


## Article

# Trajectory Data Compression Algorithm Based on Ship Navigation State and Acceleration Variation

Junbo Gao \*, Ze Cai, Wangjing Yu and Wei Sun 

College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China

\* Correspondence: jbgao@shmtu.edu.cn

**Abstract:** An active area of study under the dual carbon target, which is based on automatic identification systems (AIS), is the emission inventory of pollutants from ships. Data compression is required because there is currently so much data that it has become difficult to transmit, process, and store it. A trajectory simplification method considering the ship sailing state and acceleration rate of change is developed in this paper to assure the validity of the compressed data used in the emission inventory analysis. By carefully examining the integral relationship between acceleration and pollution emissions, the algorithm constructs an acceleration rate of change function for data compression and categorizes AIS data by ship navigation status. By dynamically altering the amount of acceleration change, the developed function can stabilize the pollutant emission calculation error and adaptively calculate the threshold value. The experimental results show that the emission calculation error of the proposed algorithm is only 0.185% when the compression rate is 90.28%.

**Keywords:** automatic identification systems; data compression; emission inventories; adaptive thresholds



**Citation:** Gao, J.; Cai, Z.; Yu, W.; Sun, W. Trajectory Data Compression Algorithm Based on Ship Navigation State and Acceleration Variation. *J. Mar. Sci. Eng.* **2023**, *11*, 216. <https://doi.org/10.3390/jmse11010216>

Academic Editor: Raúl Guanche García

Received: 24 November 2022

Revised: 3 January 2023

Accepted: 10 January 2023

Published: 13 January 2023



**Copyright:** © 2023 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

Under the two key objectives of “carbon peaking” and “carbon neutrality”, the emission inventory of pollutants from ships is currently a hot research area. The process of compiling the inventory is based on the automatic identification system (AIS). A new navigation aid called AIS is being used to improve marine safety and communication between ships and shore, as well as between ships themselves. It can automatically communicate crucial data, including the ship’s position, speed, heading, and name. When carbon dioxide emissions reach their peak and then start to decline gradually, this is referred to as “carbon peaking”. “Carbon neutrality” refers to the positive and negative offsetting of carbon dioxide or greenhouse gas emissions through energy conservation and emission reduction strategies. The production of emission inventories has been achieved at numerous ports [1–4]. A great deal of AIS data is produced due to the skyrocketing volume of maritime activity, which presents significant challenges for data transmission and processing [3]. The data collected by researchers is probably already compressed to aid in transmission. Additionally, the cost in terms of time and space needed to perform computations for pollutant emission increases with data volume. To increase the effectiveness of emission inventory investigations, large amounts of data must be compressed before analysis. Compressed data can free up storage space and make it easier to store and transmit trajectory information [5]. More importantly, the ship’s trajectory data may be thoroughly analyzed with the help of simplification, allowing it to retain pertinent information and eliminate superfluous material.

The use of trajectory simplification and compression techniques has greatly improved due to the rapid development of many disciplines and the widespread application of these techniques in a variety of sectors. Early methods for simplifying trajectories generally took into account information such as position, velocity, and time [6–10]. Douglas proposed the Douglas–Peucker (DP) algorithm in 1973, which is one of the most classical trajectory

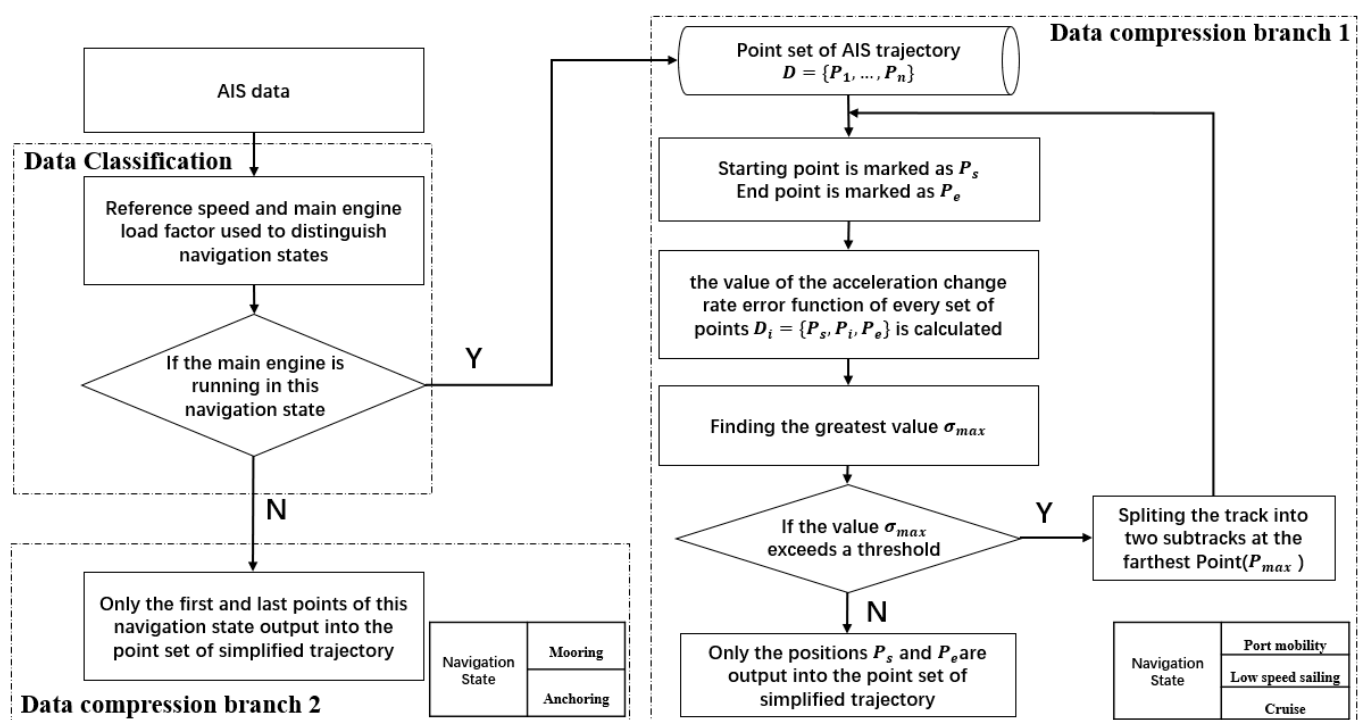
compression algorithms [6]. Meratnia et al. proposed the velocity-based top-down algorithm and top-down time ratio (TD-TR) algorithm [7]. Many researchers have improved the DP algorithm by considering the characteristics of AIS data [11–16]. Li et al. proposed that a suitable threshold interval can be selected from the experimental comparison results of different DP thresholds, according to the quality of AIS trajectory visualization [11]. Han et al. proposed the conversion of trajectories into spatial paths and time series to compress both spatial and temporal data [12]. Liangbin Zhao and Guoyou Shi proposed a method based on an improved DP algorithm that considers the shape of the ship's trajectory derived from the heading information of the trajectory points [16]. Wonhee Lee and Sung-Won Cho (2022) proposed a simplified algorithm for the AIS trajectory considering terrain information [17]. The polygon map random (PMR) quadtree was used to consider topographic information on the coast, and the intersection between topographic information and simplified trajectories was efficiently computed using the PMR quadtree. These algorithms consider other characteristics of the ship, but do not apply to emission inventory studies because the production of emission inventories requires the consideration of the ship's engine information and the deep relationship between different characteristics of the ship, and it is not enough to consider only these shallow characteristics [18]. The lack of targeted studies makes it impossible to guarantee the reliability of the data. The processing efficiency of massive data is equally important, so the selection of the threshold value is also the focus of current research, and an adaptive threshold can optimize the compression method to a great extent [19–25]. Zhaokun Wei et al. designed a new algorithm considering trajectory space and motion features which can compress AIS trajectories based on ship behavior features and apply statistical theory to help determine the threshold of motion features in the sliding window algorithm [20]. Chunhua Tang et al. proposed an adaptive threshold AIS trajectory data compression method based on the DP algorithm to improve the computational efficiency of the algorithm by taking advantage of matrix operations and reducing the number of points [21]. Ran Yan et al. proposed two trajectory compression algorithms: a static mode with a preset compression threshold and a dynamic mode that considers the distance between the trajectory point and the coastline in real-time [22]. To address the difficulties involved in selecting appropriate thresholds, adaptive thresholds are also included in this paper's design goal.

Despite the high computing performance of these techniques, they are not appropriate for the analysis of emission inventories. This is due to the bottom-up emission inventory production method's requirement that different parameter values be substituted based on the type and condition of the ship's sailing [26–29]. One of the crucial metrics, main engine load, must be calculated using both real-time speed and rated speed. As a result, in addition to position and speed information, it is important to consider the complex relationship between the motion characteristics of the ship and the pollution emissions when compressing such data. When employed for emission estimates, the compressed data output from the current trajectory simplification method will result in significant error. Therefore, a trajectory simplification technique that can be used for ship-related pollution emissions is required. Based on the peculiarities of AIS data and emission inventories, an adaptive threshold simplification algorithm suitable for emission inventories is proposed in this study. This study offers three contributions. To retain the voyage state differentiation points as the important features and speed up the compression process, the data are first categorized and then simplified. Second, a function for the acceleration rate of change that may be adaptively decided as a threshold was built. This function combined the main engine load and the rated speed to thoroughly assess the overall relationship with pollutant emissions. The suggested algorithm is then contrasted with other algorithms in terms of running time, compression ratio, and pollutant emission calculation error.

## 2. Ship Trajectory Simplification Algorithm

In this research, a simplified algorithm is put forth that can guarantee a high compression rate while maintaining the accuracy of emission calculation and critical feature

information, including latitude and longitude, real-time speed, and the acceleration of the ship. Figure 1 depicts the simplified algorithm flow, and Appendix A contains the pseudocode. The simplified algorithm is split into two halves. The data are categorized in the first part according to the sailing state, while retaining the characteristics of the sailing state. The main engine load and speed determine the sailing state, and the ship's trajectory exhibits noticeably varied features depending on the sailing state. For instance, the ship is virtually completely stationary when it is moored, whereas when it is cruising, the ship is primarily moving across the water. The crucial trajectory information is thus contained in the navigation state differentiation point. The data from the various navigation states are handled independently in the second part, and the trajectories are simplified by adaptive thresholding. The several navigation statuses are categorized and compressed independently in this section. The compression technique can be significantly improved with adaptive thresholding.



**Figure 1.** Framework diagram of the simplified algorithm. First, the AIS data are classified according to the navigation status, and only the first and last trajectory points of the trajectory are retained for the part of the data where the main engine is not running (see data compression branch 2). Then, the maximum value  $\sigma_{\max}$  of the acceleration rate of change function in all intermediate trajectory points  $P_i$  is calculated for the part the data where the main engine running, and the result is compared with the set threshold value to determine whether to retain or delete it (see data compression branch 1).

### 2.1. Classification of Data According to Navigation Status

Data must first be categorized according to the sailing status before being compressed. To determine the ship's sailing status, the IMO's speed and host load factor recommendations from the fourth GHG study are combined. The methodology is illustrated in Table 1 [26]. It is possible to determine the distinguishing speed of the relevant sailing condition by using the host load factor calculation formula. Additionally, because not all differentiating points of the navigation status may be recorded by the automatic identification system, interpolation must be used to determine some of these time points. It is important to interpolate between these two trajectory points to discern between distinct sailing states when two neighboring trajectory points are in different sailing states.

**Table 1.** Basis of the determination of vessel navigation status.

Navigational States	Judgment Conditions
Mooring	Speed < 1 knot
Anchoring	1 knot ≤ speed < 3 knot
Port mobility	Speed ≥ 3 knot and main engine load < 20%
Low speed sailing	Speed ≥ 3 knot and 20% ≤ main engine load < 65%
Cruising	Main engine load ≥ 65%

For trajectory points in different navigational states  $P_i$  and  $P_{i+1}$ , in  $P_i$ , the real-time speed is  $V_i$ , and the time point is  $t_i$ . In  $P_{i+1}$ , the real-time speed is  $V_{i+1}$ , and the time point is  $t_{i+1}$ . The distinguished speeds for different sailing states are  $V$ . When  $P_i$  to  $P_{i+1}$  has a stable rate of speed change during this time, the speed ratio can be calculated from  $r$ , and the interpolation point is calculated from  $P$  of the time point  $t$ .

For trajectory points  $P_i$  and  $P_{i+1}$  of indifferent navigational states, the velocity of  $P_i$  is  $V_i$ , and the time point is  $t_i$ . The voyage speed of  $P_{i+1}$  is  $V_{i+1}$ , and the time point is  $t_{i+1}$ . The distinguished velocity for different navigational states is  $V$ . When the period from  $P_i$  to  $P_{i+1}$  has a stable rate of change of velocity, the time point  $t$  of the interpolated point  $P$  can be calculated from the velocity ratio  $r$ .

$$r = \frac{(V - V_i)}{(V_{i+1} - V_i)} \quad (1)$$

$$t = r \times (t_{i+1} - t_i) + t_i \quad (2)$$

Table 1 divides the ship's sailing state into five categories, where the ship is almost stationary, and the main engine is not running in the moored state [26,27]. The trajectory of the ship in the other four states will change significantly, and the main engine will run; this part of the data is also the focus of trajectory simplification. Therefore, in this paper, the sailing states are grouped into two parts according to whether the main engine is running or not. When the main engine is not running, only the first trajectory point and the last trajectory point of this part of the data need to be retained. When the main engine is running, the trajectory data of this part is simplified by the adaptive threshold value designed in this paper.

## 2.2. Adaptive Thresholds

The applicability of the threshold value to the data source in compression algorithms determines whether the compressed data may be used for further analysis [23,24]. Most modern data compression algorithms demand compression criteria that have been intentionally defined. A great deal of testing is required to achieve the correct threshold value because this is a blind, speculative operation. Effective compression can be increased by adaptive thresholding. Three aspects make up the adaptive thresholding concept presented in this study. The integral relational equation between pollutant emission and acceleration is first developed after the key variables portion of the host emission equation is extracted for in-depth analysis. Second, an acceleration rate of change function is built using the integral relationship equation. This function not only reflects the accuracy of the integral relationship's emission computation, but also allows for dynamic adjustment of the acceleration change at various speed intervals. Finally, the function is used to establish a threshold value for trajectory simplification, which is an adjustable parameter and a user-preset for the accuracy of emission calculation.

The trustworthiness of the compressed data in subsequent specific research cannot be guaranteed by many trajectory simplification techniques. They can only promise that the retained information has a high trajectory similarity. The adaptive threshold suggested in this paper can ensure that the quality of the compressed data is no longer unknown. Users can adjust the threshold value to achieve a balance between the compression rate and the

data quality according to the required precision, without performing multiple experiments. This lowers the cost of compression and ensures the dependability of the compressed data.

### 2.2.1. Integral Relationship Equation

Even though a ship's track is continuous, AIS data is collected and stored discretely [30–32]. It is necessary to estimate the continuous variation of each ship's characteristics to calculate pollution emissions. If the mean value approach is used to estimate the velocity variation between two trajectory points, the error will be higher the larger the observed velocity difference between the two trajectory points. The cost will rise once more if the trajectory is interpolated with high density during this period. If the velocity information for this period is calculated using integration, it is not only more accurate than the mean technique, but also more efficient than high-density trajectory interpolation.

When calculating emissions using discrete AIS data, it is critical to determine the continuous variation of each parameter of a ship. If the deep relationship between parameter variation and emission calculation can be found, the ship trajectory data can be simplified to the maximum extent while ensuring the accuracy of emission calculation. The main engine emission estimation model in emission inventory production is shown in Equation (1) [26]. In the equation,  $E_i$  stands for the emissions of the main engine for a class  $i$  specific pollutant,  $P$  stands for rated engine power,  $LF$  stands for main engine load factor,  $Act$  stands for operation time,  $EF$  stands for pollutant emission factor,  $FCF$  stands for fuel correction factor,  $LLA$  stands for low load adjustment factor, and  $s$  stands for the sailing state of the ship. Among these, the main engine load factor must be calculated separately, and it is an important factor affecting the accuracy of the emission calculation. The classical calculation formula of the main engine load factor is shown in Equation (2).  $V_a$  is the real-time speed, and  $V_m$  is the maximum design speed.

$$E_i = \sum_{S=1}^5 \left( P \times LF_S \times Act_S \times EF_{S,i} \times FCF \times LLA \times 10^{-6} \right) \quad (3)$$

$$LF = \left( \frac{V_a}{V_m} \right)^3 \quad (4)$$

Let the velocity of the trajectory point  $P_1$  be  $V_1$ , the velocity of  $P_2$  be  $V_2$ , the time difference be  $Act$ , and the rate of change of velocity during this period be  $a$ . After determining the ship's main engine power and sailing state, extract the variable part of the main engine emission calculation equation  $LF \times Act$ . The integral transformation is then carried out. The formula for calculating the main engine emission is shown in Equation (1), and the integral calculation relationship is shown in Equations (5) and (6).

$$a = \frac{V_2 - V_1}{Act} \quad (5)$$

$$Act \times LF = \frac{Act \times V_a^3}{V_m^3} = \frac{\int_0^{Act} (V_1 + a \times t)^3 dt}{V_m^3} \quad (6)$$

Equation (6) converts the formula based on real-time velocity and time difference into an integral relationship based on acceleration. If the acceleration of a segment of the trajectory is stable, the intermediate trajectory points can be discarded without losing critical information and again, without affecting the emission calculation.

### 2.2.2. Threshold Function of Acceleration Rate of Change

It is cumbersome to design the threshold function to compress the data directly using the emission calculation formula, and this study considers simplifying the process with the help of Equations (5) and (6). For the starting point  $P_s$ , the endpoint  $P_e$ , and the

intermediate trajectory points  $P_i$ , the emissions can be calculated for three different periods  $E_{s,i}$ , the  $E_{i,e}$  and  $E_{s,e}$ , and the errors can be analyze.

$$C = E_{s,e} - (E_{s,i} + E_{i,e}) \quad (7)$$

$$\sigma'_E = \frac{C}{E_{s,e}} \times 100\% \quad (8)$$

In the above equation, the  $\sigma'_E$  is the error in  $E_{s,e}$  as the standard error.

It is more complicated to calculate the error if the emission calculation formula is used directly, due to the need to substitute all the parameters in Equation (3). When the acceleration of the three time periods is determined, the  $C$  is equal to a constant value that is not affected by the magnitude of the real-time velocity. When the acceleration change is constant,  $E_{s,e}$  and  $\sigma'_E$  in Equation (8) have the opposite trend, while  $E_{s,e}$  and real-time velocity show the same trend. Therefore, the main influence on the emission error is the variation of the real-time velocity over a certain period, which can be expressed in terms of acceleration. Since using a constant acceleration change to set the threshold leads to different simplification effects for the data of high velocity and the data of low velocity, the adaptive adjustment of the acceleration change at different velocity intervals is also required when setting the threshold function. In this study, Riemann integral relations for three accelerations can be established with the help of Equations (5) and (6).

$$\int_0^{\text{Act}_{s,i}} (V_s + a_{s,i} \times t) dt + \int_0^{\text{Act}_{i,e}} (V_i + a_{i,e} \times t) dt = \int_0^{\text{Act}_{s,e}} (V_s + a_{s,e} \times t) dt \quad (9)$$

$$\frac{V_s + V_i}{2} \times \text{Act}_{s,i} + \frac{V_i + V_e}{2} \times \text{Act}_{i,e} = \frac{V_s + V_e}{2} \times \text{Act}_{s,e} \quad (10)$$

Equation (9) is the Riemann integral relation, and Equation (10) is the integral expansion, where  $\text{Act}$  denotes the period, and  $a_{s,i}$  denotes the  $P_s$  to  $P_i$  acceleration, and  $a_{i,e}$  denotes the  $P_i$  to  $P_e$  acceleration, and  $a_{s,e}$  denotes the  $P_s$  to  $P_e$  acceleration, and  $V$  is the velocity. Set  $S$  and  $S'$  as the expressions of Equations (11) and (12), and we can obtain the acceleration rate of the change function  $\sigma$ , which is shown in Equation (13).

$$S = \frac{V_s + V_e}{2} \times \text{Act}_{s,e} \quad (11)$$

$$S' = \frac{V_s + V_i}{2} \times \text{Act}_{s,i} + \frac{V_i + V_e}{2} \times \text{Act}_{i,e} \quad (12)$$

$$\sigma = \frac{|S - S'|}{S} \quad (13)$$

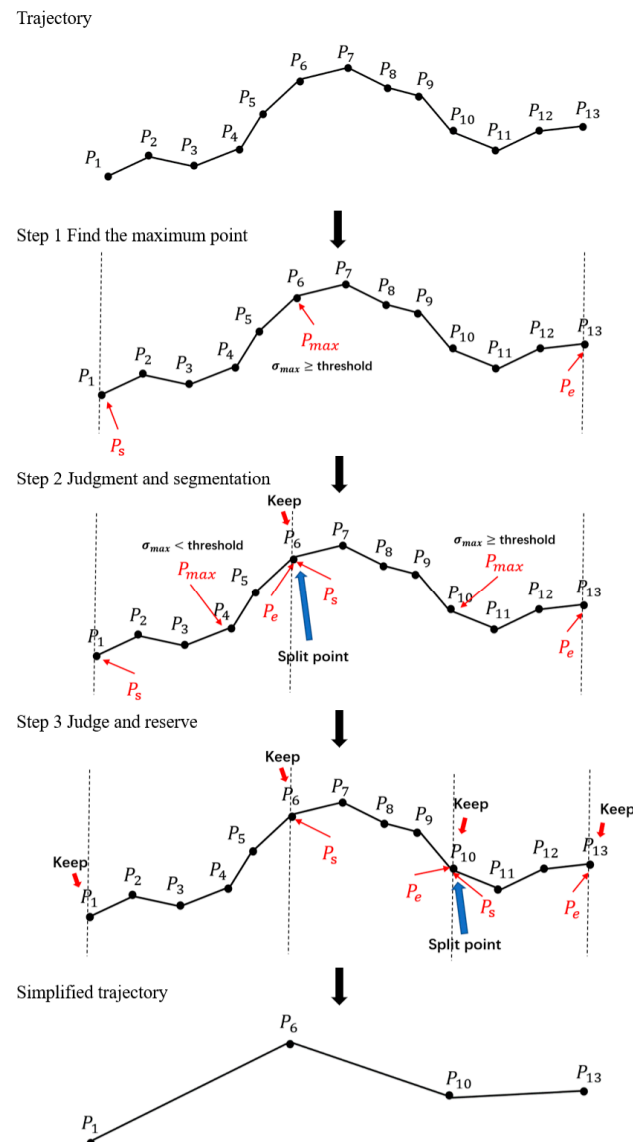
$\sigma$  reflects the fluctuation of the acceleration change  $\sigma$ ; the smaller it is, the more stable the acceleration. In addition,  $\sigma$  has the ability to adaptively change the amount of acceleration change at different velocity intervals. The smaller value also reflects the smaller error of the emission calculation. When  $\sigma$  equals 0, it means that the acceleration is constant, and the emission calculation error between the compressed data and the original data is also 0. The value of  $\sigma$  approximates the emission calculation error. Therefore, this paper uses  $\sigma$  to set an adaptive threshold for trajectory simplification, which is equivalent to presetting the emission calculation error value to ensure the quality of the compressed data, avoiding the need to determine the appropriate threshold value through extensive experiments.

### 2.3. Trajectory Simplification

According to the data classification in Section 2.1, the trajectory simplification process of AIS data in the four navigation states during host operation is described below. The AIS trajectory is represented as the set of points  $D = \{P_1, P_2, \dots, P_i\}$ . Calculate the maximum value of  $\sigma_{\max}$  as a function of the rate of change of acceleration for each point  $P_i$  on the trajectory from its starting point  $P_s$  and its ending point  $P_e$ . If  $\sigma_{\max}$  exceeds the threshold,



the maximum point  $P_{\max}$  is retained. Subsequently, the trajectory is split at that position ( $P_{\max}$ ). The algorithm is applied recursively to both sub-trajectories. If  $\sigma_{\max}$  is below the threshold, only the points  $P_s$  and  $P_e$  of the subpart of the trajectory are retained. A schematic of the trajectory simplification process is shown in Figure 2.



**Figure 2.** Schematic diagram of the trajectory simplification process. There are 13 trajectory points. In the first step, we keep the first node  $P_1$  and the last node  $P_{13}$ , and find the maximum point  $P_6$ . In the second step, if  $\sigma_{\max}$  exceeds the threshold, keep  $P_6$ , and split the trajectory. In the third step, recursively judge the two trajectories, find the maximum point, and judge  $\sigma_{\max}$ ; if it does not exceed the threshold, then discard all trajectory points except for the first and last nodes. Recursively repeat the judgment, and finally, obtain the simplified trajectory containing only four trajectory points.

#### 2.4. Compression Evaluation

In this paper, considering the needs of practical applications, the proposed algorithm pays more attention to the computational error of emissions from compressed data. The compression performance is evaluated in three aspects, namely compression ratio, emission calculation error, and runtime complexity. The compression ratio is derived by dividing the number of discarded trajectory points by the number of original trajectory points. The

emission calculation error represents the standard error between the calculated emissions from the compressed data and the calculated emissions from the uncompressed data.

$$CR = \frac{N - N_s}{N} \times 100\% \quad (14)$$

$$\sigma_E = \left| \frac{E_o - E_s}{E_o} \right| \quad (15)$$

In the above equation, CR is the compression rate, N is the number of trajectory points on the original trajectory, and  $N_s$  is the number of trajectory points on the simplified trajectory.  $\sigma_E$  denotes the error in emission calculation,  $E_o$  is the emission calculated from the original uncompressed data, and  $E_s$  is the emission calculated from the compressed data.

### 3. Experiment and Analysis

#### 3.1. Data Sources

The proposed approach is implemented and contrasted with other algorithms using one month's worth of AIS data from the sea region of the Shandong emission control area and three months' worth of data from the AIS data from the Ningbo port to further assess the algorithm's effectiveness. The static database of the ship comes from Clarkson's database and Lloyd's database, which mainly include parameters such as ship length, ship-breadth, ship depth, main engine power, auxiliary engine power, boiler power, rated speed, and ship tonnage. The dynamic AIS data includes parameters such as ship MMSI code, longitude, latitude, bow direction, heading to earth, real-time speed, and ship position accuracy. The AIS data and static database are tested after deciding on the outlier identification criteria [26,27]. The random forest model was used to fill in the missing values and outliers in the design parameters [33]. Cubic spline interpolation was used to fill in missing and anomalous values in the AIS data [34]. NOx is the pollutant type used in the calculation of emissions. The experiments make use of Python 3.9 as the programming language and PyCharm 11.0.10 as the compiler. The comparison experiments make use of an identical hardware setup and software environment. The compiler's recursion depth is set to 30,000 according to the real quantity of data for the method, necessitating recursive iteration.

#### 3.2. Experiments and Analysis

The compression rate may generally be improved by raising the threshold setting, but more information is lost in the process. Different thresholds were chosen for the suggested algorithm in this study so that they could be compared, and as shown in Tables 2 and 3; the compression rate rises when a higher threshold factor is applied. For each of the seven distinct compression rate circumstances, the discrepancy between the pollutant emissions calculated using the compressed data and the original data is minimal. This demonstrates that the suggested algorithm's data compression method may be successfully used to study emission inventories. The threshold system that directly considers the emission calculation formula and the acceleration rate of change is also used in this paper to compare compression performance. As shown in Figure 3, the errors of emission calculation for both at the same compression rate are very close. This also proves the reasonableness of the threshold design of the proposed algorithm.

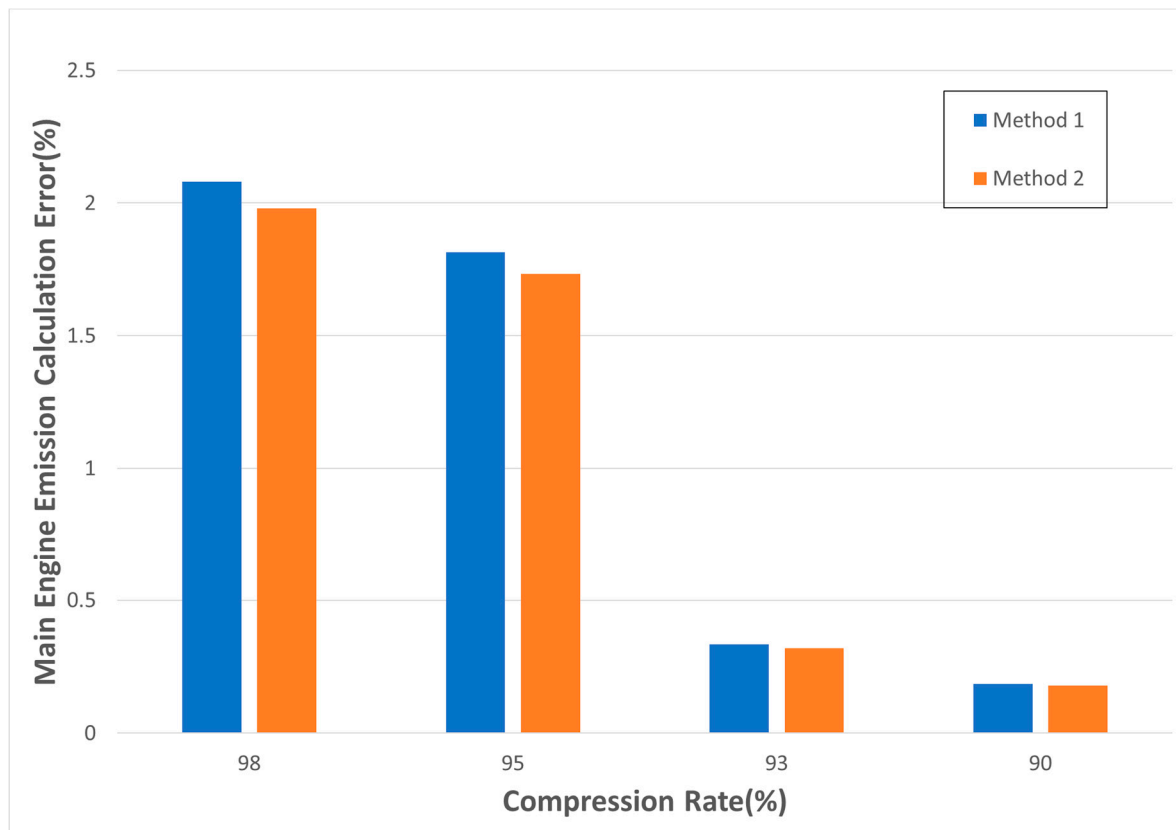


**Table 2.** Partial performance comparison of each compression algorithm at different thresholds (Shandong Province data). Seven different thresholds are set for each algorithm, corresponding to the computational error of emissions at seven compression ratios. To facilitate the cross-sectional comparison, the determined thresholds will ensure that the data compression ratios under different algorithms are as close as possible.

Proposed Algorithm			DP Algorithm			TD-TR Algorithm			CSB Algorithm		
Threshold	Compression Rate	Main Engine Emission Calculation Error	Threshold	Compression Rate	Main Engine Emission Calculation Error	Threshold	Compression Rate	Main Engine Emission Calculation Error	Threshold	Compression Rate	Main Engine Emission Calculation Error
0.01	90.28%	0.19%	0.1	90.13%	9.53%	0.1	91.14%	4.48%	1.9	91.63%	7.49%
0.05	93.48%	0.27%	0.2	94.83%	10.57%	0.2	95.84%	7.10%	2	93.63%	9.31%
0.1	94.63%	0.33%	0.5	96.49%	11.97%	0.5	96.69%	15.84%	2.1	96.44%	31.62%
0.2	95.14%	1.57%	0.8	97.55%	21.58%	0.8	97.04%	20.37%	2.2	96.84%	28.80%
0.5	97.69%	1.81%	1	97.79%	22.27%	1	97.64%	22.26%	2.3	98.49%	20.63%
0.8	97.89%	2.08%	2	98.79%	62.95%	2	98.34%	36.94%	2.4	99.44%	23.14%
1	98.14%	2.22%	5	98.89%	59.32%	5	99.34%	53.45%	2.5	99.54%	30.00%

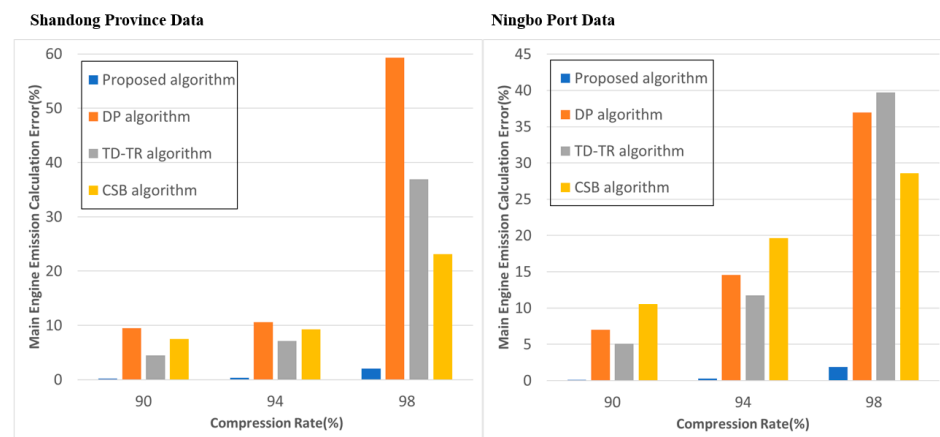
**Table 3.** Partial performance comparison of each compression algorithm under different thresholds (Ningbo Port data).

Proposed Algorithm			DP Algorithm			TR Algorithm			CSB Algorithm		
Threshold	Compression Rate	Main Engine Emission Calculation Error	Threshold	Compression Rate	Main Engine Emission Calculation Error	Threshold	Compression Rate	Main Engine Emission Calculation Error	Threshold	Compression Rate	Main Engine Emission Calculation Error
0.01	89.41%	0.12%	0.2	91.12%	7.02%	0.2	90.29%	5.08%	1.8	90.02%	10.53%
0.05	92.67%	0.23%	0.4	94.63%	14.55%	0.4	92.41%	11.76%	2	92.37%	14.45%
0.1	93.88%	0.30%	0.6	96.79%	22.61%	0.6	96.49%	19.84%	2.2	94.35%	19.64%
0.2	96.57%	1.44%	0.8	97.15%	27.37%	0.8	97.38%	23.68%	2.3	95.79%	19.26%
0.5	98.15%	1.89%	1	98.36%	36.98%	1	98.10%	39.72%	2.4	97.92%	28.57%
0.8	98.83%	2.18%	2	99.04%	49.59%	2	98.76%	47.62%	2.5	98.26%	24.98%
1	99.12%	2.49%	5	99.61%	58.19%	5	99.14%	57.03%	2.6	98.81%	33.48%



**Figure 3.** Emission calculation errors of two thresholds  $\sigma$  and  $\sigma'_E$  using methods at the same compression rate. Method 1 uses the threshold considering the acceleration rate of change, and Method 2 uses the threshold considering the standard error of emissions calculation. The emission calculation errors of the two threshold setting methods at the same compression rate are close, which also proves the reasonableness of the threshold design of the proposed algorithm.

The suggested compression technique will be contrasted with three existing compression algorithms to further assess its performance. These include the top-down time ratio (TD-TR) method [7], the Douglas–Peucker (DP) algorithm [6], and the compression algorithm considering the behavior of the ship (CSB) [20]. These three algorithms mainly consider some basic characteristics of the ship (latitude and longitude, time stamp, real-time speed, bow direction, etc.). In Appendix B, these three algorithms are explained in detail. In this study, seven sets of thresholds were chosen for each compression technique, and two datasets from the Shandong Province and the Ningbo Port were used for the experiments. Tables 2 and 3 illustrate the computational errors of the host emissions, as well as the compression rates of the four compression algorithms at various thresholds. To ensure that the four algorithms may be fully compared horizontally, the specified thresholds were set after several experiments. As shown in Figure 4, each compression algorithm was chosen for the scenarios of 90%, 94%, and 98% compression rates, while the emission calculation errors were compared horizontally. Because other algorithms have a harder time determining the precise compression rate when choosing the correct threshold, the error of the compression rate was set at within 2%.



**Figure 4.** Comparison of emission calculation errors of each compression algorithm at different compression rates. The compression rates are divided into three grades—90%, 94%, and 98%—and the four algorithms are compared horizontally. The left graph shows the comparative analysis of data from Shandong Province, and the right graph shows the comparative analysis of data from the Ningbo Port. At the compression rate of about 98%, the other algorithms have a great error in emission calculation because they lose a large amount of information concerning the ship and the emission calculation, while the proposed algorithm can still maintain a small error.

Tables 2 and 3 further illustrate how higher thresholds might result in higher compression rates and greater information loss. Each algorithm has a varied performance, as illustrated in Figure 4. The DP algorithm emission calculation error is particularly high when the compression rate is 98%, reaching 59.321% and 36.983% in the two datasets, respectively. This is so that the data can be compressed using the DP algorithm, which ignores substantial speed fluctuations in favor of position information. As a result, for various compression rates, the DP algorithm performs the poorest in this regard. The approach suggested in this research performs an order of magnitude better than do the existing algorithms, and it shows the minimum error in emission computation at various compression rates. The emission calculation error of the suggested approach still fluctuates very slightly and is only 2.221% and 1.890% in the two datasets, even when the compression rate is raised to 98%. Other algorithms with the same compression rate have emission calculation errors of more than 20%. Due to the integrated considerations of position, speed, and direction information, the emission calculation error of the compression method, taking ship behavior into account, is 23.14% at the compression rate of 98%, which is much better than that of the DP and TD-TR algorithms. Between this algorithm and the algorithm suggested in this study, there is still a sizable gap. This is due to the algorithm's additional disregard for the precise velocity fluctuation between the compressed trajectory points.

The computational error of the technique for emissions calculation is significantly reduced as the compression rate drops, which is also in line with the theory of the algorithm put forth in this study. Other compression techniques, on the other hand, follow the same pattern, but when the compression rate is too high, the emission computation error can be significant and challenging to use in emission inventory investigations. This is because other compression algorithms do not carefully consider how the motion characteristics of the ship and the pollution emissions relate to one another.

Table 4 shows the running time complexity of the four compression algorithms [35]. The algorithm proposed in this paper is divided into two parts. The first part needs to traverse all the data, with the purpose of marking and dividing the navigation state, and the time complexity of this part is  $O(n)$ . The second part needs to process the different divided sailing state data, mainly processing the data of the sailing state of the ship's main engine operation; the time complexity of this part is  $O(m \log m)$ , and  $m$  denotes the amount of data for this sailing state. The running time complexity of the DP and TD-TR algorithms depends on the different algorithm designs, which is  $O(n^2)$ , if the

dynamic sliding window approach is used, and  $O(n \log n)$ , if the iterative approach is used. The compression algorithm considering the ship's behavior is divided into two parts in parallel; the first part of the position data is compressed using the DP algorithm, so the time complexity is  $O(n^2)$  or  $O(n \log n)$ . The compression of the second part of the speed and heading data uses a fixed sliding window approach, so the time complexity is  $O(n)$ . Although the running time complexity of the second part of the proposed algorithm is not optimal among all algorithms, the division processing of the first part will reduce the amount of data processed each time. Therefore, it is possible to maintain a short running time while ensuring the superiority of the proposed algorithm.

**Table 4.** Running time complexity of each compression algorithm. The time complexity is a function that evaluates the time consumed to execute the program and allows for the estimation of program processor use. The time complexity is often expressed in large  $O$  symbolic expressions, excluding the lower order terms and first coefficients of this function. The time complexity is evaluated when the amount of input data tends to infinity. The running time complexity of the proposed algorithm is not optimal among all algorithms, but the data classification reduces the amount of data processed per iteration. Therefore, the overall efficiency is still high.

Proposed Algorithm		DP Algorithm		TD-TR Algorithm		CSB Algorithm	
Part 1	Part 2	Sliding window	Iteration	Sliding window	Iteration	Part 1	Part 2
$O(n)$	$O(m \log m)$	$O(n^2)$	$O(n \log n)$	$O(n^2)$	$O(n \log n)$	$O(n^2)$ or $O(n \log n)$	$O(n)$

The results show that the algorithm proposed in this paper can guarantee computational accuracy under the condition of a high compression rate, and it is suitable for the study of emission inventory.

#### 4. Conclusions

In this paper, we propose a trajectory data compression algorithm based on the ship's navigation state and acceleration variation, and the proposed algorithm exhibits three novelties. First, the data are classified using the navigational states, retaining the navigational state differentiation points as key features. Second, the simplified algorithm combines the main engine load and rated speed to investigate the deep relationship with pollutant emissions, and it is applicable to the study of emission inventories. Third, the simplified algorithm adaptively determines the threshold value using the acceleration rate of change function. To test the performance of the proposed algorithm, numerical experiments are employed. The results show that the proposed algorithm maintains very low emission calculation errors at high compression rates and can achieve almost the same results as the original data in the study of emission inventories. Other algorithms show high errors in emission calculations, and their compressed data are not applicable to the study of emission inventories. Compared with other algorithms, the proposed algorithm can guarantee the quality of compressed data by controlling the variation of acceleration with preset emission calculation error values, avoiding the need to determine the appropriate threshold value through extensive experiments. In addition, data classification reduces the depth of data processing iterations and improves the operational efficiency. Therefore, the proposed algorithm exhibits good comprehensive performance. Future studies may employ the distributed approach to reduce running time and may also consider the adaptive threshold in terms of compression rate [36].

**Author Contributions:** J.G.: conceptualization, software, design, analysis, writing—original draft, and reviews; Z.C.: conceptualization, structural calculations, analysis, writing—original draft, and reviews; W.Y.: conceptualization, validation, analysis, and writing—original draft; W.S.: conceptualization, validation, analysis, and writing—original draft. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Innovation Program of Shanghai Municipal Education Commission (grant no. 2021-01-07-00-10-E00121).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** The authors would like to thank the supports from the Innovation Program of Shanghai Municipal Education Commission (grant no. 2021-01-07-00-10-E00121). The authors also acknowledge the anonymous reviewers for their suggestions that improved the manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A. Pseudocode for the Proposed Compression Algorithm

The input of the algorithm is the data ( $O$ ) and the threshold, and the output is the simplified trajectory data. The data are firstly classified into  $T(NS_0)$  and  $T(NS_1)$ , according to the navigation status. The  $T(NS_0)$  data only need to keep the first and last trajectory points. The  $T(NS_1)$  data need to be compressed using the acceleration rate of the change threshold function. The process of trajectory simplification is described in detail in the form of the flow chart in Section 2.3.

---

### Algorithm A1. Compression considering acceleration

---

*Input:* Original trajectory points set  $O$ , *threshold*

*Output:* Point set *Simplified\_trajectory*

```

1:    $T(NS_0), T(NS_1) = \text{classification}(O)$ 
2:   for each  $T$  in  $T(NS_0)$  do
3:     Add  $T[0]$  into Simplified_trajectory
4:     Add  $T[n - 1]$  into Simplified_trajectory
5:   for each  $T$  in  $T(NS_1)$  do
6:      $\text{AC}(T, \text{threshold})$ 
7:   /*function*/
8:    $\text{classification}(O)\{$ 
9:   /* Classify data according to navigational states*/
10:   $n$  is the size of point set  $O$ 
11:   $\text{temp} = 0$ 
12:  for  $i = 1$  to  $n - 1$  do
13:    if  $O[i]$  is dividing point of navigational states or  $i = n - 1$ 
14:       $\text{part} = O[\text{temp} : i + 1]$ 
15:      if the navigational state of  $\text{part}$  is the navigational state of main engine stop operation
16:        Add  $\text{part}$  into  $T(NS_0)$ 
17:      else
18:        Add  $\text{part}$  into  $T(NS_1)$ 
19:      end if
20:       $\text{temp} = i$ 
21:    end for
22:  return  $T(NS_0), T(NS_1)\}$ 
23:   $\text{AC}(T, \text{threshold})\{$ 
24:   $n$  is the size of point set  $T$ 
25:  set  $\sigma_{\max}$  as 0
26:  for  $i = 1$  to  $n - 2$  do
27:    Calculate the acceleration change rate  $\sigma$  from  $T[i]$  to  $T[0]T[n - 1]$  through Equation (13)
28:    if  $\sigma > \sigma_{\max}$  then
29:       $\sigma_{\max} = \sigma$ 
30:       $\text{index} = i$ 
31:    end if
32:  end for
33:  if  $\sigma_{\max} > \text{threshold}$  then
34:     $\text{AC}(T[0 : \text{index} + 1], \text{threshold})$ 
35:     $\text{AC}(T[\text{index} : n], \text{threshold})$ 
36:  else
37:    Add  $T[0]$  into Simplified_trajectory
38:    Add  $T[n - 1]$  into Simplified_trajectory
39:  end if}

```

---

## Appendix B. Comparing the Three Algorithm Profiles in the Experiment

**Table A1.** DP algorithm.

Algorithm introduction	Select the first point $P_a$ and the last point $P_b$ in the trajectory, and connect these two points into a line segment $L_{a,b}$ . Then, calculate the perpendicular distance from all points between $P_a$ and $P_b$ to the line corresponding to $L_{a,b}$ . The maximum perpendicular distance $D_{max}$ is taken and compared with the threshold $\sigma$ . If $D_{max} > \sigma$ , the corresponding points are kept in the generated set. If $D_{max} \leq \sigma$ , then all points between $P_a$ and $P_b$ are discarded. The above process is repeated recursively on each line segment until the end of the recursion.
Main features	Location (longitude and latitude).
Threshold setting	Generally, in terms of ship length, set a multiple of the ship length as the threshold, such as 0.5 times, 0.8 times, and 1 time.

**Table A2.** TD-TR algorithm.

Algorithm introduction	The algorithm flow is basically the same as that of the DP algorithm; the only difference is that the maximum vertical distance $D_{max}$ is modified to the distance corresponding to the time-synchronized position of the line segment $L_{a,b}$ .
Main features	Location (longitude and latitude) and time.
Threshold setting	Generally, in terms of ship length, set a multiple of the ship length as the threshold, such as 0.5 times, 0.8 times, and 1 time.

**Table A3.** CSB algorithm.

Algorithm introduction	The proposed algorithm has two main parts: the DP algorithm is employed to simplify the trajectories according to spatial features, and a sliding window is adopted to simplify the trajectories based on motion features. Furthermore, the statistical theory is applied to help determine the thresholds of the motion features in the sliding window algorithms. Finally, the two results are combined to form a simplified trajectory.
Main features	Location (longitude and latitude), speed, and ship heading.
Threshold setting	The DP algorithm portion is still chosen as a multiple of the ship length, and the sliding window simplified trajectory part sets the threshold with reference to the published literature of this algorithm.

## References

- Huang, L.; Wen, Y.; Geng, X.; Zhou, C.; Xiao, C.; Zhang, F. Estimation and spatio-temporal analysis of ship exhaust emission in a port area. *Ocean Eng.* **2017**, *140*, 401–411. [\[CrossRef\]](#)
- Toscano, D.; Murena, F.; Quaranta, F.; Mocerino, L. Assessment of the impact of ship emissions on air quality based on a complete annual emission inventory using AIS data for the port of Naples. *Ocean Eng.* **2021**, *232*, 109166. [\[CrossRef\]](#)
- Huang, L.; Wen, Y.; Zhang, Y.; Zhou, C.; Yang, T. Dynamic calculation of ship exhaust emissions based on real-time ais data. *Transp. Res. Part D Transp. Environ.* **2020**, *80*, 102277. [\[CrossRef\]](#)
- Goldsworthy, B. Spatial and temporal allocation of ship exhaust emissions in Australian coastal waters using AIS data: Analysis and treatment of data gaps. *Atmos. Environ.* **2017**, *163*, 77–86. [\[CrossRef\]](#)
- Makris, A.; Kontopoulos, I.; Alimisis, P.; Tserpes, K. A Comparison of Trajectory Compression Algorithms Over AIS Data. *IEEE Access* **2021**, *9*, 92516–92530. [\[CrossRef\]](#)
- Douglas, D.H.; Peucker, T.K. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartogr. Int. J. Geogr. Inf. Geovisualization* **1973**, *10*, 112–122. [\[CrossRef\]](#)
- Meratnia, N.; By, R. Spatiotemporal Compression Techniques for Moving Point Objects. In *Extending Database Technology*; Springer: Berlin/Heidelberg, Germany, 2004.
- Jensen, I.H. *Compressing Spatio-Temporal Trajectories*; Springer: Berlin/Heidelberg, Germany, 2014.
- Potamias, M.; Patroumpas, K.; Sellis, T. Sampling Trajectory Streams with Spatiotemporal Criteria. In *International Conference on Scientific & Statistical Database Management*; IEEE Computer Society: Washington, DC, USA, 2006.



10. Cudremauroux, P.; Wu, E.; Madden, S.R. *TrajStore: An Adaptive Storage System for Very Large Trajectory Data Sets*; IEEE: Piscataway, NJ, USA, 2010.
11. Yan, L.; Liu, R.W.; Liu, J.; Yu, H.; Hu, B.; Kai, W. Trajectory Compression-Guided Visualization of Spatio-Temporal AIS Vessel Density. In *Proceeding of the International Conference on Wireless Communications & Signal Processing*; IEEE: Piscataway, NJ, USA, 2016.
12. Han, Y.; Sun, W.; Zheng, B. Compress: A comprehensive framework of trajectory compression in road networks. *ACM Trans. Database Syst.* **2017**, *42*, 1–49. [\[CrossRef\]](#)
13. Hershberger, J.; Snoeyink, J. Speeding up the douglas-peucker line-simplification algorithm. *Proc. Intl. Symp. Spat. Data Handl.* **2000**, 134–143.
14. Hershberger, J.; Snoeyink, J. An  $O(n \log n)$  implementation of the Douglas-Peucker algorithm for line simplification. *Tenth Symp. Comput. Geom. DBLP* **1994**, 383–384.
15. Visvalingam, M.; Whyatt, J.D. The douglas-peucker algorithm for line simplification: Re-evaluation through visualization. *Comput. Graph. Forum* **2010**, *9*, 213–225. [\[CrossRef\]](#)
16. Zhao, L.; Shi, G. A method for simplifying ship trajectory based on improved douglas-peucker algorithm. *Ocean. Eng.* **2018**, *166*, 37–46. [\[CrossRef\]](#)
17. Cho, S.W. Ais trajectories simplification algorithm considering topographic information. *Sensors* **2022**, *22*, 7036.
18. Peng, X.; Wen, Y.; Wu, L.; Xiao, C.; Han, D. A sampling method for calculating regional ship emission inventories. *Transp. Res. Part D Transp. Environ.* **2020**, *89*, 102617. [\[CrossRef\]](#)
19. Ji, Y.; Qi, L.; Balling, R. A dynamic adaptive grating algorithm for ais-based ship trajectory compression. *J. Navig.* **2022**, *75*, 213–229. [\[CrossRef\]](#)
20. Wei, Z.; Xie, X.; Zhang, X. Ais trajectory simplification algorithm considering ship behaviours. *Ocean. Eng.* **2020**, *216*, 108086. [\[CrossRef\]](#)
21. Tang, C.; Wang, H.; Zhao, J.; Tang, Y.; Xiao, Y. A method for compressing ais trajectory data based on the adaptive-threshold douglas-peucker algorithm. *Ocean. Eng.* **2021**, *232*, 109041. [\[CrossRef\]](#)
22. Yan, R.; Mo, H.; Yang, D.; Wang, S. Development of denoising and compression algorithms for ais-based vessel trajectories. *Ocean. Eng.* **2022**, *252*, 111207. [\[CrossRef\]](#)
23. Han, Z.R.; Guang-Luan, X.U.; Huang, T.L.; Ren, W.J.; Electronic, S.O. Vessel trajectory outlier detection algorithm based on adaptive threshold. *Comput. Mod.* **2018**, *9*, 42.
24. Li, R.; Li, S.-X.; Liu, X.R.; Zhang, J.F. Research on Ship Trajectory Compression Algorithm Based on Cumulative Heading Variation. In *Proceedings of the 2019 International Conference on Artificial Intelligence, Control and Automation Engineering (AICAE 2019)*, Dalian, China, 23–24 June 2019.
25. Smierzchalski, R.; Michalewicz, Z. Adaptive Modeling of a Ship Trajectory in Collision Situations at Sea. In *Proceedings of the IEEE World Congress on IEEE International Conference on Evolutionary Computation*; IEEE: Piscataway, NJ, USA, 1998.
26. IMO-MEPC Reduction of GHG Emissions from Ships. Fourth IMO GHG Study 2020. *Int. Marit. Organ.* **2020**. Available online: <https://imoarcticsummit.org/publications/imo-papers/mepc-75/reduction-of-ghg-emissions-from-ships-fourth-imogh-study-2020-final-report/> (accessed on 14 December 2021).
27. Buhaug, O.; Corbett, J.J.; Endresen, O.; Eyring, V.; Faber, J.; Hanayama, S.; Lee, D.S.; Lee, D.; Lindstad, H.; Markowska, A.Z.; et al. *Second IMO Greenhouse Gas Study 2009*; International Maritime Organization: London, UK, 2009; Available online: <https://www.imo.org/en/OurWork/Environment/Pages/Greenhouse-Gas-Study-2009.aspx> (accessed on 14 December 2021).
28. Yang, L.; Zhang, Q.; Zhang, Y.; Lv, Z.; Mao, H. An ais-based emission inventory and the impact on air quality in Tianjin port based on localized emission factors. *Sci. Total Environ.* **2021**, *783*, 146869. [\[CrossRef\]](#)
29. Jalkanen, J.P.; Brink, A.; Kalli, J.; Pettersson, H.; Kukkonen, J.; Stipa, T. A modelling system for the exhaust emissions of marine traffic and its application in the baltic sea area. *Atmos. Chem. Phys.* **2009**, *9*, 9209–9223. [\[CrossRef\]](#)
30. Ristic, B.; Scala, B.L.; Morelande, M.; Gordon, N. Statistical Analysis of Motion Patterns in AIS Data: Anomaly Detection and Motion Prediction. In *Proceedings of the Information Fusion, 2008 11th International Conference*; IEEE: Piscataway, NJ, USA, 2008.
31. Pallotta, G.; Vespe, M.; Bryan, K. Vessel pattern knowledge discovery from ais data: A framework for anomaly detection and route prediction. *Entropy* **2013**, *15*, 2218–2245. [\[CrossRef\]](#)
32. Iperen, E.V. Detection of hazardous encounters at the North Sea from AIS data. In *Proceedings of the IWNTM' 2012*, Shanghai, China, September 2012.
33. Liaw, A.; Wiener, M. Classification and regression by randomforest. *R News* **2002**, *23*, 18–22.
34. Dyer, S.A.; Dyer, J.S. Cubic-spline interpolation. 1. *IEEE Instrum. Meas. Mag.* **2001**, *4*, 44–46. [\[CrossRef\]](#)
35. Michiels, W.; Korst, J.; Aarts, E. Time Complexity. In *Theoretical Aspects of Local Search*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 97–134.
36. Bertsekas, D.P.; Tsitsiklis, J.N. *Parallel and Distributed Computation: Numerical Methods*; Prentice Hall: Englewood Cliffs, NJ, USA, 1989.

**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.