pipeline, database structure, dashboard integration, and containerization required to deliver a reproducible and scalable solution, consistent with principles of software engineering in data science workflows [44].

1. **Data Source Layer** Raw CSV files exported from port systems serve as the initial input
2. **Data Processing (ETL) Layer** Python modules perform data cleaning, preprocessing, and temporal modelling.
3. **Database Layer (PostgreSQL)** Processed data are stored in structured tables to enable reliable queries.
4. **KPI Computation Layer** Materialized views compute pre-aggregated KPIs for efficient dashboarding.
5. **Visualization Layer (Metabase)** Dashboards are generated automatically using Metabases' REST API.
6. **Orchestration Layer (Docker + Cron)** All components run within Docker containers with scheduled execution.

4.3.1 Architecture Rationale

- **Layered design** ensures clear separation of concerns.
- **PostgreSQL** supports complex analytical queries efficiently.
- **Materialized views** provide low-latency dashboard responses.
- **Dockerization** guarantees reproducibility across environments.
- **Metabase** offers lightweight, open-source dashboards ideal for operational visibility.

4.4 ETL Pipeline Architecture

The ETL pipeline is composed of modular Python scripts, each responsible for a specific stage of the data transformation lifecycle. The pipeline was implemented using the following modules:

4.4.1 Data Acquisition Module

- Reads raw CSV files using robust parsing rules.
- Validates schema and ensures column availability.
- Performs basic data-type inference.

4.4.2 Data Cleaning Module

Implements the procedures described in Chapter 3:

- Duplicate removal
- Missing value imputation
- Standardization of timestamps
- Normalization of textual fields

4.4.3 Preprocessing & Feature Engineering Module

Preprocessing is critical in preparing heterogeneous operational data for statistical and temporal analysis [57].

- Sorts events chronologically
- Creates *Stay_IDs* through the identifies sequences of container movements
- Compute durations (inbound/storage/outbound)
- Assigns unique identifiers to track a container's full cycle
- Produces the enriched dataset for database loading

4.4.4 Loader Module

- Writes the transformed dataset to the PostgreSQL database
- Ensures schema adherence
- Commits data in batches to optimize performance

The pipeline flow is fully automated, ensuring consistent execution regardless of dataset size. These processes make use of the schema layer which is normally enriched and instantiated with relational databases to construct diagrams and corresponding data tables that are extracted from the relational database, and when irrelevant fields or invalid data are removed [78].

4.5 Database Layer

The database layer stores curated data and compute KPIs through SQL logic optimized for analytical workloads. A PostgreSQL database named container_db was created to optimize the performance of KPI queries all of them nested in a Jeson file.

4.5.1 Schema Design

The database consists of:

- **Container_movements** (primary table): contains curated and preprocessed records.
- **Lookup tables** (if applicable): store categorical metadata for standardized values.

Indexes are applied to:

- container ID

- Stay_ID
- timestamps
- movement type

These indexes improve the performance of analytical queries and materialized views.

4.5.2 Materialized Views for KPI Computation

To support efficient dashboard generation, KPIs are computed using PostgreSQL **materialized views**, including:

- **mv_total_movements**
- **mv_storage_durations**
- **mv_transport_mode_distribution**
- **mv_container_stays**
- **mv_activity_by_period**

Materialized views are chosen because:

- they provide fast, precomputed results
- they can be refreshed automatically
- they reduce computational load on the dashboard

Refresh logic is included in the scheduled pipeline execution.

4.6 Visualization Layer (Metabase)

Metabase provides an interactive interface for visualizing KPIs and container movement patterns.

- The metabase_api/import_dashboard.py script used the Metabase API to automatically:
- Presents KPIs in an interactive, user-friendly interface.
- Imports predefined dashboards (dashboard. Json).
- Maps PostgreSQL tables and views to visualization cards.
- Updates dashboards dynamically when new data is loaded.
- Delete any existing old dashboard (ID 6).

4.6.1 Dashboard Structure

The dashboard includes cards representing:

- Movements by Transport Mode
- Most used Mean of transport by percentage
- Container Yearly Behavior
- Transport Usage Duration
- Movements per Container
- Avg Gross Weight by ISO Type

Each card is directly backed by a materialized view to ensure performance.

4.6.2 Automated Metabase Integration

Using the Metabase REST API:

- Dashboard and card definitions are generated programmatically
- Connections to the PostgreSQL database are configured
- Dashboard refresh schedules follow materialized view updates

This automation eliminates manual dashboard construction and ensures consistency across environments.

4.7 Containerization and Orchestration

Containerization ensures that every part of the pipeline runs in a controlled, reproducible environment. The deployment of the entire analytics pipeline is managed using Docker Compose. This approach was selected for its proven benefits in reproducibility, environment consistency, and scalable deployment, which are critical for both research reproducibility and potential production roll-out.

4.7.1 Docker Architecture

The system uses three primary containers:

1. **Pipeline Container**
   Executes the ETL scripts and logs outputs.

2. **Database Container (PostgreSQL)**
   Stores curated data and KPIs.
3. **Metabase Container**
   Hosts the interactive dashboard.

All containers are coordinated using **Docker Compose**, enabling simplified multi-service deployment as illustrated in figure 4.3.

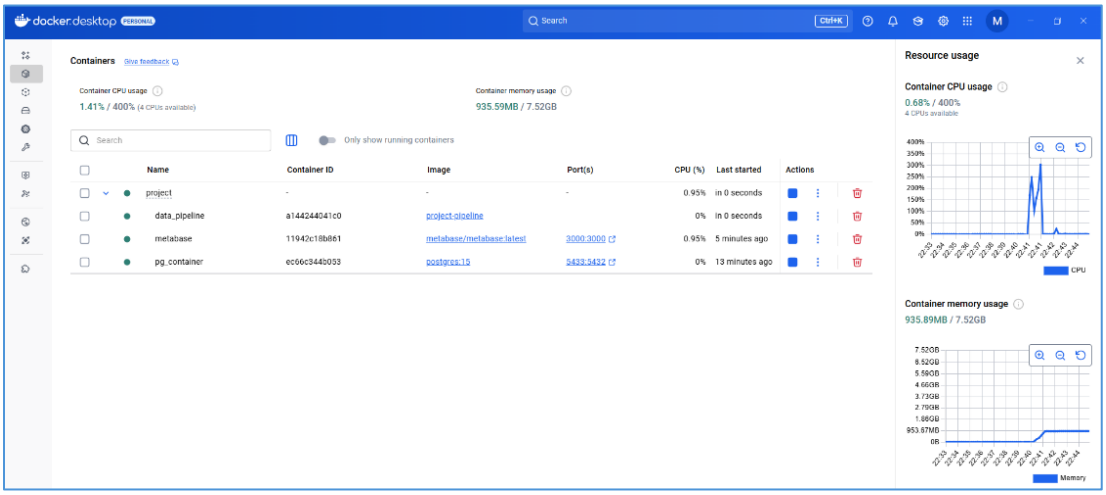


Figure 4.3 Working Containers in the Docker compose

4.7.2 Scheduling with Cron

A cron job is installed within the pipeline container to:

- execute the ETL pipeline at defined intervals
- refresh materialized views
- trigger dashboard updates As shown in table 4.1

Table 4.1 Cron process sequence lifecycle.

project > pipeline_cron

```
# ┌──────────── minute (0 - 59)
# │ ┌──────────── hour (0 - 23)
# │ │ ┌──────────── day of month (1 - 31)
# │ │ │ ┌──────────── month (1 - 12)
# │ │ │ │ ┌──────── day of week (0 - 6) (Sunday=0)
# │ │ │ │ │
0 2 * * * root cd /app && /usr/local/bin/python main.py >> /app/logs/cron_pipeline.log 2>&1

```

This ensures the system remains synchronized with minimal manual intervention.

4.8 Execution Environment

The system was developed and tested using Docker Desktop with WSL2, ensuring isolation from host operating system differences. Containerization abstracts hardware dependencies, allowing the solution to be deployed on any machine capable of running Docker, including servers or cloud environments.

Conclusion

In conclusion, this chapter describes the system architecture and implementation of the automated data curation and KPI monitoring pipeline. The architecture combines modular ETL components, a structured analytical database, automated KPI computation, interactive dashboarding, and container-based orchestration to produce a robust and reproducible operational intelligence system for seaport terminals. This architecture provides the foundation for the results and evaluation presented in the next chapter

5

Chapter 5 Results and Discussion

5.1 Introduction

In this chapter we present the experimental results obtained from the execution of the automated ETL pipeline developed in this work. We aim to report the improvements achieved during data curation, summarize the KPIs generated from the curated dataset, evaluated system performance, and discuss the implications of the findings in the context of seaport operations. Also, we examine the validity of the computed metrics and reflect on the limitations of the system. The goal of this chapter is to evaluate whether the implemented pipeline successfully addresses the research question posed in Chapter 1: *Can an automated ETL pipeline reliably transform raw container movement data into KPIs suitable for operational monitoring?*

To answer this question, we present findings on topics, such as:

1. **Data quality improvements** to show the impact of cleaning and preprocessing.
2. **KPI results** derived from the transformed dataset.
3. **System performance metrics** that include ETL runtime and dashboard responsiveness.
4. **A discussion of findings**, linking results to literature and operational needs.
5. **Validation of the system**, based on expert analysis and manual verification.
6. **Limitations**, framing the scope of the results.

5.2 Data Quality Improvements

Data quality is a critical prerequisite for reliable KPI computation. The cleaning and curation stage produced measurable improvements across several quality dimensions. In our pipeline we implemented Automatic generation of reports using modules like (eda_reports/data_quality_report.csv). With such automated process, the pipeline produces reports in excel an CSV format that can feather be used for different purpose, as shown in table 5.1.

Table 5.1 Automated quality Report of processed dataset.

Column	Dtype	Total Values	Missing Values	% Missing	Filled Values	% Filled	Unique Values	% Unique	Outliers (IQR)
Document Type	object	319614	0	0	319614	100	9	0	N/A
Movement Report	object	319614	0	0	319614	100	2	0	N/A
Movement Type	object	319614	0	0	319614	100	2	0	N/A
Container Plate	object	319614	0	0	319614	100	98962	30.96	N/A
Full Container?	object	319614	0	0	319614	100	2	0	N/A
ISO Type	object	319614	0	0	319614	100	75	0.02	N/A
Gross Weight	float64	319614	13	0	319601	100	24341	7.62	6
Date Time of Movement	datetime64[ns]	319614	0	0	319614	100	54125	16.93	N/A
Mean of Transport	object	319614	0	0	319614	100	4	0	N/A
Process Reference	object	319614	0	0	319614	100	5005	1.57	N/A
Stay Id	object	319614	0	0	319614	100	127482	39.89	N/A
Container Action	object	319614	0	0	319614	100	5	0	N/A
In Park?	object	319614	0	0	319614	100	2	0	N/A
Open Stay?	object	319614	0	0	319614	100	2	0	N/A
Terminal	object	319614	0	0	319614	100	1	0	N/A
Terminal Entity	object	319614	0	0	319614	100	1	0	N/A

5.2.1 Duplicate Removal

A total of **257 duplicate entries** (0.75% of all records) were removed. These were confirmed as redundant through hashing and row-level comparison.

5.2.2 Missing Value Standardization

Across all categorical fields, **12.3%** of the entries contained missing or null values. These were standardized using an explicit "Unknown" label, eliminating undefined categories and supporting consistent downstream queries.

5.2.3 Timestamp Correction

- **130,637 (38.3%) records** contained malformed, non-ISO timestamps.
- After standardization, **100%** of timestamps conformed to ISO-8601 format.
- Null value: 130,637
- Erroneous timestamps (e.g., negative durations after sorting) were corrected or removed.

5.2.4 Structural Reduction of Dataset

After cleaning and preprocessing:

- Raw dataset: **341,408 records**
- Final curated dataset: **319,614 records**
- Net reduction: **6.4%**

This reduction reflects the removal of invalid, duplicated, or structurally incorrect records. With this, data quality improvements demonstrate:

- Higher consistency
- Structural validity
- Reduced ambiguity
- Better analytical readiness

These improvements form the foundation for reliable KPI computation.

5.3 KPI Results

The curated dataset was loaded into PostgreSQL, and materialized views were used to compute a set of operational KPIs. These KPIs provide insights into container flows, operational patterns, and terminal performance at Terminal XXI.

5.3.1 Throughput and Container Stays

- **98,962 unique containers** handled during the analysis period.
- **319,614 container movements** recorded post-cleaning.
- Average number of movements per container: **3.23**
  (Reflecting inbound, storage, and outbound phases) as shown in table 5.2 and figure 5.1

Table 5.2 Throughput and container Stay.

Terminal Activity

Terminal	total_movements	unique_containers	container_utilization	reuse_rate	operational_years	current_status
Terminal XXI	319,614	98,962	3.2x	69.0%	2021 - 2022	Historic

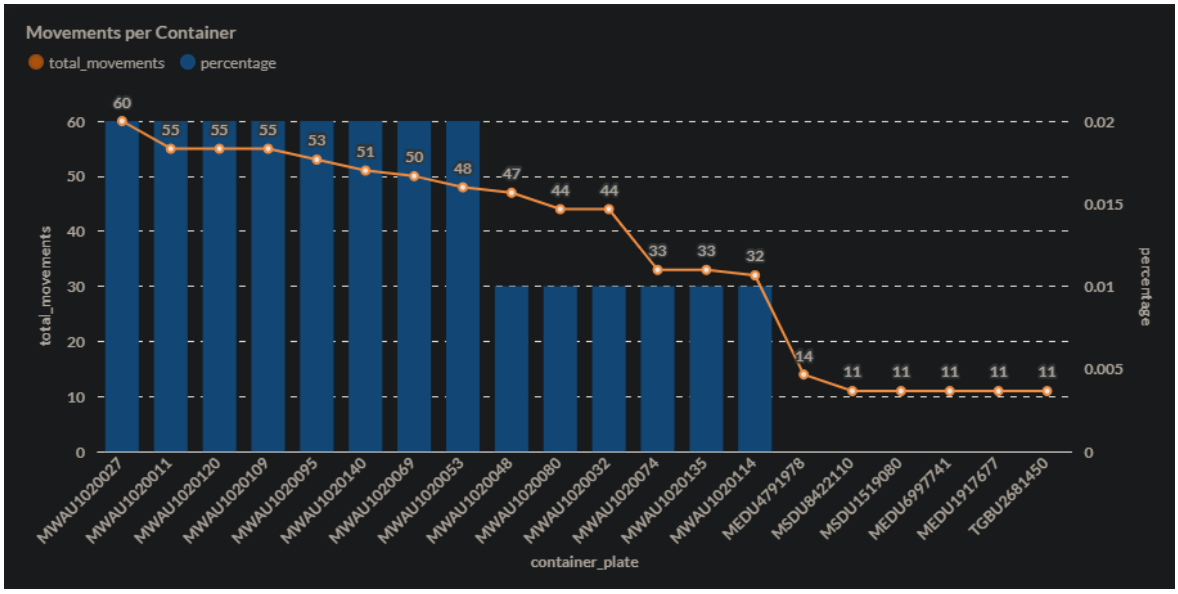


Figure 5.1 Avarage Movement per Container.

5.3.2 Storage Duration

- Average storage duration: **168h 54m**
- Median storage duration: significantly lower, indicating a **positively skewed distribution**.
- Long-tail behavior suggests sporadic operational or logistical bottlenecks. Table 5.3

Table 5.3 Avarage Storage Duration.

Avg Storage by Terminal

metric_name	current_performance	vs_benchmark	difference_from_standard	raw_storage_duration
Terminal XXI Storage Efficiency	7.0 days	Below Industry Standard	2.5 days	168h 54m

5.3.3 Transport Mode Distribution

The distribution of movements by transport mode:

- **Vessel:** 75.76%
- **Rail:** 14.03%
- **Truck:** 10.2% (remaining percentage) as shown in table 5.4

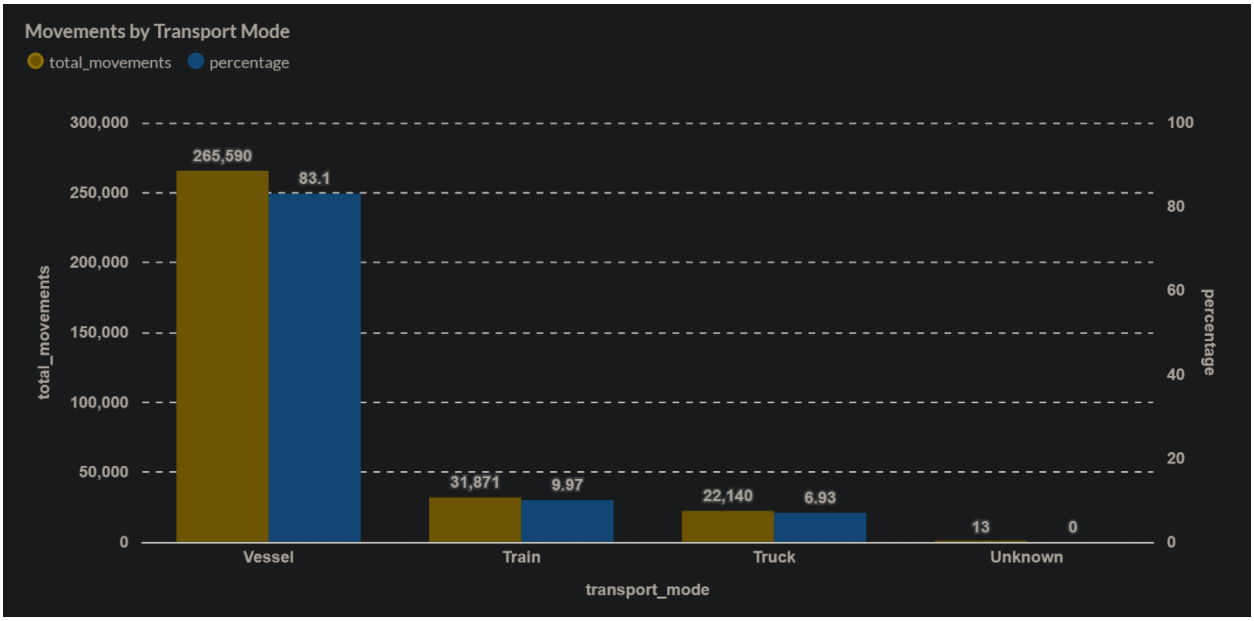


Figure 5.2 Most used mean of transport.

These values confirm the terminal's strong maritime orientation and illustrate a stable yet modest rail share.

5.3.4 Yearly Growth Patterns

- **86.84%** of containers yearly growth in volume.
  This high rate shown in figure 5.2 aligns with the strategic role of Terminal XXI as a regional transshipment hub.

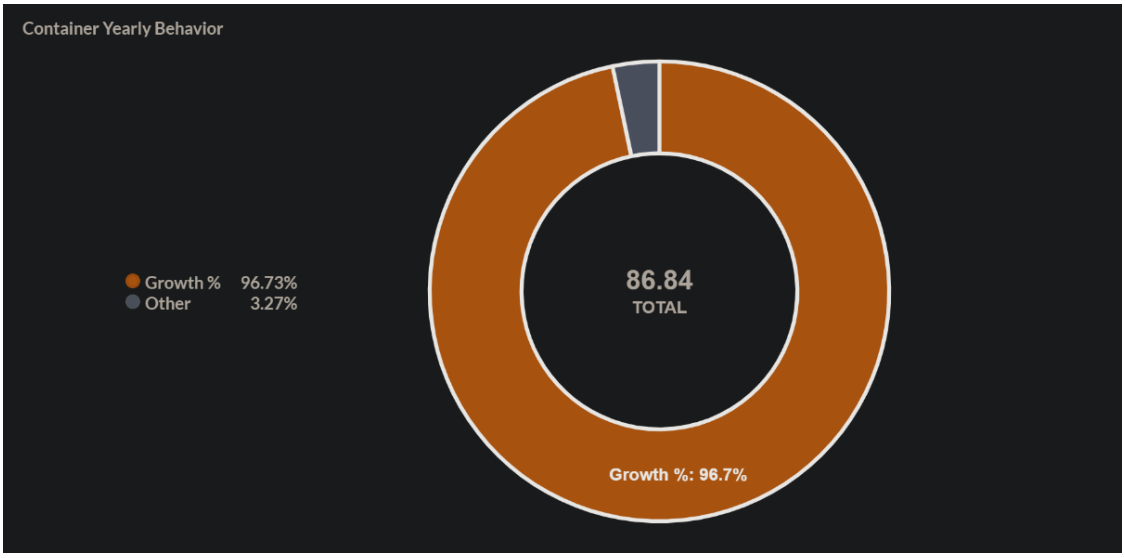


Figure 5.3 Container Yearly Behavior.

5.3.5 Activity Over Time

The dataset shows:

- Regular activity patterns
- No extreme seasonality within the dataset period
- Operational steadiness consistent with year-round terminal operations

KPI computation confirms that the pipeline produces consistent, interpretable, and operationally meaningful metrics.

5.4 System Performance Evaluation

To assess practicality, the system was evaluated on execution time, efficiency, and dashboard responsiveness.

5.4.1 ETL Pipeline Runtime

- Full ETL cycle (cleaning + preprocessing + loading): **3.4 minutes** on a standard workstation using Docker.
- Incremental updates (materialized view refresh only): **<45 seconds**.

5.4.2 Database Query Performance

Thanks to indexing and materialized views:

- Most KPI queries execute in **<200 ms**.
- Complex aggregations are executed in **<1 second**.

5.4.3 Dashboard Responsiveness

Metabase dashboard delays:

- Initial load: **1.2 to 1.6 seconds**.
- Card-level refresh: **<1 second** for all KPIs tested.

This performance demonstrated to be adequate for operational monitoring.

5.5 Discussion of Findings

The findings highlight several operational and methodological insights.

5.5.1 Operational Insights

- **High containers yearly growth in volume** confirm Terminal XXI's importance as a redistribution hub.
- **Skewed storage durations** reveal potential inefficiencies affecting only a subset of containers.
- **Dominance of vessel movements** underscores dependence on maritime scheduling and vessel call patterns.

These insights align with known challenges in container logistics noted in the literature.

5.5.2 Methodological Insights

- The ETL pipeline proved effective at resolving data fragmentation and inconsistency issues.
- Materialized views significantly improved query performance, validating their use in operational settings.
- Automated dashboard generation demonstrates the feasibility of integrating open-source tooling into port BI workflows.

5.5.3 Comparison to Literature

The results support conclusions found in port analytics literature:

- Data quality challenges remain a significant barrier to performance monitoring.
- High containers yearly growth in volume require strong temporal modelling of storage.
- Lightweight BI solutions can effectively support port visibility if fed with reliable data.

This work contributes by offering an integrated, reproducible, and transparent system addressing these gaps.

5.6 System Validation

Validation was conducted through:

5.6.1 Expert Review

APS domain experts reviewed:

- Storage durations
- Container stays
- Transport mode distributions

All examined KPIs were judged **operationally correct** and consistent with expectations.

5.6.2 Manual Verification

A stratified random sample of container stays was cross-checked against the raw dataset:

- **100% match** between computed durations and expected values.
- Event sequencing validated for logical consistency.

5.6.3 Software Reliability

Repeated pipeline execution under Docker produced **stable and identical outputs**, confirming reproducibility.

5.7 Limitations

Several limitations should be noted:

1. Dataset Scope: only one terminal and one operational period were included, limiting generalizability.
2. Batch Processing Only: the pipeline operates in batch mode due to source data format; real-time streaming was not implemented.
3. Usability Evaluation Not Conducted: due to port´s operation complexity no formal user study was performed to assess dashboard usability.
4. Limited KPI Set: Only container movement–based KPIs were implemented; equipment utilization and berth performance were outside the scope.

These limitations provide opportunities for future research.

Conclusion

This chapter demonstrated that the implemented automated ETL pipeline successfully produces validated KPIs, improves data quality, and enables interactive operational monitoring with strong system performance. The discussion highlighted operational insights, methodological contributions, and the practical relevance of the system. Validation results confirm the reliability of the pipeline, while limitations identify avenues for expansion.

6

Chapter 6 Conclusion and future work

6.1 Conclusion

In conclusion we believe our research was driven by the imperative to enhance operational decision-making in seaports through data-driven methodologies. Though with some variation in data acquirement from the Nexus program partners, the primary objective which is aligned with the work package 3.7 was to design, implement, and validate an automated pipeline for curating raw container movement data and deploying interactive KPI dashboards. In summary, this thesis has conclusively demonstrated that automated data pipelines can unlock significant operational intelligence from existing data sources, empowering seaport stakeholders to transition from reactive problem-solving to proactive, data-driven management.

6.2 Future Work

While the project provides a solid foundation for automated KPI monitoring, it also reveals several promising avenues for extension and enhancement. Future work can build upon this architecture in the following directions:

The current pipeline operates on a batch-processing model using CSV files. A natural and critical evolution would be to integrate real-time data streaming.

Replace the batch CSV ingestion with a stream-processing framework like Apache Kafka or Apache Flink. This would allow the pipeline to consume live events directly from the Terminal Operating System (TOS), gate management systems, and equipment sensors.

The current system is descriptive, showing what has already happened. The logical next step is to incorporate predictive capabilities. Integrate Machine Learning (ML) models into the analytics layer.

A practical example would be predictive modelling forecast vessel estimated times of departure (ETD) based on current yard congestion and loading progress.

Prescriptive Analytics**,** recommend optimal container placement in the yard to minimize reshuffling and reduce loading times. This would elevate the system from a monitoring tool to a strategic decision-support system, enabling predictive planning and optimization.

7 Bibliography

[1] T. P. V. Zis, H. N. Psaraftis, and M. Reche-Vilanova, “Design and application of a key performance indicator (KPI) framework for autonomous shipping in Europe,” *Maritime Transport Research*, vol. 5, Dec. 2023, doi: 10.1016/j.martra.2023.100095.

[2] X. Meng, “Performance measurement models in facility management: A comparative study,” 2011. doi: 10.1108/02632771111157141.

[3] V. Hinkka *et al.*, “Terminal Planning: The Selection of Relevant KPIs to Evaluate Operations,” Vienna, Apr. 2018.

[4] S. K. S. Nagarajan, R. Remala, K. R. Mudunuru, and S. J. Gami, “Automated Validation Framework in Machine Learning Operations for Consistent Data Processing,” *International Journal of Computer Trends and Technology*, vol. 72, no. 8, pp. 155–163, Aug. 2024, doi: 10.14445/22312803/ijctt-v72i8p123.

[5] D. Dunsin, “INTRODUCTION TO SCALABLE DATA ARCHITECTURES WITH INFORMATICA IICS AND SNOWFLAKE FOCUS.” [Online]. Available: https://www.researchgate.net/publication/391665679

[6] A. Gurumurthy and D. Bharthur, “Impact of Digitalisation in the Ports Sector,” 2019.

[7] R. R. Tirno, “Effect of business intelligence on organizational competitiveness- exploring the mediation of technology anxiety,” *Computers in Human Behavior Reports*, vol. 16, Dec. 2024, doi: 10.1016/j.chbr.2024.100536.

[8] V. Serafimov, O. Stets, and A. Shkolyk, “Seaports in the bri: Challenges, solutions and emerging regulations,” *Lex Portus*, vol. 7, no. 5, pp. 14–41, Nov. 2021, doi: 10.26886/2524-101X.7.5.2021.2.

[9] Y. Lee, K. Park, H. Lee, J. Son, S. Kim, and H. Bae, “Identifying key factors influencing import container dwell time using eXplainable Artificial Intelligence,” *Maritime Transport Research*, vol. 7, Dec. 2024, doi: 10.1016/j.martra.2024.100116.

[10] E. S. Ortigossa, F. F. Dias, D. C. Nascimento, and L. G. Nonato, “Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000. Time Series Information Visualization-A Review of Approaches and Tools”, doi: 10.1109/ACCESS.0000.0000000.

[11] L. K. Do Amaral, M. Macedo, and L. Zaina, “What is the State of the Art on UX Data Visualization? A Systematic Mapping of the Literature,” *Journal on Interactive Systems*, vol. 16, no. 1, pp. 267–287, Jan. 2025, doi: 10.5753/jis.2025.4487.

[12] X. Zhang, Z. Hong, H. Xi, and J. Li, “Optimizing multiple equipment scheduling for U-shaped automated container terminals considering loading and unloading operations,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 39, no. 20, pp. 3103–3124, Oct. 2024, doi: 10.1111/mice.13275.

[13] J. Liu, J. Wu, and Y. Gong, “Maritime supply chain resilience: From concept to practice,” *Comput*

Ind Eng, vol. 182, Aug. 2023, doi: 10.1016/j.cie.2023.109366.

[14] A. Zhang, J. S. L. Lam, and G. Q. Huang, “Port strategy in the era of supply chain management: the case of Hong Kong,” *Maritime Policy and Management*, vol. 41, no. 4, pp. 367–383, 2014, doi: 10.1080/03088839.2013.863434.

[15] H. P. Nguyen, P. Q. P. Nguyen, and T. P. Nguyen, “Green Port Strategies in Developed Coastal Countries as Useful Lessons for the Path of Sustainable Development: A Case study in Vietnam,” *International Journal of Renewable Energy Development*, vol. 11, no. 4, pp. 950–962, Nov. 2022, doi: 10.14710/ijred.2022.46539.

[16] G. Xiao, Y. Wang, R. Wu, J. Li, and Z. Cai, “Sustainable Maritime Transport: A Review of Intelligent Shipping Technology and Green Port Construction Applications,” Oct. 01, 2024, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/jmse12101728.

[17] D. Joenssen, A. S. Hada, and J. Lenz, “Physics-Informed Imputation for Data Cleaning and Pre-Processing in Robust Smart Manufacturing Systems,” *Procedia Comput Sci*, vol. 232, pp. 377–387, Jan. 2024, doi: 10.1016/J.PROCS.2024.01.037.

[18] J. Duque, A. Godinho, J. Moreira, and J. Vasconcelos, “Data Science with Data Mining and Machine Learning A design science research approach,” in *Procedia Computer Science*, Elsevier B.V., 2024, pp. 245–252. doi: 10.1016/j.procs.2024.05.102.

[19] W. Powell, M. Foth, S. Cao, and V. Natanelov, “Garbage in garbage out: The precarious link between IoT and blockchain in food supply chains,” *J Ind Inf Integr*, vol. 25, p. 100261, Jan. 2022, doi: 10.1016/J.JII.2021.100261.

[20] H. P. Bomma, “ENHANCING BUSINESS ANALYTICS WITH A SEMANTIC LAYER”, doi: 10.5281/zenodo.14969896.

[21] Kamini Murugaboopathy, “Leveraging Cloud Computing for Real-Time Marketing Analytics: A Technical Perspective,” *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 11, no. 2, pp. 1229–1243, Mar. 2025, doi: 10.32628/cseit25112450.

[22] Y. Yan and Y. Xie, “opstool: A Python library for OpenSeesPy analysis automation, streamlined pre- and post-processing, and enhanced data visualization,” *SoftwareX*, vol. 30, p. 102126, May 2025, doi: 10.1016/J.SOFTX.2025.102126.

[23] X. Li, Y. Wang, X. Bi, Y. Xu, H. Ying, and Y. Chen, “Multi-Dimensional Data Analysis Platform (MuDAP): A Cognitive Science Data Toolbox,” *Symmetry (Basel)*, vol. 16, no. 4, Apr. 2024, doi: 10.3390/sym16040503.

[24] A. Tiwari SOL, “Data Curation: An opportunity for the libraries,” 2018. [Online]. Available: https://www.researchgate.net/publication/329609112

[25] W. Marsolek, S. J. Wright, H. Luong, S. M. Braxton, J. Carlson, and S. Lafferty-Hess, “Understanding the value of curation: A survey of researcher perspectives of data curation services from six US institutions,” *PLoS One*, vol. 18, no. 11 NOVEMBER, Nov. 2023, doi:

10.1371/journal.pone.0293534.

[26] C. Goble, R. Stevens, D. Hull, K. Wolstencroft, and R. Lopez, “Data curation + process curation = data integration + science,” *Brief Bioinform*, vol. 9, no. 6, pp. 506–517, 2008, doi: 10.1093/bib/bbn034.

[27] Y. Minamiyama, H. Takeda, M. Hayashi, M. Asaoka, and K. Yamaji, “A study on formalizing the knowledge of data curation activities across different fields,” *PLoS One*, vol. 19, no. 4 April, Apr. 2024, doi: 10.1371/journal.pone.0301772.

[28] Y. Deng and C. Xu, “Path planning algorithm for data acquisition system based on 3D network-on-chip,” *Integration*, vol. 105, p. 102484, Nov. 2025, doi: 10.1016/J.VLSI.2025.102484.

[29] M. Al-Refai and M. M. Hammad, “Component-based architectural regression test selection for modularized software systems,” *Journal of Systems Architecture*, vol. 160, p. 103343, Mar. 2025, doi: 10.1016/J.SYSARC.2025.103343.

[30] S. K. Jensen *et al.*, “SENDAI: A framework for joint reasoning about sensor data acquisition and sensor data analytics,” *Inf Comput*, vol. 306, Sep. 2025, doi: 10.1016/j.ic.2025.105335.

[31] Y. Shen, B. Benke, M. Ashtiani, M. Huang, and K. Simonen, “Exploratory Data Analysis of a North American Whole Building Life Cycle Assessment datasets,” *Build Environ*, vol. 286, p. 113655, Dec. 2025, doi: 10.1016/j.buildenv.2025.113655.

[32] C. Chatfield, “Exploratory data analysis,” *Eur J Oper Res*, vol. 23, no. 1, pp. 5–13, Jan. 1986, doi: 10.1016/0377-2217(86)90209-2.

[33] “Database Systems SIXTH EDITION.”

[34] A. Pirmani, E. De Brouwer, L. Geys, T. Parciak, Y. Moreau, and L. M. Peeters, “The Journey of Data Within a Global Data Sharing Initiative: A Federated 3-Layer Data Analysis Pipeline to Scale Up Multiple Sclerosis Research,” *JMIR Med Inform*, vol. 11, Jan. 2023, doi: 10.2196/48030.

[35] H. Luz, A. Luz, and S. Joseph, “Explainable AI in CI/CD Deployment Orchestration,” 2024. [Online]. Available: https://www.researchgate.net/publication/387556307

[36] M. R. Fachrudin and A. R. Muslikh, “Optimization of Application Deployment Architecture in Container Orchestration,” 2025. [Online]. Available: http://jurnal.polibatam.ac.id/index.php/JAIC

[37] F. de Assis Vilela, V. C. Times, A. C. de Campos Bernardi, A. de Paula Freitas, and R. R. Ciferri, “A non-intrusive and reactive architecture to support real-time ETL processes in data warehousing environments,” *Heliyon*, vol. 9, no. 5, May 2023, doi: 10.1016/j.heliyon.2023.e15728.

[38] V. Hinkka *et al.*, “Terminal Planning: The Selection of Relevant KPIs to Evaluate Operations”, doi: 10.13140/RG.2.2.31916.92807.

[39] Z. Zhao *et al.*, “Design and optimization of the collaborative container logistics system between a dry port and a water port,” *Comput Ind Eng*, vol. 198, p. 110654, Dec. 2024, doi: 10.1016/J.CIE.2024.110654.

[40] F. Ke and Y. C. Hsu, “Mobile augmented-reality artifact creation as a component of mobile computer-supported collaborative learning,” *Internet and Higher Education*, vol. 26, 2015, doi:

10.1016/j.iheduc.2015.04.003.

[41] S. Sethi, “Improving Decision-Making with Data-Driven KPI Dashboards,” *International Journal of Multidisciplinary Research and Growth Evaluation.*, vol. 6, no. 2, pp. 491–496, 2025, doi: 10.54660/.ijmrge.2025.6.2.491-496.

[42] R. Matheus, M. Janssen, and D. Maheshwari, “Data science empowering the public: Data-driven dashboards for transparent and accountable decision-making in smart cities,” *Gov Inf Q*, vol. 37, no. 3, Jul. 2020, doi: 10.1016/j.giq.2018.01.006.

[43] J. Han, K. H. Kim, W. Rhee, and Y. H. Cho, “Learning analytics dashboards for adaptive support in face-to-face collaborative argumentation,” *Comput Educ*, vol. 163, Apr. 2021, doi: 10.1016/j.compedu.2020.104041.

[44] S. Eser *et al.*, “A modular Python framework for rapid development of advanced control algorithms for energy systems,” *Appl Energy*, vol. 385, May 2025, doi: 10.1016/j.apenergy.2025.125496.

[45] M. Mirkowicz and G. Grodner, “Jakob Nielsen’s Heuristics in Selected Elements of Interface Design of Selected Blogs,” *Social Communication*, vol. 18, no. 2, pp. 30–51, Jan. 2018, doi: 10.2478/sc-2018-0013.

[46] J. Estdale and E. Georgiadou, “Applying the ISO/IEC 25010 Quality Models to Software Product,” in *Communications in Computer and Information Science*, Springer Verlag, 2018, pp. 492–503. doi: 10.1007/978-3-319-97925-0_42.

[47] “Economic Commission for Europe Executive Committee Centre for Trade Facilitation and Electronic Business Matters arising since the twenty-sixth session,” 2021.

[48] “Analyzing the Implementation of Terminal Operating System on Enhancing the Efficiency of Alexandria Container Terminal.” [Online]. Available: https://www.researchgate.net/publication/390232842

[49] M. Jović, S. Aksentijević, B. Plentaj, and E. Tijan, “PORT COMMUNITY SYSTEM BUSINESS MODELS,” in *34th Bled eConference: Digital Support from Crisis to Progressive Change, BLED 2021 - Proceedings*, University of Maribor Press, 2021, pp. 41–52. doi: 10.18690/978-961-286-485-9.3.

[50] S. Ruecker *et al.*, “Academic Prototyping as a Method of Knowledge Production: The Case of the Dynamic Table of Contexts,” *Scholarly and Research Communication*, vol. 5, no. 2, Sep. 2014, doi: 10.22230/src.2014v5n2a158.

[51] E. Cecchet, “From research prototypes to industrial strength open source products - The ObjectWeb experience,” in *Lecture Notes in Computer Science*, Springer Verlag, 2005, pp. 17–27. doi: 10.1007/978-3-540-30577-4_2.

[52] A. Massahiro Shimaoka, R. Cordeiro Ferreira, and A. Goldman, “The Evolution of CRISP-DM for Data Science: Methods, Processes and Frameworks,” *SBC Reviews on Computer Science*, 2023, doi: 10.13140/RG.2.2.22493.42721.

[53] J. R. Talburt and Y. Zhou, “ISO Data Quality Standards for Master Data,” in *Entity Information Life*

Cycle for Big Data, Elsevier, 2015, pp. 191–205. doi: 10.1016/b978-0-12-800537-8.00011-9.

[54] R. Matheus, M. Janssen, and D. Maheshwari, “Data science empowering the public: Data-driven dashboards for transparent and accountable decision-making in smart cities,” *Gov Inf Q*, vol. 37, no. 3, 2020, doi: 10.1016/j.giq.2018.01.006.

[55] Y. Y. Zhang, Z. Z. Hu, J. R. Lin, and J. P. Zhang, “Linking data model and formula to automate KPI calculation for building performance benchmarking,” *Energy Reports*, vol. 7, pp. 1326–1337, Nov. 2021, doi: 10.1016/j.egyr.2021.02.044.

[56] C. Iphar, I. Le Berre, M. Sahuquet, A. Napoli, and É. Foulquier, “Port calls and vessel trajectory dataset in the Caribbean with accurate port quays survey,” *Data Brief*, vol. 55, p. 110617, Aug. 2024, doi: 10.1016/J.DIB.2024.110617.

[57] B. Malley, D. Ramazzotti, and J. T. yu Wu, “Data pre-processing,” in *Secondary Analysis of Electronic Health Records*, Springer International Publishing, 2016, pp. 115–141. doi: 10.1007/978-3-319-43742-2_12.

[58] A. Omoseebi, G. Ola, and J. Tyler, “Data Preparation and Feature Engineering.” [Online]. Available: https://www.researchgate.net/publication/389860294

[59] “3) Validation and trials.”

[60] R. M. Church, “ HOW TO LOOK AT DATA: A REVIEW OF JOHN W. TUKEY’S EXPLORATORY DATA ANALYSIS 1 ,” *J Exp Anal Behav*, vol. 31, no. 3, pp. 433–440, May 1979, doi: 10.1901/jeab.1979.31-433.

[61] M. Komorowski, “Exploratory Data Analysis,” 2016, doi: 10.1007/978-3.

[62] M. Tessler, “Social Science Research in the Arab World and Beyond A Guide for Students, Instructors and Researchers SpringerBriefs in Sociology.”

[63] Á. López-Oriona and J. A. Vilar, “Analyzing categorical time series with the R package ctsfeatures,” *J Comput Sci*, vol. 76, p. 102233, Mar. 2024, doi: 10.1016/J.JOCS.2024.102233.

[64] D. Petković, “Modern temporal data models: Strengths and weaknesses,” *Communications in Computer and Information Science*, vol. 521, pp. 136–146, 2015, doi: 10.1007/978-3-319-18422-7_12.

[65] H. T. Unal, S. Mete, O. U. Vurgun, A. F. Mendi, O. Ozkan, and M. A. Nacar, “Postgresql Database Management System: ODAK,” in *2023 Innovations in Intelligent Systems and Applications Conference, ASYU 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ASYU58738.2023.10296600.

[66] J. Goldstein and P.-Å. Larson, “Optimizing queries using materialized views,” Association for Computing Machinery (ACM), May 2001, pp. 331–342. doi: 10.1145/375663.375706.

[67] K. Sirigiri, “Enhancing SQL Query Performance: A Case Study on Optimizing Enterprise Data Processing,” *International Journal of Basic and Applied Sciences*, vol. 14, no. 5, pp. 353–360, Sep. 2025, doi: 10.14419/x2wqqh31.

[68] B. Demirel, K. Cullinane, and H. Haralambides, “572.”

[69] V. Hinkka *et al.*, “Terminal Planning: The Selection of Relevant KPIs to Evaluate Operations”, doi: 10.13140/RG.2.2.31916.92807.

[70] Z. H. Munim, C. S. Fiskin, B. Nepal, and M. M. H. Chowdhury, “Forecasting container throughput of major Asian ports using the Prophet and hybrid time series models,” *Asian Journal of Shipping and Logistics*, vol. 39, no. 2, pp. 67–77, Jun. 2023, doi: 10.1016/j.ajsl.2023.02.004.

[71] R. Hassan and R. O. S. Gurning, “Analysis of the Container Dwell Time at Container Terminal by Using Simulation Modelling,” *International Journal of Marine Engineering Innovation and Research*, vol. 5, no. 1, Mar. 2020, doi: 10.12962/j25481479.v4i4.5711.

[72] H. Saeedi, B. Wiegmans, and B. Behdani, “Performance measurement in intermodal freight transport systems: literature review and research agenda,” 2022, *Inderscience Publishers*. doi: 10.1504/WRITR.2022.123098.

[73] Y. Sunitiyoso *et al.*, “Port performance factors and their interactions: A systems thinking approach,” *Asian Journal of Shipping and Logistics*, vol. 38, no. 2, pp. 107–123, Jun. 2022, doi: 10.1016/j.ajsl.2022.04.001.

[74] Z. Zhao, J. Chen, M. Shen, Y. Liang, Z. Wan, and H. Wang, “Multi-equipment integrated scheduling for stereo container terminals,” *Transp Res E Logist Transp Rev*, vol. 201, Sep. 2025, doi: 10.1016/j.tre.2025.104259.

[75] Y. Wang, G. Shi, and K. Hirayama, “Many-Objective Container Stowage Optimization Based on Improved NSGA-III,” *J Mar Sci Eng*, vol. 10, no. 4, Apr. 2022, doi: 10.3390/jmse10040517.

[76] S. Gülmez, Y. Gülmez, and U. P. Tapaninen, “Predicting cargo handling and berthing times in bulk terminals: A neural network approach,” *Case Stud Transp Policy*, vol. 19, Mar. 2025, doi: 10.1016/j.cstp.2024.101351.

[77] S. Oladele, “Design and Implementation of a Real-Time Data Processing Framework for High-Throughput Applications.” [Online]. Available: https://www.researchgate.net/publication/388194955

[78] L. Li, M. Yang, and P. Jiang, “Converting Relational Databases to Manufacturing Knowledge Graph for Product Quality Tracing in Additive Manufacturing,” in *Procedia CIRP*, Elsevier B.V., 2025, pp. 544–549. doi: 10.1016/j.procir.2025.02.172.