Journal of *Marine Science and Engineering*

MDPI

Article

Development of a Maritime Transport Emulator to Mitigate Data Loss from Shipborne IoT Sensors

Chae-Rim Park, Do-Myeong Park, Tae-Hoon Kim, Byung O Kang and Byung-Kwon Park *

Smart Logistics and Research Center, Dong-A University, Busan 49315, Republic of Korea; clpark@dau.ac.kr (C.-R.P.); home21cc@dau.ac.kr (D.-M.P.); rider7979@dau.ac.kr (T.-H.K.); bokang@dau.ac.kr (B.O.K.)

* Correspondence: bpark@dau.ac.kr; Tel.: +82-10-3254-9260

Abstract: Recently, the maritime logistics industry has been transitioning to smart logistics by leveraging such technologies as AI and IoT. In particular, maritime big data plays a significant role in providing various services, including ship operation monitoring and greenhouse gas emissions assessment, and is considered essential for delivering maritime logistics services. Marine big data comprise real-world data collected during ship operations, but it is susceptible to loss due to temporal and environmental constraints. To address this issue, an Emulator is proposed to generate supplemental data, including location data, data count, and average distance, using accumulated maritime transport data. This study proposes an Emulator that repetitively generates new data such as location data, data count, and average distance using maritime transport data accumulated up to now. The location data is generated using the cumulative distance and trigonometric ratios based on the location information of standard routes. The data count and average distance are calculated based on user-input parameters such as voyage time and data interval. The generated data is inserted into a database and monitored on a map in real time. Experiments were conducted using maritime transport route data, and the results validated the effectiveness of the Emulator.

Keywords: big data; emulator; data generation; smart container; smart logistics

check for updates

Academic Editor: Fausto Pedro García Márquez

Received: 3 February 2025
Revised: 14 March 2025
Accepted: 18 March 2025
Published: 22 March 2025

Citation: Park, C.-R.; Park, D.-M.; Kim, T.-H.; Kang, B.O.; Park, B.-K. Development of a Maritime Transport Emulator to Mitigate Data Loss from Shipborne IoT Sensors. *J. Mar. Sci. Eng.* **2025**, *13*, 637. https://doi.org/10.3390/jmse13040637

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Technologies of the 4th Industrial Revolution, such as AI (Artificial Intelligence), IoT (Internet of Things), Big Data, Blockchain, AR (Augmented Reality), and VR (Virtual Reality), are driving innovation across various industries [1,2]. In maritime logistics, these technologies are facilitating the transition to smart logistics through process innovation and real-time data analysis. The conversion to smart logistics is accelerating the collection of logistics data and the advancement of marine big data technologies. Particularly, there is growing interest in the creation of new services based on marine big data, which serves as a vital foundation for expanding new marine industries such as renewable energy, autonomous ships, and smart maritime ports [3]. Governments and companies are collecting and analyzing data from ships, integrating it with legacy systems to improve operational efficiency and market analysis [4,5]. In South Korea, the Ministry of Oceans and Fisheries leverages marine data to provide customized services across various fields, including marine fisheries, aquaculture, and port operations. The Marine Fisheries Big Data Platform offers diverse services such as average import seafood prices, marine accident forecasts, and ship operation monitoring and analysis [6]. Companies are also developing tailored services. For example, AllSeaData has developed a maritime shipping big data

system that objectively evaluates greenhouse gas emissions from ships by utilizing global ship movement information and maritime environmental data [7,8]. Similarly, Swinnus has created a device called "ConTracer", which attaches to containers to collect location, temperature, and humidity data, enabling monitoring through a dedicated platform.

As illustrated in the preceding examples, several maritime services are actively operating, and the collection of extensive maritime transport data is essential for providing such services. However, maritime transport faces challenges such as erroneous values from IoT equipment malfunctions and permanent data loss caused by environmental factors like adverse weather. These factors may result in the loss of containers equipped with IoT devices. To mitigate these challenges, this study proposes an Emulator algorithm that generates virtual location data based on collected maritime transport data. Previous studies employed random-matching techniques, whereas this study improves upon such approaches by allowing users to input specific voyage parameters.

While earlier research efforts attempted to address data loss issues through statistical methods, a notable study [9] utilized standard deviation calculations to generate new location data points. This approach, although straightforward, relied solely on pre-existing coordinate variations and did not account for spatial continuity or navigational accuracy. Consequently, it introduced positional inconsistencies and lacked adaptability to real-world maritime transport patterns.

In contrast, the proposed Emulator in this study advances beyond such limitations by incorporating trigonometric calculations and user-defined voyage parameters, ensuring that generated data maintains spatial integrity and realistic movement trajectories. By leveraging accumulated maritime transport data, the Emulator systematically computes expected data points, inter-location distances, and average speed, aligning with actual vessel navigation behavior. Additionally, the ability to dynamically adjust voyage parameters offers enhanced control and adaptability, significantly improving the accuracy of reconstructed data.

The need for this research is strongly driven by the increasing complexity of global maritime logistics and the growing reliance on real-time data for efficient supply chain management. Traditional data collection methods often fail to account for data gaps caused by equipment failure, harsh maritime environments, and limited satellite coverage. In response to these challenges, this study improves upon the previous approach by allowing users to input specific parameters such as "voyage time" and "data collection interval". Based on these inputs, the system automatically calculates collection times, the number of collected data points, the average speed between data points, and the distances between generated location data. To ensure data accuracy and distribution, the newly generated data is inserted into a database in real time. It is then monitored through a map interface for streamlined visualization. This study introduces an innovative approach to address data loss issues while enhancing the predictive accuracy of maritime transport patterns. As a result, it improves decision-making and optimizes shipping routes through automated data generation.

2. Development of a Maritime Transport Emulator

The development of the maritime transport Emulator is guided by the necessity of reconstructing missing transport data while maintaining spatial consistency [10,11]. Traditional methods, such as interpolation techniques or stochastic modeling, often fail to reflect real-world maritime movement patterns accurately [12,13]. This study employs a structured algorithm that integrates trigonometric calculations with user-defined voyage parameters to ensure precise and scalable data reconstruction. The key design principles

include flexibility, adaptability to real-world transport conditions, and automation to minimize manual intervention.

The structure of the algorithm proposed in this study is illustrated in Figure 1. To facilitate data processing, maritime transport data collected in real time from smart containers equipped with IoT devices are utilized. Among the collected data, the location data of the standard route are used, and the Emulator's operational flow is described in Figure 2. The algorithm begins by calculating distances and angles between existing location data points, followed by statistical analysis to determine standard deviations. Based on these calculations, the Emulator dynamically generates missing location data points while maintaining spatial accuracy. Before generating supplementary location data, the Emulator processes the maritime transport data collected through several preprocessing steps. These include filtering out erroneous data points caused by IoT sensor malfunctions, interpolating missing coordinates where feasible, and normalizing voyage trajectories to standard geospatial formats. The processed data are then structured into sequential transport patterns, which serve as the foundation for generating realistic maritime routes.

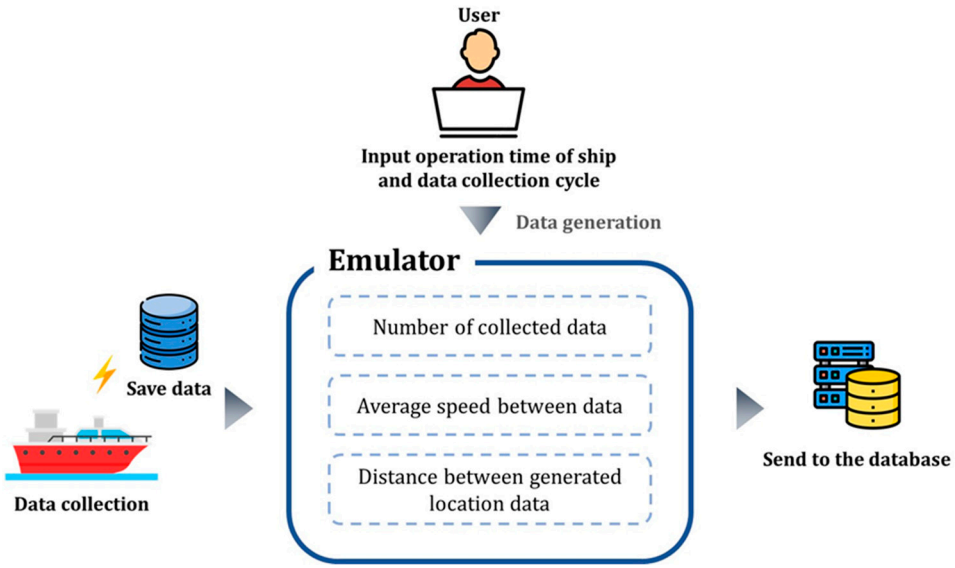


Figure 1. Proposed Emulator Algorithms.

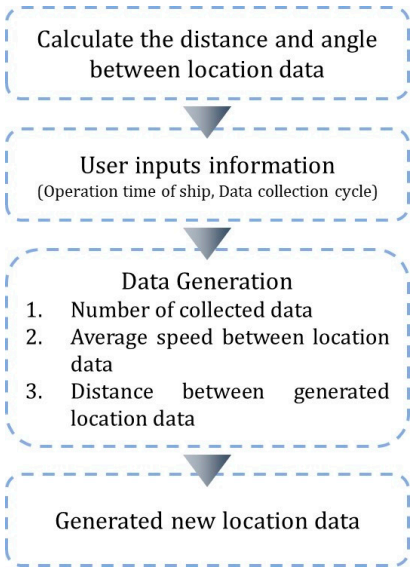


Figure 2. Development of a Maritime Transport Emulator.

Once users input voyage parameters such as voyage duration and data collection intervals, the Emulator computes the expected number of data points, the average speed between data points, and the distances between newly generated locations. The core mechanism of data generation relies on trigonometric calculations to maintain the natural trajectory of maritime movements. The Emulator determines angular deviations based on historical route data and calculates new coordinates using sine and cosine transformations. Unlike conventional random-matching techniques, this approach ensures that generated data points reflect realistic navigational behavior, preserving maritime route integrity. By applying trigonometric ratios, the system accurately determines the coordinates of the missing data, ensuring smooth integration into the existing transport dataset. Finally, the computed data are inserted into a database and visualized through a real-time map interface, facilitating seamless monitoring and analysis. The generated location data are stored in a structured database and integrated with geospatial visualization platforms for real-time monitoring. This structured data format allows seamless interoperability with existing maritime data analysis tools, enabling enhanced predictive analytics and decision support. The effectiveness of this approach is further validated through experimental evaluations, as detailed in the subsequent sections.

Building upon this foundation, the next phase focuses on calculating the coordinates of the newly generated location data using trigonometric ratios. Following this step, the computed location data is systematically inserted into a database and continuously monitored through a map interface. This structured approach ensures that the Emulator maintains data accuracy and enables real-time visualization, ultimately enhancing maritime logistics management.

2.1. Calculation of Distance and Angles Between Location Data

Accurate calculation of distances and angles between location data points is fundamental for generating realistic maritime transport routes. Conventional distance estimation methods, such as Euclidean distance, may not account for the curvature of maritime routes. Instead, this study utilizes great-circle distance calculations, ensuring that generated waypoints align with actual vessel trajectories. Additionally, angular deviations are computed to reflect natural navigational adjustments made by ships due to weather conditions and maritime traffic patterns. These calculations serve as the foundation for generating supplementary location data that accurately represent real-world shipping movements.

2.1.1. Distances Between Standard Routes (Distgroup)

To generate realistic location data, the emulator categorizes distance values into groups (Distgroup) based on statistical properties. This approach mitigates anomalies caused by irregular sensor readings and ensures that the generated location points follow realistic spatial distributions. The standard deviation values used in Distgroup are derived through Poisson-based probability filtering, allowing the selection of the most probable distances while discarding outliers. This structured grouping facilitates robust distance estimation, preventing the clustering of generated data points in specific regions and enhancing spatial consistency.

First, using the distance group from the standard routes collected in Section 2.1, a group of standard deviations is generated. The standard deviation values are derived based on a Poisson distribution, from which the 40 most probable values are selected to form a group. The number of selected values can be adjusted as needed. The standard deviations in the group are then randomly sampled and sequentially summed, as illustrated in Figure 3. This process serves as the foundation for generating new data points with realistic variability.

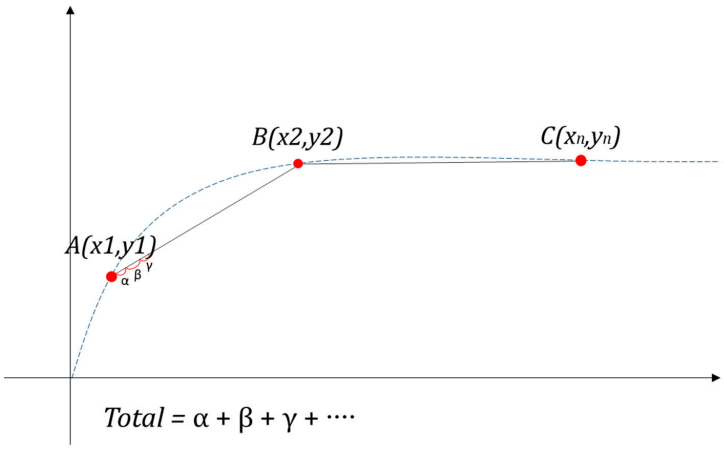


Figure 3. Add the Standard Deviation of the Distance Between the Standard Routes.

The dashed curve represents the standard route, with coordinates A, B, and C assumed as the location coordinates of the standard route. The values α, β, and γ represent randomly sampled standard deviations, and Total denotes the cumulative sum of the extracted standard deviations. This process of randomly extracting and summing standard deviations is repeated, ensuring that the total sum does not exceed the total distance of the standard route.

2.1.2. Angles Between Standard Routes

Angular calculations play a crucial role in maritime navigation, influencing turn radius, vessel maneuverability, and course deviation. The emulator extracts angular deviations by analyzing multi-point connections along standard routes, filtering out irregularities using a weighted moving average technique. These refined angles are then incorporated into the generated data, ensuring that newly created waypoints align with realistic vessel heading adjustments. This approach effectively replicates navigational patterns observed in real-world maritime transport.

The angles between standard routes were grouped, as illustrated in Figure 4. The dashed curve represents the standard route, with coordinates A, B, and C assumed as the location coordinates of the standard route. The angles α and β correspond to the angles between these location coordinates. The distance and angle data obtained from the standard routes in Section 2.1 serve as essential inputs for generating new location data in Section 2.3. These data provide the foundation for accurately replicating realistic positional patterns in the newly generated data.

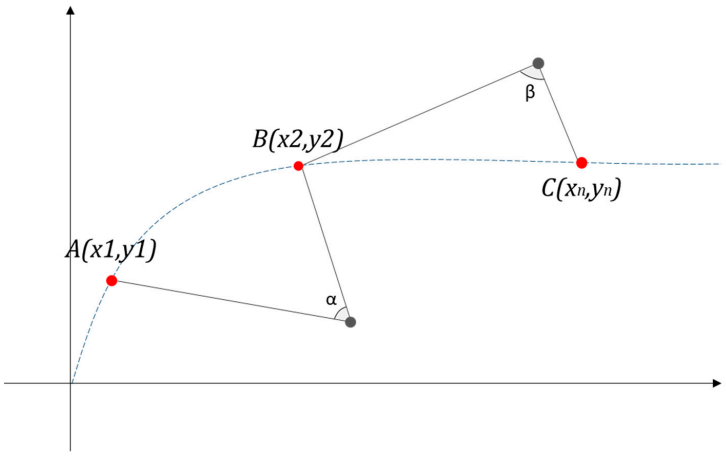


Figure 4. Angle of Distance Between Standard Routes.

2.2. Users Input Information

User-defined parameters, such as voyage duration and data collection interval, significantly influence the accuracy and distribution of generated location data. To ensure consistency, the emulator incorporates a validation mechanism that dynamically adjusts for unrealistic input values, such as excessively high or low collection intervals. Additionally, sensitivity analyses are conducted to determine optimal parameter ranges, minimizing deviations from real-world transport data.

In the user information input process, the user provides the "voyage duration" and "data collection interval" as inputs. Based on these inputs, the system approximately calculates and extracts the following parameters:

- **Number of Data Points** represents the total number of data collection points during the voyage.
- **Average Speed Between Data Points** represents the estimated average speed for each segment of the voyage.
- **Distances Between Locations** represents approximate intervals between consecutive location data points.

This process ensures that the generated data aligns with the user's specified conditions, providing a realistic foundation for further data generation and analysis.

2.2.1. Number of Collected Data

The number of collected data points directly impacts the resolution and accuracy of the generated transport data. A higher number of data points allows for finer granularity in reconstructing maritime routes, while an insufficient number may result in overly simplified trajectories. Through empirical testing, this study determined an optimal range for data collection intervals, balancing computational efficiency with spatial precision.

The number of collected data points refers to the total count of new data generated along a single route based on the user-defined interval. The calculation is defined by Equation (1) below:

In Equation (1):

- **Total time** represents the voyage duration input by the user.
- **Data collection interval** denotes the time interval between consecutive data collections.

The integer quotient of the result is used to approximate the number of collected data points, ensuring alignment with the user-defined parameters.

$$\text{Number of collected data} = \frac{\text{Total time}}{\text{Data collection interval}} \tag{1}$$

2.2.2. Average Speed Between Data

The average speed between data points is calculated using the group of distances between standard routes and the data collection interval. The calculation is expressed as follows in Equation (2):

In Equation (2):

- **Distgroup** represents the group of distances between standard route points.
- **i** is the index within the Distgroup.
- **Data collection interval** denotes the time interval for data collection.

This formula provides a realistic estimation of the average speed between data points based on the spatial and temporal parameters of the standard route.

$$\text{Average speed} = \frac{\text{Distgroup}[\text{i}]}{\text{Data collection interval}} \tag{2}$$

2.2.3. Distance Between Generated Location Data (DGL)

The generated location data must be created at regular intervals to prevent clustering in specific areas. The calculation is defined by Equation (3) below:

In Equation (3):

- **Distgroup** represents the total length of the standard route.
- **Number of Collected Data** refers to the total number of collected data points.

This formula ensures that the generated location data points are evenly distributed along the route, maintaining spatial consistency and avoiding excessive clustering.

$$\mathrm{DGL} = \frac{\mathrm{Distgroup}}{\mathrm{Number\ of\ Collecte\ ddata}} \tag{3}$$

In Figure 5, the red circular markers (●) represent the location data collected from the standard route during transportation, while the blue square markers (■) indicate the generated location data. The values for Speed and Distance between locations are derived from Equations (2) and (3), providing random speeds and distances between the newly generated location data points. The number of these data points (i) is dynamically determined based on the calculated distances between the standard route points and the extracted number of data points collected. This approach ensures flexibility in the data generation process, aligning it with the characteristics of the standard route and the user-defined parameters.

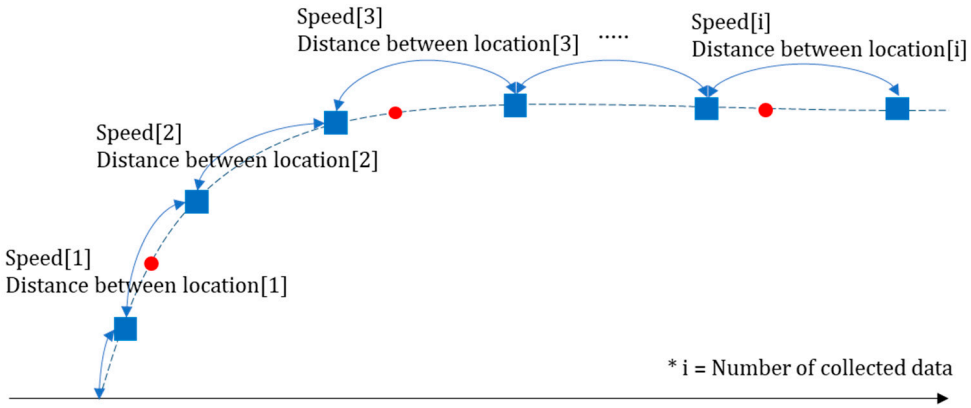


Figure 5. Distance and Speed Between Generated Location data.

Table 1 provides an example of the user information input described earlier. The distance corresponds to the route used in this study, which is approximately 993,000 m. Total time and Collection cycle are user-input values, while the Number of collected data, Average speed, and Distance between locations are automatically calculated based on the user-provided inputs. This table demonstrates the system's ability to process input parameters and generate corresponding output values dynamically.

Table 1. User-Input Items and Auto-Calculated Data Items.

Total Route Distance (km)	Total Time (minute)	Collection Cycle	Number of Collected Data	Average Speed (km/h)	Distance Between Location (km)
993,000	150	10	15	0.0199	66
	250	12	20	0.0166	49
	400	27	14	0.0074	70

2.3. Generation of New Location Data

The Emulator generates new location data by leveraging previously recorded transport patterns and integrating user-specified parameters. Trigonometric calculations are applied to interpolate missing coordinates, ensuring that generated points maintain alignment with standard maritime routes. Additionally, data integrity is preserved by cross-validating generated values against historical datasets, mitigating deviations from real-world navigational patterns. This ensures that the emulator not only reconstructs missing data but also enhances the predictive accuracy of maritime transport routes.

2.3.1. Extraction of Location Coordinates (Latitude and Longitude)

New location data is generated using distance, angle, and trigonometric ratios, following the methodology described below. The newly generated location data must avoid being concentrated in specific areas and should maintain appropriate distances between points. To achieve this, the distances between the generated location data were extracted and applied, as expressed in Equation (4):

In Equation (4):

- **STD** represents the randomly selected standard deviation.
- **DGL** denotes the distance between the generated location data points.
- **Number of Collected Data** refers to the total number of collected data points.
- **i** represents the index of the data point.

$$(\text{STD} + \text{DGL}) \times \text{Number of Collected Data[i]} \tag{4}$$

To generate the coordinate values of the new location data, both distance and angle values are required. The newly generated location data identifies the angle value closest to its current position and uses this angle to calculate its coordinates. Notably, the angle values used in this process are derived from the angle set of the standard route, ensuring that the new location data aligns with the directional patterns of the original route. This approach maintains spatial consistency and a realistic distribution of the generated data.

Figure 6 and Table 2 illustrate the process of generating location coordinate values based on the methods described earlier. In Figure 6, the red circles (●) represent the location data collected from the standard route, while the blue squares (■) indicate the generated location data. This representation shows how the generated location data retrieves angle values according to specific criteria and methods.

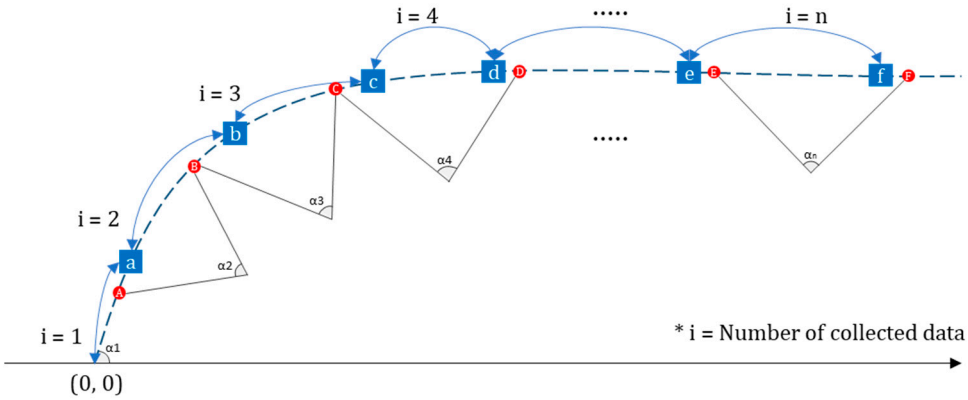


Figure 6. To Read the Angle Value.

Table 2. Example of Reading Angle Value.

Generated Location	Near-Location of Standard Routes	Near-Angle Value
a	A	A2
b	B	A3
c	C	A4
d	D	A5
e	E	α_{n-1}
f	F	α_n

Through this process, the newly generated location data is assigned both distance and angle values, which are used to calculate its coordinates. To calculate the location coordinate values, trigonometric ratio equations are used, as shown in Equations (5) and (6):

$$\mathrm{X} = \mathrm{Distance} \times \cos(\mathrm{angle} \times (\mathrm{Angle})) \quad (5)$$

$$\mathrm{Y} = \mathrm{Distance} \times \cos(\mathrm{angle} \times (\mathrm{Angle})) \quad (6)$$

In Equations (5) and (6):

- **Distance** represents the distance of the generated location data.
- **Angle** refers to the angle closely associated with the distance.
- **X** and **Y** correspond to the generated latitude and longitude values, respectively.

The latitude and longitude values extracted using these equations are unique and non-overlapping, ensuring that each generated location data point maintains its distinct spatial position. This approach ensures accurate and consistent coordinate generation while preventing duplication.

2.3.2. Normalization

As previously explained, the new location data is generated based on the total distance of the standard route. The start point is set at (0, 0), and the endpoint corresponds to the endpoint of the total distance of the standard route. However, since the fixed start and end points are often unclear, generating coordinates within the bounds of the standard route becomes challenging. To address this issue, the latitude and longitude values of the generated location data are normalized within the range of the standard route's latitude and longitude values. This normalization process is represented by Equation (7):

$$\mathrm{Normalization} = (\mathrm{b} - \mathrm{a}) + \frac{x - x(min)}{x(max) - x(min)} + a \quad (7)$$

In Equation (7):

- **x** represents the newly generated location data.
- **x(min)** and **x(max)** are the minimum and maximum values of the generated location data.
- **[a,b]** is the specified range for normalization.

3. Experimental Validation

Unlike the subsequent experiment section, which explores various operational scenarios, the validation phase focuses on confirming that the emulator correctly reconstructs missing transport data [14,15]. This phase specifically examines the emulator's ability to maintain spatial consistency, logical route progression, and statistical alignment with real-world maritime transport data. The validation process involves comparing gener-

ated data with historical vessel movement datasets, evaluating deviation metrics, and ensuring that the emulator consistently reproduces route characteristics under different parameter settings.

The validation of the proposed Emulator is essential to ensure that the generated location data accurately reflects real-world maritime transport patterns. Unlike conventional interpolation techniques or stochastic modeling approaches, which often struggle to capture dynamic environmental variations, the emulator dynamically adjusts its output based on voyage-specific parameters. Therefore, a structured validation process is necessary to assess the accuracy, consistency, and practical applicability of the generated data under different maritime conditions.

In this study, the Google Maps service was utilized to verify the routes of the generated location data [16]. Table 3 presents the results calculated based on a total distance of 993,000 m. The location data generated and stored in the database, as shown in Table 3, were used to reflect the routes in Figure 7. The visualized route, based on the inserted location data, is displayed in Figure 8. This integration demonstrates how the generated location data can be effectively mapped and monitored using Google Maps, providing a clear and accurate representation of the maritime transport paths.

Table 3. Data for Generating Location Data.

Total Route Distance (km)	Total Time (minute)	Collection Cycle	Number of Collected Data	Average Speed (km/h)	Distance Between Location (km)
993,000	150	15	10	0.0137	99

collected_time	latitude	longitude	speed
2024-10-14 15:10:00	33.038139639165017	151.56517085652217	0.0132667
2024-10-14 15:20:00	30.659686172148806	153.75750266255255	0.0132667
2024-10-14 15:30:00	29.121625191173898	153.76094468824158	0.0132667
2024-10-14 15:40:00	28.60073684148079	153.7211077070785	0.0132667
2024-10-14 15:50:00	28.139059312518384	153.7923966333563	0.0132667
2024-10-14 16:00:00	27.672606812764183	153.85071	0.0132667
2024-10-14 16:10:00	27.11402629153266	153.3388437888523	0.0132667
2024-10-14 16:20:00	26.72114	152.8547729812981	0.0132667
2024-10-14 16:30:00	27.359841	153.214518	0.0132667
2024-10-14 16:40:00	27.359901	153.178906	0.0132667

Figure 7. Example of Database New Location Data.

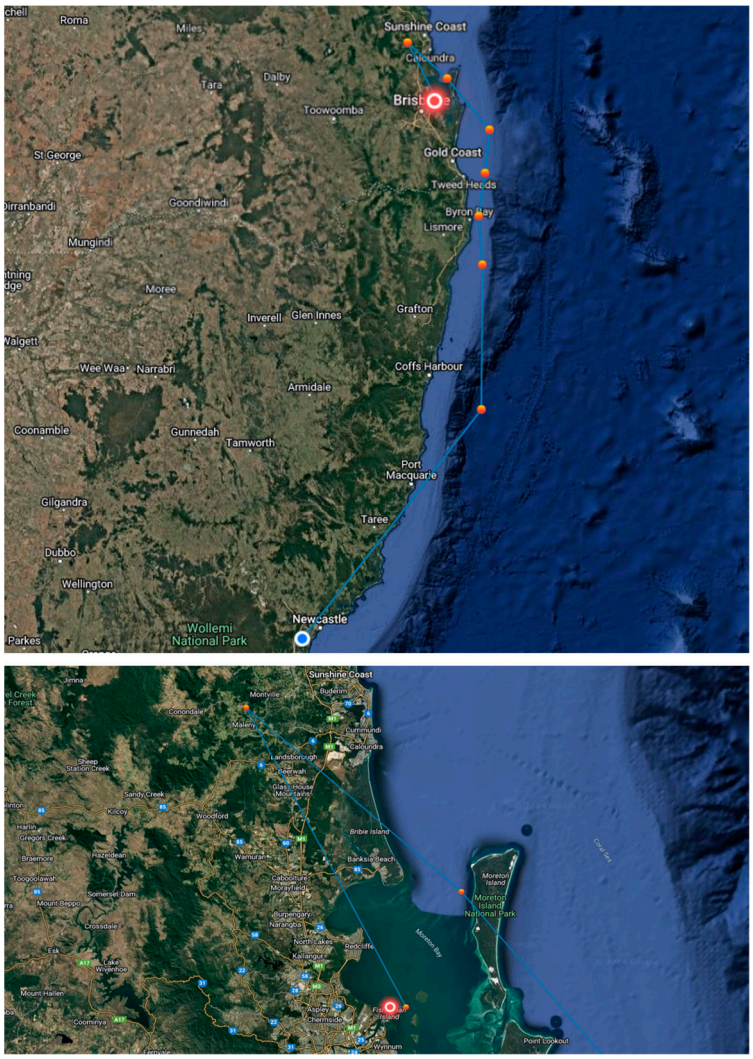


Figure 8. Monitoring of Location Data Generated Based on Table 3 and Figure 7.

4. Experiment

To evaluate the Emulator's performance, experiments were conducted using maritime transport route data, specifically from the Sydney to Brisbane port route [17]. Figure 9 illustrates the standard service route obtained from the HMM website. Table 4 provides the location data range and total route distance used in the experiments. This validation demonstrates the Emulator's applicability in real-world maritime transport scenarios.

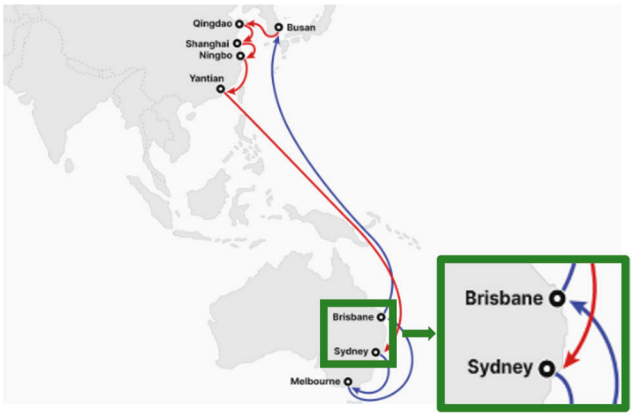


Figure 9. Sea route in HMM.

Table 4. Location Data Information for a Standard Route.

Latitude	Longitude	Distance of a Standard Route
26.7211~34.0274	151.2145~153.8507	993,000 km

In this study, experiments were conducted using the standard route shown in Figure 9, dividing the scenarios into three cases, as outlined in Table 5. The results of these experiments are presented in Figures 10–12. Among these, Figure 10 illustrates the same conditions as Figure 8, but it demonstrates that the generated location data points have different coordinates. Figures 11 and 12 show the results of a larger number of location data points being generated. The figure below highlights the ability of the Emulator to generate more detailed datasets under various conditions, demonstrating the flexibility and scalability of the proposed approach.

Table 5. Experimental Data from Emulator.

Total Time (minute)	Collection Cycle	Number of Collected Data	Average Speed (km/h)	Distance Between Location (km)
150	15	10	0.01327	99
500	25	20	0.00796	49
900	30	30	0.00663	33

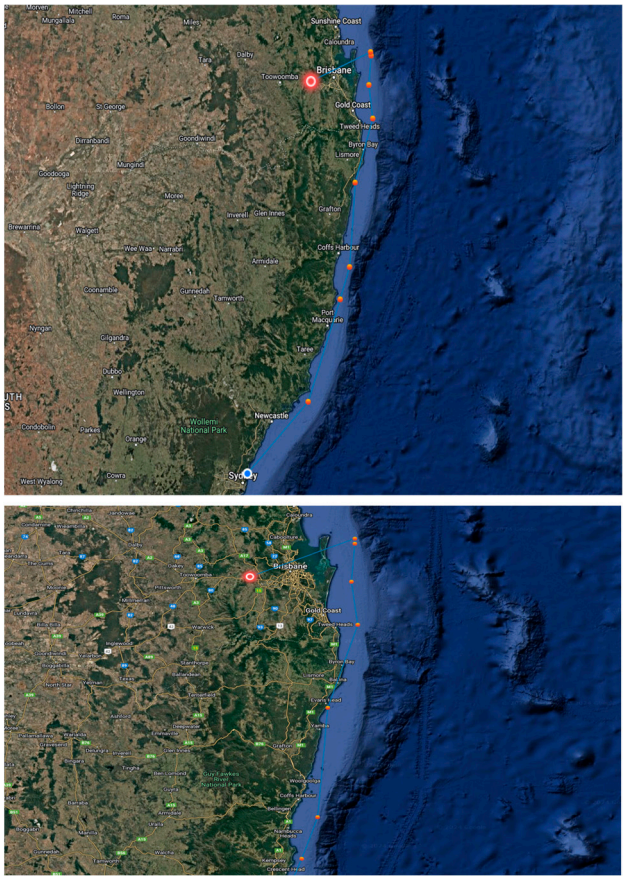


Figure 10. First Experiment (Number of Collected Data = 10).

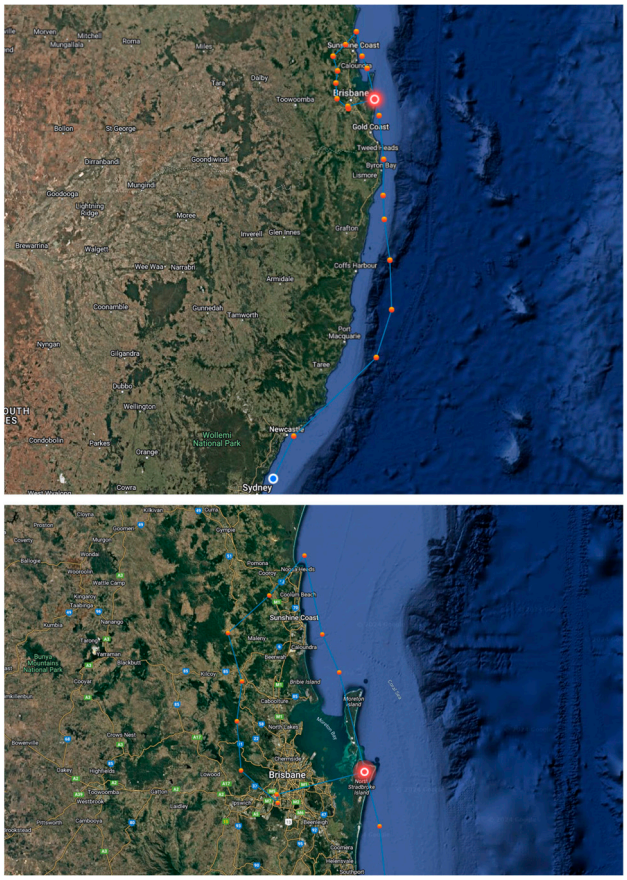


Figure 11. Second Experiment (Number of Collected Data = 20).

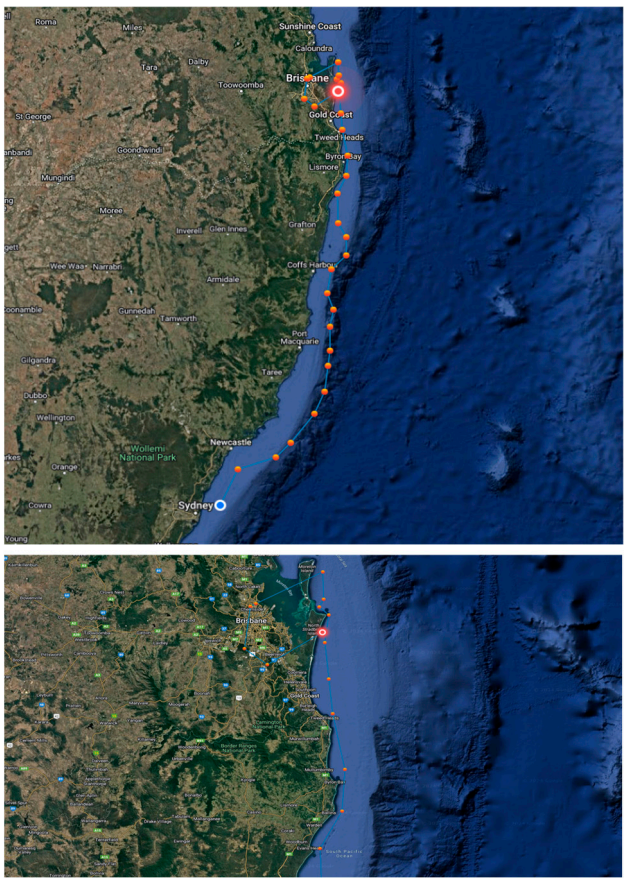


Figure 12. Third Experiment (Number of Collected Data = 30).

To assess the accuracy of the maritime transport emulator, the coefficient of determination (R-Square, R^2) was employed. The R^2 metric quantifies how well the generated location data aligns with actual voyage data, serving as an indicator of the Emulator's effectiveness in reconstructing missing transport data. The experiments were conducted using a maritime transport route between Sydney and Brisbane, with different numbers of generated data points for comparative evaluation. Table 6 presents the results of the R-Square analysis for latitude and longitude data generated by the emulator under varying conditions.

$$R^2 = \frac{\sum(\hat{Y} - \overline{Y})^2}{\sum(Y - \overline{Y})^2} \tag{8}$$

Table 6. Precision of Generated Location data.

Latitude	R^2	Longitude	R^2
Figure 10	0.8627	Figure 10	0.8527
Figure 11	0.9028	Figure 11	0.9089
Figure 12	0.9458	Figure 12	0.9478

Through the experiments described above, the proposed Emulator algorithm demonstrated its ability to generate new location data using collected data points. By comparing Figures 10 and 12, it was observed that even under identical conditions; the algorithm produces different location data, resulting in varied route generation [18]. Additionally, as the number of data-generated points increases, the resulting routes become more detailed and clearly defined. This indicates that the algorithm is highly effective not only for the experimental route but also for generating diverse routes when approximate location data of a standard route is available. The proposed Emulator is designed to generate new location data based on the specific requirements of users, and its effectiveness has been validated through the experiments conducted in this study [19].

Additionally, we analyzed the accuracy between the existing route location data and the Emulator-generated location data. The results from the experiments presented in Figures 10–12 indicate that the Emulator consistently demonstrated a high level of accuracy, confirming its effectiveness. The experimental results demonstrate a clear trend: as the number of generated data points increases, the R-Square values improve correspondingly. Specifically, the Emulator achieves an accuracy of 86.27% and 85.27% in latitude and longitude, respectively, when generating 10 data points. When the data density is increased to 30 points, the Emulator exhibits a much higher accuracy, reaching 94.58% for latitude and 94.78% for longitude. These results indicate that the Emulator is highly effective in reconstructing maritime transport data with strong spatial consistency. The increasing R-Square values suggest that the Emulator's data generation method aligns well with actual vessel movement patterns, ensuring that the synthetic data accurately represents real-world maritime routes. Furthermore, it was observed that as the number of generated data points increased, the accuracy values also improved, suggesting a proportional relationship between the amount of generated data and accuracy. However, further research is required to validate this relationship more comprehensively, and additional studies will be conducted in future work.

For the validation of the proposed Emulator, the maritime transport route between Sydney and Brisbane ports was selected as the test scenario. This route was designated by the shipping company participating in the research project, serving as a representative maritime corridor for Emulator validation. The selection was based on operational considerations and practical applicability within the project scope, ensuring that the Emulator's performance could be evaluated under realistic maritime transport conditions. While this

study focuses on a specific route, the proposed approach is designed to be adaptable to various shipping lanes globally.

5. Conclusions

Maritime IoT data is prone to loss due to various factors such as equipment malfunctions and adverse weather [20,21], often leading to irreversible data gaps that hinder service delivery and operational efficiency. To address this issue, this study proposes an Emulator that reconstructs missing transport data using historical maritime data and voyage-specific parameters. By allowing users to input key variables such as 'voyage duration' and 'data collection interval', the Emulator automatically generates supplementary location data, ensuring data continuity and accuracy.

Compared to previous studies that relied on random-matching techniques or simple statistical estimations, this study introduces a more structured and adaptive approach. A prior study utilized standard deviation calculations from collected coordinates to generate location data. While effective in producing additional data points, this method lacked trajectory preservation and real-world navigational accuracy, resulting in inconsistencies in reconstructed maritime routes. In contrast, the proposed Emulator enhances spatial consistency by employing trigonometric calculations to dynamically reconstruct missing data, ensuring smooth and natural vessel movement patterns. Furthermore, by integrating user-defined voyage parameters, this study allows greater control over data generation, thereby improving adaptability to various maritime conditions.

The proposed Emulator has significant practical applications in the maritime logistics sector. Shipping companies can utilize it to enhance route analysis and operational planning, while port authorities can integrate the generated data into digital twin systems for real-time vessel movement simulations, improving berth scheduling efficiency. Additionally, marine insurers can assess risk profiles more effectively by analyzing reconstructed voyage data, leading to more accurate insurance premium calculations [22,23]. By ensuring seamless maritime data continuity, this approach enhances decision-making and contributes to safer and more cost-effective global shipping operations.

Despite its effectiveness, the Emulator has certain limitations. This study primarily validated the approach using a specific maritime transport route, necessitating further testing across diverse geographical regions and shipping conditions [24,25]. Moreover, while the Emulator successfully generates supplementary data, real-time adaptability and responsiveness to dynamic environmental changes require further investigation. Future research will focus on integrating machine learning techniques to improve predictive modeling, enhance real-time processing capabilities, and expand the dataset to encompass global shipping routes. These advancements will further strengthen the Emulator's applicability in maritime logistics, ensuring more reliable and adaptive data-driven solutions.

Author Contributions: Conceptualization, C.-R.P. and D.-M.P.; methodology, D.-M.P.; software, C.-R.P. and D.-M.P.; validation, B.-K.P. and T.-H.K.; investigation, C.-R.P.; writing—original draft preparation, C.-R.P.; writing—review, and editing, B.-K.P., B.O.K., D.-M.P. and T.-H.K.; visualization, C.-R.P.; supervision, B.-K.P., B.O.K. and D.-M.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Dong-A University research fund.

Data Availability Statement: All experimental data are available on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Salah, K.; Alfalasi, A.; Alfalasi, M.; Alharmoudi, M.; Alzaabi, M.; Alzyeodi, A.; Ahmad, R.W. IoT-Enabled Shipping Container with Environmental Monitoring and Location Tracking. In Proceedings of the 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 10–13 January 2020; pp. 1–6.
2. Mahmood, S.; Hasan, R.; Ullah, A.; Sarker, K.U. SMART Security Alert System for Monitoring and Controlling Container Transportation. In Proceedings of the 2019 4th MEC International Conference on Big Data and Smart City (ICBDSC), Muscat, Oman, 15–16 January 2019; pp. 1–5.
3. Kim, T.H.; Choi, S.P.; Moon, Y.S.; Lee, B.H.; Jung, J.W.; Park, B.K.; Kim, J.J.; Choi, H.R. A Study on the Container Monitoring Loaded into the Hold in Maritime Logistics. *J. Korean Inst. Commun. Inf. Sci.* **2016**, *41*, 1446–1455. [CrossRef]
4. Strengthen Marine Data Collection, Sharing, and Analysis System. Available online: https://www.hankyung.com/article/2023082336081 (accessed on 15 January 2025).
5. KIMST of Oceans and Fisheries Marine and Fisheries Big Data Platform. Available online: https://www.vadahub.go.kr/user/main/index.do (accessed on 5 January 2025).
6. DSME Information Systems-Allsea Data Shipbuilding and Maritime Big Data System Development. Available online: www.cnbnews.com/news/article.html?no=435511 (accessed on 4 January 2025).
7. Moon, Y.S.; Choi, H.R.; Lee, E.K.; Jo, S.J.; SJ, R.; Choi, S.P. Develop of Container Security Device (Contracer) using M2M technology. In *Proceeding of the Korea Institute of Communication Sciences*; The Korean Institute of Communications and Information Sciences: Seoul, Republic of Korea, 2010; pp. 520–521.
8. BigdataSea, Data Utilization Case Studies. Available online: https://www.bigdata-sea.kr/dataanalysis/datacoa/datacoaList.do (accessed on 7 January 2025).
9. Park, C.R.; Kim, T.H.; Lee, E.K. Development of IoT Sensor Data Generation Emulator for Smart Marine Logistics. *J. Trans. Korea Inf. Process. Soc.* **2024**, *13*, 551–558.
10. Kim, T.H.; Jung, J.W.; Park, D.M.; Kim, D.W.; Park, B.K. Development of Smart Container for Real-time Monitoring of Global Maritime Logistics. *J. Korea Inst. Inf. Commun. Eng.* **2023**, 1449–1457. Available online: https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE11611808 (accessed on 17 March 2025).
11. Robusto, C. The cosine-haversine formula. *Am. Math. Mon.* **1957**, *64*, 38–40.
12. Australian Transport Safety Bureau (ATSB). *Loss of Containers Overboard from YM Efficiency off Newcastle, New South Wales, on 1 June 2018*; ATSB Transport Safety Report MO-2018-008; Australian Transport Safety Bureau: Canberra, Australia, 2020. Available online: https://www.atsb.gov.au/sites/default/files/media/5777329/mo-2018-008_final.pdf (accessed on 17 March 2025).
13. Containers Lost at Sea—2023 Update. World Shipping Council Releases Containers Lost at Sea Report—2023 Update—World Shipping Council. Available online: https://www.worldshipping.org/news/world-shipping-council-releases-containers-lost-at-sea-report-2023-update?utm_source=chatgpt.com (accessed on 17 March 2025).
14. Jeon, H.M.; Kim, H.N. Analysis on Potential Benefits of Smart Containers. *Ocean Polar Res.* **2023**, *38*, 109–127. Available online: https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE11714248 (accessed on 17 March 2025).
15. Kim, H.W.; Choi, H.R.; Kim, C.S.; Kim, J.J.; Park, D.M.; Jeon, M.J. A Study on The Operation Strategy for Smart Container and Smart Port Linkage. In Proceedings of the KICS Fall Conference 2021, Yeosu, Republic of Korea, 17–19 November 2021; pp. 751–752.
16. HMM. Available online: https://www.hmm21.com/ (accessed on 14 January 2025).
17. Moon, Y.S.; Choi, S.P.; Lee, E.K.; Kim, J.J.; Choi, H.R. Performance Evaluation of Advanced Container Security Device (ACSD) system based on IoT (Internet of Things). *J. Korea Inst. Inf. Commun. Eng.* **2013**, *17*, 2183–2190.
18. ORBCOMM Smart Dry Container Telematics, Dry Shipping Container Telematics for Cargo Visibility, Security and Integrity. Smart Dry Container Telematics | ORBCOMM. Available online: https://www.orbcomm.com/en/solutions/maritime/hardware/ct-1000 (accessed on 18 January 2025).
19. Hwang, D.J. A Discussion on Container Loss Accidents and Responses During Ship Voyage. *J. Navig. Port Res.* **2022**, *46*, 331–337.
20. THELOADSTAR. 2018.10.15.. MSC Opts for Traxens Technology to Create 50,000 'Smart' Containers. Available online: https://theloadstar.com/msc-opts-traxens-technology-create-50000-smart-containers/ (accessed on 23 January 2025).
21. Jing, W.; Zhao, X.; Deng, Y.; Xu, S.; Li, J. Research on key technologies and standard framework of secure and smart container. In Proceedings of the 2016 IEEE Trustcom/BigDataSE/ISPA, Wuhan, China, 23–26 August 2016; pp. 2063–2067.
22. UNECE. Trade Facilitation White Paper on Smart Containers. 2019. Available online: https://unece.org/trade/publications/trade-facilitation-white-paper-smart-containers-ecetrade446 (accessed on 1 February 2025).
23. Global Infrastructure Hub, Smart Container. Available online: https://www.gihub.org/infrastructure-technology-use-cases/case-studies/smart-shipping-containers-to-enhance-port-productivity/ (accessed on 1 February 2025).

24. NEXXIOT, Intelligence for Every Cargo Journey. Edge—Nexxiot. Available online: https://nexxiot.com/?utm_source=chatgpt.com (accessed on 4 February 2025).
25. TRAXENS. 2019.5.27. A.P. Moller—Maersk Will Join Traxens. Available online: https://www.traxens.com/press/a-p-moller-maersk-will-join-traxens (accessed on 8 February 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.