Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 19, No. 1, 2025

Design and Implementation of a Ship Entry–Exit Report Data Analysis Platform Based on Big Data Technologies

Taizhi Lv *

School of Information Engineering, Jiangsu Maritime Institute, Nanjing 211170, China

* **Corresponding author Email:** lvtaizhi@163.com

Abstract: With the continuous expansion of global maritime transport, ship entry–exit report data have become essential for port management, shipping operations, and safety supervision. This study builds a complete data analysis platform based on MySQL, Spark, Spring Boot, and Echarts, enabling full-process handling from data acquisition, preprocessing, and distributed analysis to visual presentation. The platform adopts a hybrid architecture that integrates batch processing and stream processing to enhance both real-time monitoring and historical data analytics. Through the visualization module, the system improves interpretability of data and provides intuitive decision support for port authorities. Featuring high extensibility, clear modular design, and applicability to real port scenarios, the proposed platform offers a feasible technical solution for smart port construction.

Keywords: Ship Entry–Exit Report; Spark Distributed Computing; Spring Boot; Data Visualization; Echarts; AIS Data.

1. Introduction

With the accelerated development of global supply chains, maritime transport now carries over 90% of global trade volume, making ports crucial hubs that connect national and international economies. Ship entry–exit data, as core operational data of ports, contain not only temporal information of port operations but also reflect cargo flow characteristics, port resource load, and route operational status [1]. These data serve as key indicators for evaluating port efficiency, logistics stability, and shipping organizational capability. In recent years, as port throughput continues to grow, ship sizes increase, and route networks become more complex, both the scale and dimensionality of ship entry–exit data have expanded exponentially, significantly increasing the difficulty of data management.

Traditional approaches relying on manual statistics, standalone databases, or offline analyses can no longer meet the needs of real-time monitoring, dynamic scheduling, and trend forecasting. For example, ship schedule changes, berth delays, anchorage congestion, and cargo flow fluctuations must be identified promptly to maintain operational stability. However, manual processing cannot achieve minute- or second-level responsiveness. Furthermore, heterogeneous formats, inconsistent data quality, and fragmentation among data sources exacerbate governance challenges, causing a considerable portion of valuable information to remain underutilized [2-3].

Against this background, building a digital analysis platform tailored to ship entry–exit operations has become imperative. The platform must support high-concurrency data processing, rapid access to multi-source heterogeneous inputs, real-time stream processing, and extraction of interpretable indicators from massive historical data. Additionally, intuitive visualization is required to enable port operators to quickly grasp underlying business patterns to support berth planning, tugboat scheduling, yard allocation, and route optimization. Internationally, advanced ports such as Singapore and Hamburg have already established minute-level response systems based on AIS data, big data analytics, and intelligent decision-making models. In contrast, although many Chinese ports have built digital infrastructures, gaps remain in data governance, real-time computation, visualization depth, and engineering implementation. Thus, a high-performance, scalable platform provides not only research value but also practical significance for smart port development..

2. Integration of Smart Shipping Projects into Teaching Resource Development

To meet the challenges described above and develop a system capable of stable operation in real port environments, this study incorporates several key technologies, including MySQL relational databases, the Spark distributed computing framework, the Spring Boot application framework, and the Echarts visualization engine. These technologies collectively form the platform’s storage foundation, computational core, and presentation layer.

MySQL handles structured storage of ship entry–exit records, basic ship information, and cargo attributes. Through well-designed table schemas, primary keys, and partitioning, MySQL enables efficient access under high concurrency and supports large-scale historical queries [4].

Spark serves as the core computational engine. Its in-memory computing capabilities significantly improve the efficiency of cleaning, transforming, aggregating, and analyzing massive ship data [5]. The platform employs both batch modules and streaming modules in Spark: batch modules handle long-term trend analyses, such as port stay duration distribution, ship type structure evolution, and annual throughput; streaming modules parse real-time AIS signals to detect schedule deviations, berth conflicts, anchorage clustering, and abnormal trajectories.

Spring Boot provides unified service orchestration for APIs, data interfaces, distributed task triggering, and permission management [6]. RESTful APIs allow the front-end to access analytics results efficiently, while Spark execution can be triggered on demand.

Echarts provides dynamic, interactive visualization with zooming, filtering, timeline sliding, and multi-dimensional overlays. More than 20 visualization components—including trend graphs, distribution charts, heatmaps, and port-stay histograms—enable intuitive decision support [7].

3. Data Acquisition and Preprocessing

3.1. Data Sources and Structuring

The data on ship arrivals and departures primarily originate from several core operational systems, including maritime supervision systems, AIS (Automatic Identification System) vessel positioning systems, and port ERP/TOS operational systems. Maritime supervision systems typically provide structured administrative data such as vessel declarations, port entry and exit permits, and dangerous goods notifications. AIS data, automatically transmitted by vessels, contain real-time information on a ship's latitude and longitude, course, speed over ground, and intended destination. Port ERP/TOS systems record key operational data including berthing plans, loading and unloading schedules, and container flow information. These datasets differ significantly in source, format, and update frequency, making a unified data-structuring workflow essential for standardization and integration.

The core of the structuring process includes field alignment, time-format unification, code standardization, and spatial coordinate transformation. First, vessel identifiers from different sources must be mapped to a unified primary key (such as MMSI or IMO number). Second, time fields must be standardized to local port time or UTC to support accurate time-series analysis. Third, positional data must be converted to the WGS-84 coordinate system to ensure compatibility with geospatial analytical models. In addition, raw data need to be reorganized into thematic tables—such as vessel basic information, entry–exit records, and AIS trajectory datasets—followed by type validation and initial normalization to ensure that the processed data meet the requirements of large-scale computation and analytics in subsequent stages.

3.2. Data Cleaning and Missing-Value Handling

Due to the influence of signal quality, device conditions, manual input errors, and other operational factors during data acquisition, raw ship-related data often contain timestamp anomalies, inconsistencies between speed and trajectory, GPS drift, duplicate records, and missing fields. Therefore, data cleaning becomes a critical step to ensure analytical accuracy. To address common issues in AIS data, such as sudden jumps and drift points, this study employs velocity-threshold detection and spatial continuity checks to identify anomalies, and applies trajectory interpolation methods to repair abnormal points. For duplicate records that may appear in maritime supervision systems or TOS systems, deduplication is performed using composite primary keys (e.g., MMSI + timestamp).

Missing-value handling is conducted based on data types and analytical requirements using different imputation strategies. For highly continuous trajectory data, interpolation is performed using time-based linear methods, nearest-neighbor trajectory points, or velocity-vector estimation. For categorical fields such as ship type or cargo category, missing values are completed using recently recognized attributes or corresponding port operation records. For operational data—such as incomplete berthing plans—missing items may be matched and filled through auxiliary systems (e.g., port scheduling systems). Records with excessive missing values or severe structural defects are removed according to predefined thresholds to ensure the reliability and internal consistency of the final dataset.

3.3. Standardization and Index Construction

Data standardization is a critical step for achieving interoperability across heterogeneous data sources. The first requirement is to unify field definitions by mapping vessel identifiers, dimensional parameters, voyage information, and port codes into standardized attributes. Second, measurement units must be harmonized across datasets—for example, normalizing speed to knots, tonnage to DWT/GT, and cargo weight to metric tons—to ensure dimensional consistency. Third, all positional data are converted into a common coordinate system and spatially annotated according to port boundary polygons to support subsequent regional identification, density analysis, and other geospatial computations.

Index construction is primarily aimed at performance optimization. Given the pronounced time-series characteristics of port-related data, this study establishes a timestamp-based single-column index and further constructs composite indexes combining vessel identifiers (MMSI/IMO) to accelerate trajectory retrieval and event tracing. For spatial analytical tasks—such as vessel density estimation or anchorage clustering—spatial indexes based on GeoHash or QuadTree significantly improve query efficiency. In addition, secondary indexes on operational fields, such as port codes and route identifiers, enhance multi-dimensional cross-analysis performance. Through the combined design of standardized fields and an optimized indexing system, the platform provides a robust and high-quality data foundation for subsequent Spark-based computation and online querying..

4. Spark-Based Data Analysis Models

4.1. Batch Processing Model Design

The batch processing model primarily targets large-scale historical data for periodic analysis and statistical computation. Over long-term operations, ports accumulate massive volumes of entry–exit records, AIS trajectory data, cargo handling logs, and scheduling information, all of which exhibit characteristics such as long temporal spans, large storage sizes, and multiple analytical dimensions. Leveraging its distributed in-memory computation engine, Spark can efficiently aggregate, join, and model terabyte-level datasets, making it well-suited for batch processing of historical vessel data.

Typical batch tasks include calculating port stay durations, discovering route patterns, generating monthly or annual throughput statistics, and evaluating port resource utilization. For route-cycle analysis, vessel MMSI, destination port, and voyage identifiers are used to aggregate trajectories, enabling identification of average route cycles and their fluctuations, which supports schedule forecasting and route optimization. Spark MLlib further provides large-scale machine learning capabilities—including regression, classification, and clustering—which can be applied to ETA (Estimated Time of Arrival) prediction, congestion risk identification, and behavioral pattern modeling. Trained models may also be periodically updated to adapt to dynamic changes in port operations, thereby enhancing the intelligence and adaptability of the data analysis platform.

4.2. Stream Processing Model Design

The stream processing model focuses on real-time data, particularly second- or minute-level parsing of AIS signals and port monitoring data. Spark Structured Streaming offers micro-batch and near-real-time processing capabilities, enabling filtering, aggregation, and windowed computations on continuous incoming data streams to support real-time vessel monitoring and event detection. The streaming module identifies critical events such as berthing delays, anchorage congestion, abnormal speed changes, sudden course shifts, extended stays, and potential trajectory deviations. These anomalies have direct operational implications: prolonged stays may lead to channel congestion, whereas plan deviations disrupt berth allocation and scheduling. Through window functions—such as sliding and tumbling windows—the system performs real-time temporal aggregation, generating dynamic indicators including the number of vessels currently in port and the number of vessels with abnormal speeds in specific anchorages.

To ensure performance and stability, the system adopts message queue middleware such as Kafka as the data ingestion layer, enabling Spark Streaming to operate reliably under high-throughput conditions. Furthermore, checkpointing and state management mechanisms ensure task recovery after failures or network fluctuations, significantly enhancing reliability in high real-time computing scenarios.

4.3. Hybrid Architecture Combining Batch and Stream Processing

In the platform designed in this study, batch and stream processing complement each other. The real-time streaming module continuously interprets AIS data to detect operational anomalies promptly. The batch processing module periodically updates distributions of port stay durations, vessel-type composition, and cargo throughput trends. Outputs of batch analyses—such as ETA prediction model parameters or anomaly detection thresholds—are fed back to the stream processing module to enhance real-time detection accuracy. This bidirectional interaction allows the system to maintain high real-time responsiveness while continuously improving predictive performance through long-term historical data, thereby forming an integrated closed-loop analytical framework.

5. Spark-Based Data Analysis Models

5.1. Batch Processing Model Design

The platform's core modules—including data querying, task triggering, result encapsulation, and visual output—are uniformly orchestrated and managed through Spring Boot. To support continuous access to and parsing of multi-source heterogeneous port data, the backend is responsible for building a reliable data access layer that enables efficient querying, retrieval, and aggregation of structured MySQL tables such as vessel basic information, entry–exit logs, and AIS trajectory datasets. This ensures that both the historical data analysis module and the front-end presentation module receive stable and consistent data inputs.

To achieve millisecond-level rendering of real-time data in the visualization layer, the backend integrates multithreaded processing and a Redis caching module, which together improve throughput in high-concurrency scenarios while significantly reducing database load. After Spark completes distributed computation, Spring Boot performs data compression, organization, and formatting, converting analytical outputs into JSON structures suitable for Echarts visualization before delivering them to the front end via REST APIs. Through this mechanism, charts such as ship type distribution, port stay duration analysis, and cargo throughput trends can be rendered rapidly, enabling the entire platform to maintain excellent responsiveness and usability even when processing large-scale datasets.

5.2. Visualization Using Echarts

In the front-end presentation layer, Echarts serves as the primary visualization framework, responsible for dynamic display and interactive exploration of analytical results. Compared with traditional static charts, Echarts supports smooth animations, diverse chart types, and highly configurable interactive features, enabling users to intuitively interpret complex maritime operational data. Based on Echarts, the platform constructs a series of visualization components—including vessel type distribution charts, port stay duration distributions, and cargo throughput trend plots—to enhance interpretability and readability of analytical outcomes.

To illustrate its functionality, the platform uses real operational data from a typical river port as an example. This port generates large volumes of vessel entry–exit records, AIS dynamic data, and loading/unloading logs daily. Given the complexity, heterogeneity, and high update frequency of these datasets, traditional manual statistics and isolated query methods are insufficient for real-time monitoring, synchronized data interaction, or data-driven decision-making.

Figure 1 presents the proportion of different vessel categories—such as container ships, bulk carriers, tankers, ro-ro vessels, and passenger–ro-ro vessels—during the statistical period at the selected port. The distribution offers insights into the port's functional attributes and service specialization.

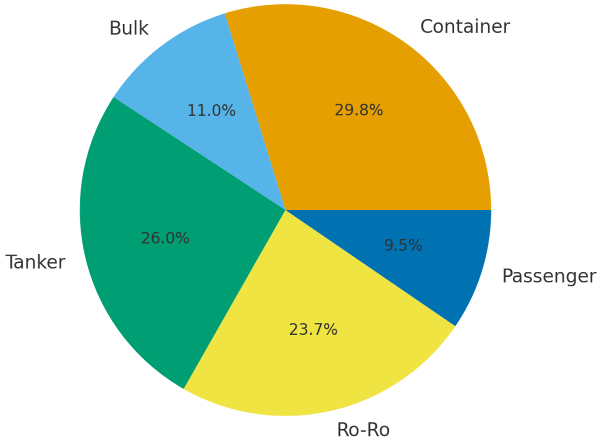

Figure 1. Ship type distribution

Figure 2 illustrates the distribution of port stay durations, a key metric directly reflecting operational efficiency, berth turnover capability, and organizational performance. The distribution typically exhibits right skewness, indicating that while most vessels complete operations within a normal timeframe, a small number exhibit extended stays.

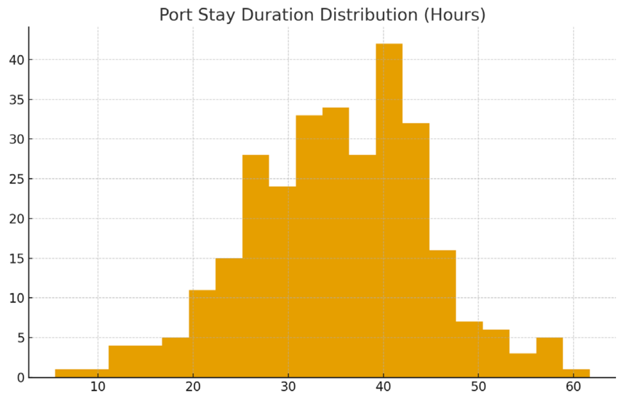

Figure 2. Distribution of port stay durations

Figure 3 displays the monthly cargo throughput of the port over the course of a year, capturing fluctuations and seasonal patterns in port operations. Throughput values are influenced by multiple external and operational factors, including global trade conditions, market demand, holidays, seasonal operations, and extreme weather, resulting in clearly observable cyclical characteristics.

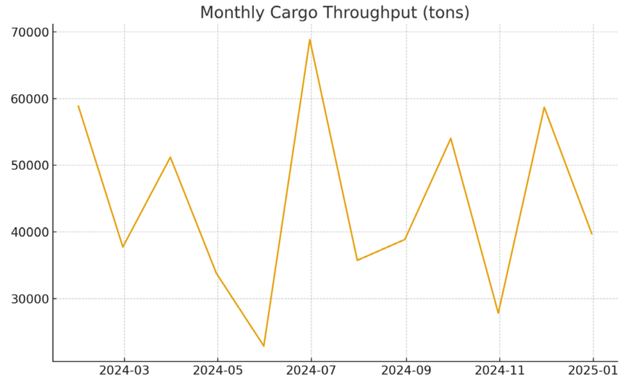

Figure 3. Monthly cargo throughput

6. Conclusion

The study constructs a ship entry–exit data analysis platform tailored to port management scenarios, integrating modules for data acquisition, preprocessing, standardized governance, Spark-based distributed computing, and Echarts-driven visualization. The platform achieves an end-to-end data processing pipeline that connects raw data with business-level analytical outputs. Its architectural design maintains strong modularity and loose coupling, ensuring high availability even in complex and heterogeneous data environments. Functionally, the deep integration of batch and stream processing enables both large-scale historical trend analysis and real-time operational monitoring. In terms of system performance, the combination of Spring Boot service encapsulation and Redis-based caching allows the front-end visualization layer to render analytical results within milliseconds, significantly enhancing user experience and the timeliness of port operational decision-making.

Despite the comprehensive achievements made in system engineering, data governance, and visualization application, there remain opportunities for further enhancement. Future research may advance the platform in three directions. First, incorporating machine learning and deep learning models can support more intelligent functionalities such as ETA prediction, congestion risk estimation, and operational efficiency assessment. Second, developing a finer-grained risk warning system—powered by time-series modeling, behavioral pattern recognition, and anomaly detection algorithms—can provide real-time safety and congestion alerts for port operations. Third, exploring cross-port data collaboration mechanisms will enable integration and sharing of data from different regions and port types, forming regional or even national-level maritime big data frameworks. Such capabilities would extend the platform's applicability and offer a robust data foundation for the construction of smart port clusters.

Acknowledgements

This work was financially supported by the funding of the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (23KJA580002),, and the Excellent Teaching Team for QingLan Project of the Jiangsu Higher Education Institutions of China (Big Data Technology Teaching Team with Shipping Characteristic).

References

[1] Tsou, Ming-Cheng. "Big data analytics of safety assessment for a port of entry: A case study in Keelung Harbor." Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment 233.4 (2019): 1260-1275.

[2] Liu, Zhao, et al. "A data mining method to extract traffic network for maritime transport management." Ocean & Coastal Management 239 (2023): 106622.

[3] Hong, Hyunsu, et al. "Incorporation of shipping activity data in recurrent neural networks and long short-term memory models to improve air quality predictions around busan port." Atmosphere 12.9 (2021): 1172.

[4] Rawat, Bhupest, and Suryari Purnama. "Mysql database management system (dbms) on ftp site lapan bandung." International Journal of Cyber and IT Service Management 1.2 (2021): 173-179.

[5] Azeroual, Otmane, and Anastasija Nikiforova. "Apache spark and mllib-based intrusion detection system or how the big data technologies can secure the data." Information 13.2 (2022): 58.

[6] Menezes, Gabriel, Bruno Cafeo, and Andre Hora. "How are framework code samples maintained and used by developers? The case of Android and Spring Boot." Journal of Systems and Software 185 (2022): 111146.

[7] Cao, Wenjun, et al. "Epidemic Management System based on SpringBoot and Oriented for Echarts." International Core Journal of Engineering 8.12 (2022): 202-211.