Optimising Maritime Big Data by K-means Clustering with Mapreduce Model

Tuan-Anh Pham[1,2], Xuan-Kien Dang[1(✉)], and Nguyen-Son Vo[3]

[1] Ho Chi Minh City University of Transport, Ho Chi Minh City 700000, Vietnam
kien.dang@ut.edu.vn
[2] Southern Vietnam Maritime Safety Corporation,
Ho Chi Minh City 700000, Vietnam
[3] Institute of Fundamental and Applied Sciences, Duy Tan University,
Ho Chi Minh City 700000, Vietnam
vonguyenson@duytan.edu.vn

Abstract. During the management and operation, the maritime industry has collected a large amount of data in marine navigation, which has posed a great challenge in terms of resource saving (memory and processing capacity) and utility efficiency. Therefore, the highly specialised nature of the marine navigation and the maritime big data must be analysed to assist the scientists and operational engineers to extract the useful information from this data using algorithms with big data platforms. However, a specific model for big data application, which has a lot of methods for performing such as data visualisation techniques, machine learning, deep learning, etc., has not been extensively studied in the field of marine navigation to provide adequate comparisons. In this paper, we apply Mapreduce (MR) model to the big data of marine navigation. Particularly, we use a standard clustering algorithm called K-means based on the MR model to process the data of marine traffic in the South Vietnam Sea region. According to the main results obtained, we consider making the inference or the prediction of the clustering data which is collected and shown the dashboard of maritime ships traffic, including the scale, the spatial and time-series distribution situation.

Keywords: AIS data · Data mining · K-means clustering · Mapreduce model

1 Introduction

In recent years, the significant growth of the marine sector has occurred in maritime big data along with a dense network of ships [1], particularly the concentration of large seaports is capable of accepting ships up to 160,000 DWT. Marine data is often used in an automatic identification system (AIS) [2] which offers a wealth of real-time information on a ship's navigation utilised for maritime

N.-S. Vo et al. (Eds.): INISCOM 2022, LNICST 444, pp. 136–151, 2022.
https://doi.org/10.1007/978-3-031-08878-0_10

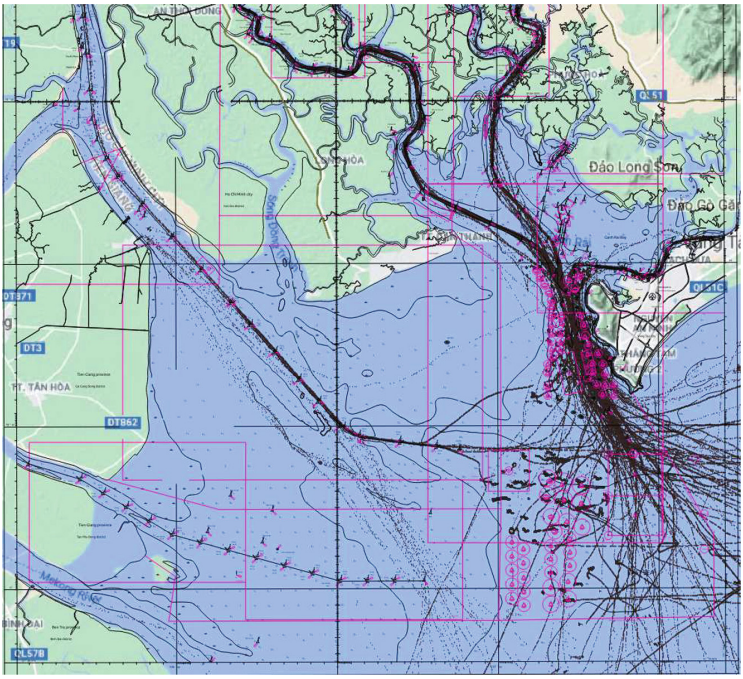

Fig. 1. Dynamic visualisation of AIS data, South Vietnam Sea region, January 1st, 2019.

situational awareness and ocean surveillance. According to the statistics on the quantity of data, i.e., more than 2 million notices, collected from the AIS system in the previous years in the South Vietnam Sea region, it is quite important to provide a superb supply of data mining for maritime traffic research. For example, Fig. 1 shows a sample data of dense maps captured on January 1st, 2019.

Generally, the maritime data is collected through the AIS and contains a lot of information like time, name, maritime mobile service identity (MMSI), speed over ground (SOG), course over ground (COG), etc. [3,4]. Moreover, the analysis and investigation of big data can quickly, automatically, and intelligently determine the characteristics of a ship such as position and navigation behavior, thereby orienting the effective development of the maritime industry and contributing to the development of the marine economy. The data acquired, some of which is repeated, along with the tremendous data of the ship's position, performs two obstacles for its use including large-scale data manipulation (e.g., sensor fault identification, data classification [2], data compression, data expansion, data integrity, and data regression [5]) and data complexity mining [6].

Dealing with the aforementioned issues, Hadoop architecture processing engine enables parallel data processing in a cluster [7,8]. It is necessary to design a K-means clustering (KMC) by means of maritime big data based on Hadoop architecture that implements the Mapreduce (MR) model [9]. We further use the Elbow

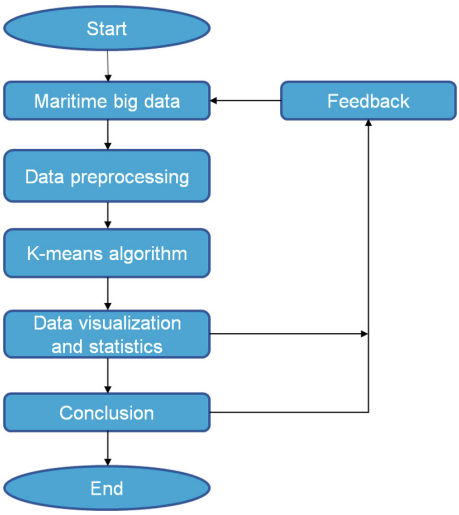

Fig. 2. Flowchart of data visualisation and data analysis.

Rule to determine the optimal number of clusters and recalculate the pairs of vectors SOG and COG, i.e., the feature vector when performing clustering, which is better perceived by the statistics and distribution of ships. As a result, the goal of this study is to improve the KMC of maritime data using the MR model (MRM) [10]. To do so, a flowchart of data visualisation and analysis is proposed in Fig. 2. The workflow includes choosing marine data fields, preprocessing data, K-means algorithm (KMA), statistics and data visualisation, conclusion, and feedback. It is noted in Fig. 2 that the data preprocessing step can be omitted if the maritime data is correct. The main contributions of this work are given as follows:

1. We first detect the data errors and remove them, convert the data, and extract the data from the source.
2. We then use the KMA [11–13] to perform the corresponding clustering step after preprocessing the received data and marine data field selection.
3. Finally, through data visualisation, we analyse the results and make some recommendations on the selection of content that displays the information for efficient marine navigation.

The rest of the paper is organised as follows. Section 2 shows the KMA and Hadoop architecture implementation of MRM for analysing the clustered characteristics of the maritime data. In Sect. 3, by using Hadoop architecture to perform the MRM, we determine the optimal number of clusters evaluating the navigation of ship traffic in the South Vietnam Sea. Then, we perform the KMC with two testing cases and analyse the results in Sect. 4. Finally, we conclude the paper in Sect. 5.

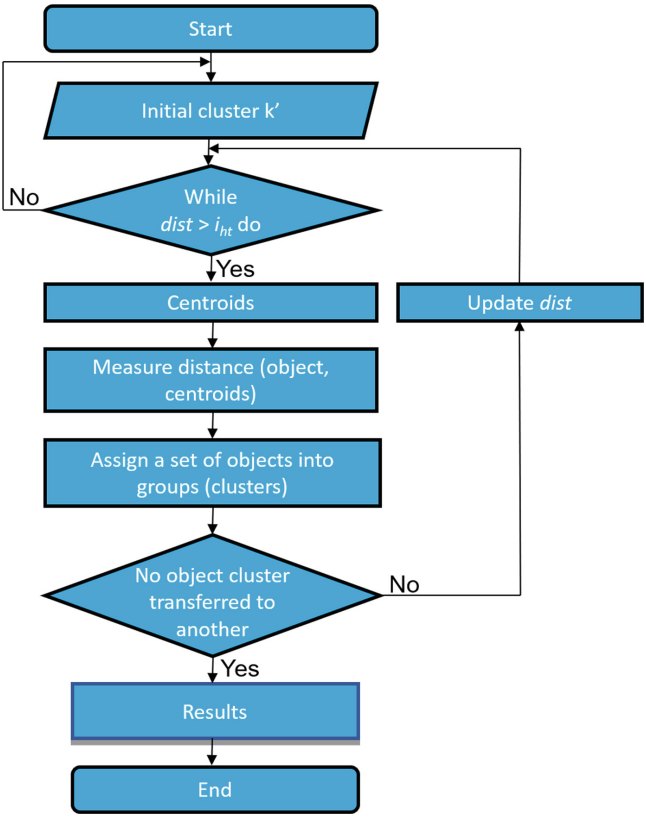

Fig. 3. Flowchart of the KMA analysis.

2 KMA and Hadoop Architecture Implementation of MRM

2.1 KMA

The KMA is used to analysing the clustered characteristics of the data. Figure 3 presents the flowchart of the KMA analysis. In this flowchart, the KMC method utilises the input, standard function E, and output. The Pseudo code describing the KMA [14,15] is presented in Algorithm 1. In Algorithm 1, the standard function E using Euclidean distance and the new point of k' clusters are given by

$$E = \sum_{v=1}^{N} \sum_{\substack{i=1 \\ x_v \in C_i}}^{k'} \| x_v - c_i \|^2, \tag{1}$$

$$C_i^{(j)} = \{x_v : \| x_v - c_i^{(j)} \|^2 \leq \| x_v - c_m^{(j)} \|^2, m = 1, 2, ..., k'\}, \tag{2}$$

Algorithm 1. K-means (X', k')

Input: Data set of N objects $X' = \{x_v | v = 1, 2, ..., N\}, x_v \in R^d$, d-dimensional vector
Output: Separated clusters C_i $(i = 1, 2, ..., k')$ and minimum standard function E (1)

1: Generate initial parameters
   $j = 1$
   Convergent boundary i_{ht}
   k' centers from X' as the initial cluster centers $C^{(0)} = \{c_i^{(j)}\}, i = 1, 2, ..., k'$ using KMC
   $dist \leftarrow E$ using (1)
2: **while** $dist > i_{\text{ht}}$ **do**
3: Form k' clusters by assigning all points in the set X' to the nearest central point
4: Find the new point of k' clusters $c_1^{(++j)}, c_2^{(++j)}, ..., c_{k'}^{(++j)}$ using (2) and (3)
5: Update $dist \leftarrow \sum_{i=1}^{k'} \| c_i^{(j)} - c_i^{(j-1)} \|^2$
6: **end while**

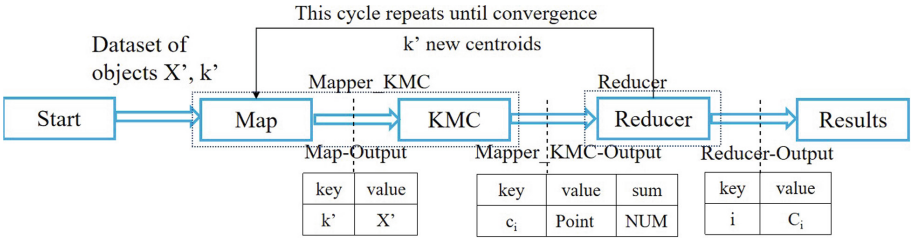

Fig. 4. Basic process of MRM with KMC.

$$c_i^{(++j)} = \frac{1}{|C_i^{(j)}|} \sum_{x_v \in C_i^{(j)}} x_v. \tag{3}$$

The Algorithm 1 works on a d-dimensional vector set, the data set X' includes N elements. The KMA repeats the process many times including data assignment and centroid update. The KMA assigns each point c_i to the cluster with the nearest center, which is the average for each distinct dimension overall point in the cluster. The process stops when the centroids converge and each object is part of a cluster.

2.2 MRM

The problem of ever-increasing data volume generated by technological advancements makes clustering a major undertaking, especially in maritime data. The studies in [7,8] attempt to solve this problem by developing effective clustering methods. Furthermore, the MRM, which is a model exclusively designed by Google, has the ability to programmatically process the large data sets in parallel and distributed algorithms on a cluster of computers. The MRM includes a map

Function 1. Mapper_KMC(X', k')

Input: A list of $< X', k' >$ pairs and a list of center global centroids, i.e, k' is the index of data point and X' is the content of object and d-dimensional array

Output: $< i, \text{Point}, NUM >$, i.e., i is the index of cluster (nearest centroid), Point is the value of the sample information series of objects, and NUM is the sum of data points belonging to that cluster

```
Generate initial parameters
Initialise a sample scenario from X', i.e., the values of the d-dimensional array
nearest_distance ← Double.max_value (1000000000)
nearest_cluster_id ← None
for i = 0 to length.center do
    Distance = distance_function(scenario, c_i)
    if Distance < nearest_distance then
        nearest_distance = Distance
        nearest_distance_id = i
    end if
end for
Create an empty dictionary dict() and initialise a cont_index
Set the NUM counter record to be the sum of samples scenario belonging to the same cluster
Calculate NUM by
    - Adding the nearest_distance as NUM into dict()
    - Adding NUM+ = scenario
    - cont_index++
    - Adding NUM and cont_index into dict()
Obtain < i, Point, NUM >
```

function and a reduce function. The map function covers the task of assigning each sample to the nearest center, whereas the reduce function deals with the process of updating the new centers. This paper further considers integrating the process of KMC into the map function of the MRM as shown in Fig. 4 [14,16]. The so-called mapper and KMC function and reducer function [17], which are used in Sect. 4 for the process of testing and analysing, are described in Function 1 and Function 2.

2.3 Hadoop Architecture Implementation of MRM

In this section, we focus on a distributed system known as Hadoop architecture that uses commodity machines to create a combined and powerful system. This system is the most popular open-source framework for the process of MRM proposed by Google [9], which is able to process the maritime big data much more efficiently. This framework includes map and reduce functions. The major input for the framework is a key-value pair (key, value) and the map function is performed to process the input in key-value pair (key, value) one by one. The map function is created from more intermediate key-value pairs (key', value'). After that, it groups these intermediate key-value pairs by the key k', so the system

Function 2. Reducer(i, Point, NUM)

Input: $< i, \text{Point}, NUM >$, where i is the index of cluster (nearest centroid), Point is the value of the sample information series of objects and NUM is the sum of data points belonging to that cluster

Output: $< i, C_i >$ pairs, where i is the index of the cluster and C_i is its new global centroid

1: Initialise an array containing the sum of data points belonging to that cluster from the list of Point, i.e., the value of the sample information series of objects
2: Generate C_i from the mean of NUM belonging to the same cluster as a string value
3: Obtain $< i, C_i >$ pairs

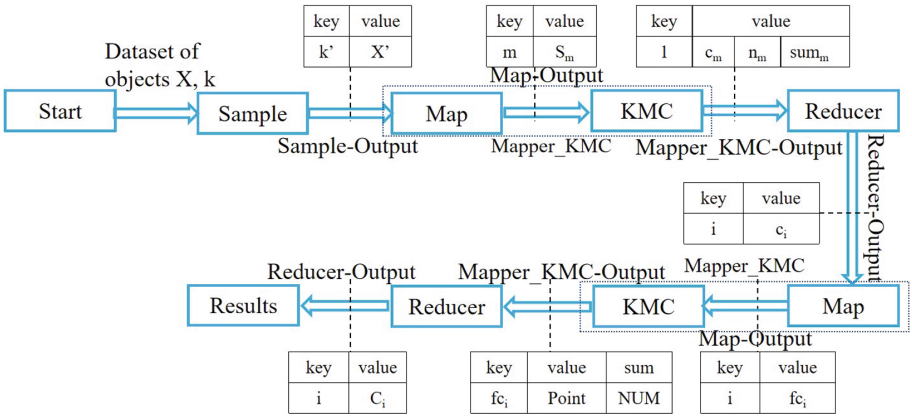

Fig. 5. Optimised MRM with KMC.

connects the reduce function for each clustering, which collects and aggregates into results from the map function. One of the main advantages of this framework is the important task of defining the map and reduce functions to perform a large-scale data analysis. However, the I/O performance of the Hadoop architecture depends on the Hadoop distributed system, which is a major open-source project to design for high volume and highly reliable storage.

3 Optimised MRM with KMC

The optimised MRM with KMC is shown in Fig. 5, which it is improved over the traditional method [14] by further calculating the total within-cluster sum of the objects for each cluster. We set k clusters and repeat the process until the center point of the cluster converges. It is noted in Fig. 5 that fc_i is the final center i. This rule indicates the center point of the cluster to the mean point of the data set and then splits the elements within it. To determine the number of

Table 1. The contents of AIS data [19] and AIS data record.

Type	Contents	AIS data record
Static data	imo, mmsi, class, shipname, shiptype, callsign, length, beam, deadweight	Time: 01/01/2019; Data (sample data): 61.091.450
Dynamic data	tagblock_times (UTC), status (navigation status), lon (longitude), lat (latitude), SOG (speed), COG (course), heading, turnrate	Marine objects (*): 943 Data field (**): 25
Auxiliary data	band, destination (port), draught	

*Marine objects: (ships, AIS is integrated signaling devices, ...)
**Data field (including mobile and static, information of marine objects)

clusters k, we use the Elbow Rule to calculate the optimal number of clusters [18], given as below

$$\underset{k}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{x \in C_i} \|x - c_i\|^2, \tag{4}$$

where k is the number of clusters, c_i is the center point of cluster C_i, and x is the feature vector of each ship's trajectory in X.

In particular, we calculate the Euclidean interval from every sample to the center point of the cluster by using (4). Then, we proceed to different values of k. The total distance decreases as k increases, so it will converge and the position of the largest point is considered as the convergence point, i.e., the elbow.

4 Testing and Analysis

The collected AIS data of the Southern Vietnam Sea is very rich. It is a variety of navigation status information for maritime traffic. The trajectory of the ships is determined by linking the ship's position information collected by the AIS to the system operation center. However, the amount of AIS data collected by each ship is not uniform, which can be caused a signal congestion or a failure of the transmitters and conducted by identifying the unique MMSI of each ship. In this case, we remove this AIS data collected because it completely does not represent the sailing pattern of this ship. The standard deviation of the feature SOG and the feature COG of each ship converted from degrees (^{0}) to radians is calculated using the feature $\mathrm{SOG_{sd}}$ and pairs of vectors ($\mathrm{SOG_{sd}}$, COG) to assess the stability of the ship. In addition, we normalise the sample before clustering by using logarithmic normalisation [18], expressed as

$$\mathrm{SOG_{sd}} = \frac{\log_{10}(\mathrm{SOG} + 1)}{\log_{10}(\mathrm{SOG_{max}} + 1)}. \tag{5}$$

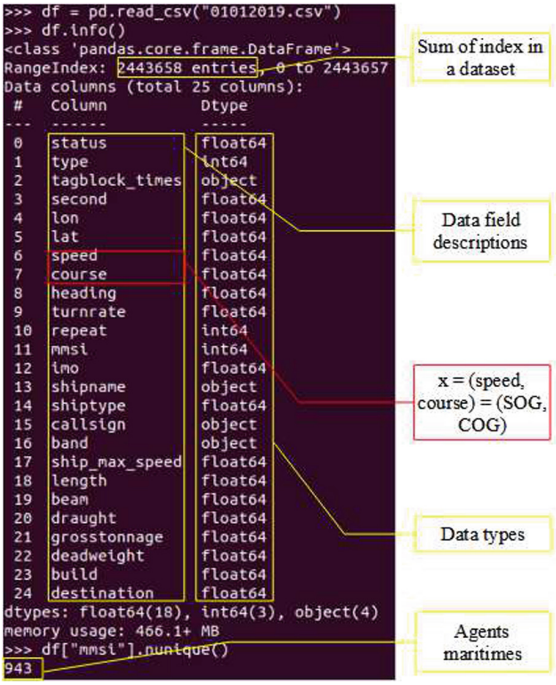

Fig. 6. Detail of the framework for the AIS data collected.

4.1 Data Input

The contents of AIS data and maritime AIS data sample are described in Table 1. There are three types of data, including static data, dynamic data, and auxiliary data. In addition, the detail of the framework for the collected AIS data is shown in Fig. 6, in which the standard deviation values of SOG are marked in red rectangle. These values are used as the pairs of feature vectors SOG and COG of each object in the implementation of KMC process presented in the sequel.

4.2 Implementation of MRM with KMC

In this paper, we implement the MRM with KMC (MRM-KMC) in two scenarios. i.e., randomly taking 3 central points (3CP) and 7 central points (7CP). We also provide the clustering results of these two scenarios to demonstrate the performance of the proposed MRM-KMC solution.

3CP Scenario. To implement the 3CP scenario, three initial parameters with the feature vector of SOG and COG of each ship are listed in Table 2. Furthermore, the Hadoop architecture used to build the model is shown in Fig. 7. We receive the result of the 3CP scenario using 3 different starting the pairs of vectors SOG and COG (Table 3) which is applied to the model to find out the

Table 2. Selecting three parameters with the feature vector of SOG and COG of each ship.

k clusters	SOG	COG
1	0.0	104.400001525879
2	0.0	160.0
3	0.4000000059604645	227.3000030517578

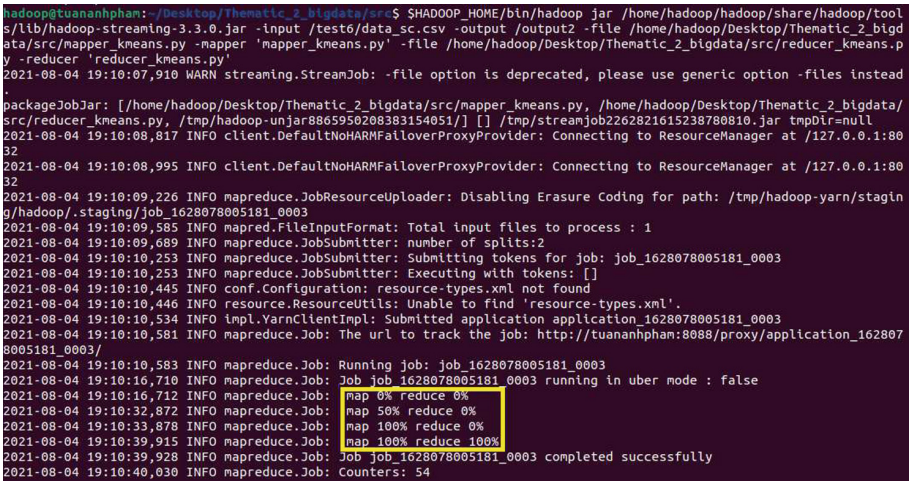

Fig. 7. Implementation of MRM-KMC with $k = 3$.

Table 3. Result of 3CP scenario.

k clusters	SOG_new	COG_new
1	1.1448615977471592	66.9534619292097
2	1.4988419143613	160.48269968421656
3	2.7800271048373233	280.0375339552592

characteristics of clusters. The result in Fig. 8 shows that the objective function gets saturated at a minimal value of 2182.00 when $k = 3$. The increase of k cannot improve the result any more. Importantly, as shown in Fig. 9, the majority of the ships is green and blue, which indicates that the ship's navigation is relatively stable. Meanwhile, the orange ones are more unstable, i.e., the pairs of feature vectors SOG and COG are changed over time frequently. In Fig. 9, the results change after each iteration from left to right and top to bottom until the end when $dist$ is less than i_{ht} as mentioned in the flowchart of KMA (Fig. 3).

7CP Scenario. Similarly, to implement the 7CP scenario, seven initial parameters with the feature vector of SOG and COG of each ship are listed in Table 4.

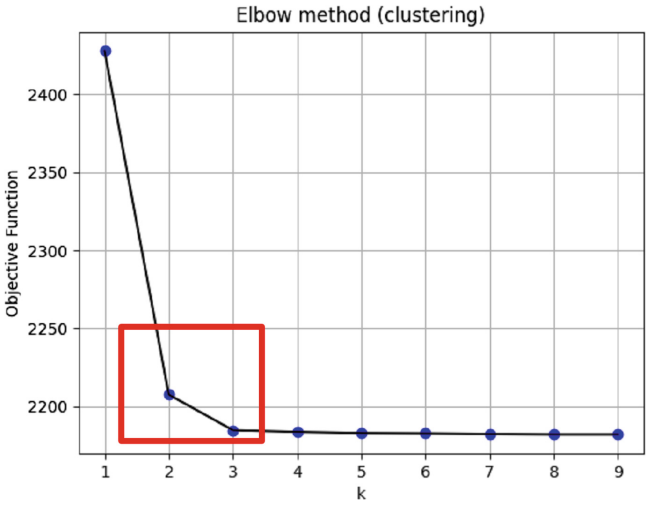

Fig. 8. Objective function value with $k = 3$.

Table 4. Selecting seven parameters with the feature vector of SOG and COG of each ship.

k clusters	SOG	COG
1	0.0	41.79999923706055
2	0.0	104.400001525879
3	0.0	155.0
4	0.0	160.0
5	8.699999809265137	184.0
6	0.4000000059604645	227.3000030517578
7	0.4000000059604645	355.20001220703125

The Hadoop architecture used to build the model is shown in Fig. 10. We also receive the result of the 7CP scenario using 7 different starting the pairs of vectors SOG and COG (Table 5) which is applied to the model to find out the characteristics of clusters. The result in Fig. 11 shows that the objective function starts getting saturated at a minimal value of 1235.73 when $k = 7$. The increase of k cannot make any further improvements. Importantly, as shown in Fig. 12, the majority of the ships is red, blue, purple, and green, which indicates that the ship's navigation is relatively stable. Meanwhile, the pink and brown ones are more unstable due to more changes in the SOG and the COG over time.

Generally, according to the implemented results in Fig. 8, Fig. 9, Fig. 10, Fig. 11, and Fig. 12, we can see that the majority of the ships is green and blue (with $k = 3$) or red, blue, purple, and green (with $k = 7$), which indicates that the ship's navigation is relatively constant over the period. Meanwhile, the

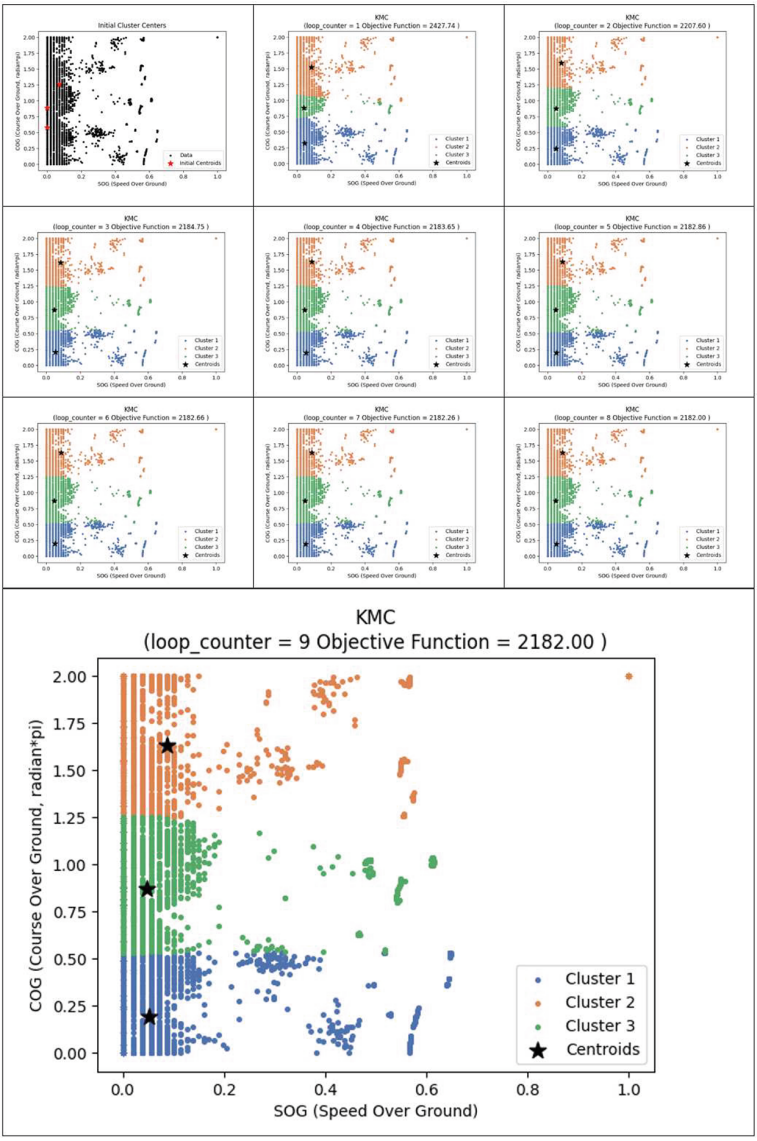

Fig. 9. Normalized the KMC results with $k = 3$.

orange, pink, and brown zones are less stable. Based on the obtained results, the standard deviation is a feature of SOG, which means the courses and speeds of ships are unstable and changing frequently. To arrange together with the objective function value, we determine the optimal number of clusters. Thereby, we can evaluate the navigation process of ships more easily.

```
hadoop@tuananhpham:~/Desktop/Thematic_2_bigdata/src$ $HADOOP_HOME/bin/hadoop jar /home/hadoop/hadoop/share/hadoop/tool
s/lib/hadoop-streaming-3.3.0.jar -input /test6/data_sc.csv -output /output1 -file /home/hadoop/Desktop/Thematic_2_bigd
ata/src/mapper_kmeans.py -mapper 'mapper_kmeans.py' -file /home/hadoop/Desktop/Thematic_2_bigdata/src/reducer_kmeans.p
y -reducer 'reducer_kmeans.py'
2021-08-04 19:07:28,728 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead
.
packageJobJar: [/home/hadoop/Desktop/Thematic_2_bigdata/src/mapper_kmeans.py, /home/hadoop/Desktop/Thematic_2_bigdata/
src/reducer_kmeans.py, /tmp/hadoop-unjar6265058714676650259/] [] /tmp/streamjob7239163228008924271.jar tmpDir=null
2021-08-04 19:07:29,653 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /127.0.0.1:80
32
2021-08-04 19:07:29,850 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /127.0.0.1:80
32
2021-08-04 19:07:30,088 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/stagin
g/hadoop/.staging/job_1628078005181_0002
2021-08-04 19:07:30,863 INFO mapred.FileInputFormat: Total input files to process : 1
2021-08-04 19:07:30,961 INFO mapreduce.JobSubmitter: number of splits:2
2021-08-04 19:07:31,113 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1628078005181_0002
2021-08-04 19:07:31,114 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-08-04 19:07:31,329 INFO conf.Configuration: resource-types.xml not found
2021-08-04 19:07:31,330 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-08-04 19:07:31,414 INFO impl.YarnClientImpl: Submitted application application_1628078005181_0002
2021-08-04 19:07:31,469 INFO mapreduce.Job: The url to track the job: http://tuananhpham:8088/proxy/application_162807
8005181_0002/
2021-08-04 19:07:31,471 INFO mapreduce.Job: Running job: job_1628078005181_0002
2021-08-04 19:07:38,599 INFO mapreduce.Job: Job job_1628078005181_0002 running in uber mode : false
2021-08-04 19:07:38,601 INFO mapreduce.Job:  map 0% reduce 0%
2021-08-04 19:07:55,793 INFO mapreduce.Job:  map 100% reduce 0%
2021-08-04 19:08:01,835 INFO mapreduce.Job:  map 100% reduce 100%
2021-08-04 19:08:02,855 INFO mapreduce.Job: Job job_1628078005181_0002 completed successfully
2021-08-04 19:08:02,937 INFO mapreduce.Job: Counters: 54
```

Fig. 10. Implementation of MRM-KMC with $k = 7$.

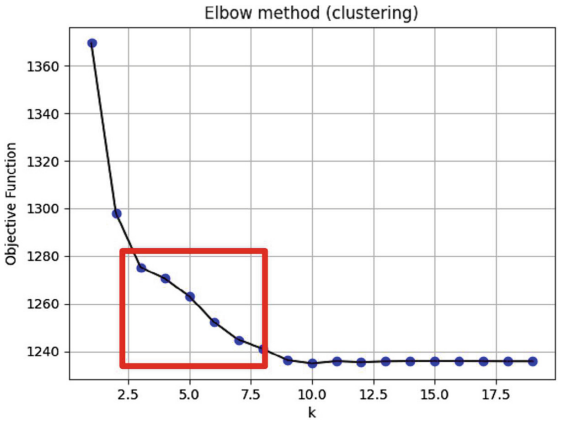

Fig. 11. Objective function value with $k = 7$.

Table 5. Result of 7CP scenario.

k clusters	SOG_new	COG_new
1	0.9424347227602381	33.06937842932461
2	1.374841608471513	102.5497472633251
3	1.4336294004340628	144.46374699151812
4	1.4844708358424163	164.190693091593
5	1.1571614708710054	190.33812071821583
6	0.6553970429182934	249.26870687848287
7	5.574822245091656	328.82117546900054

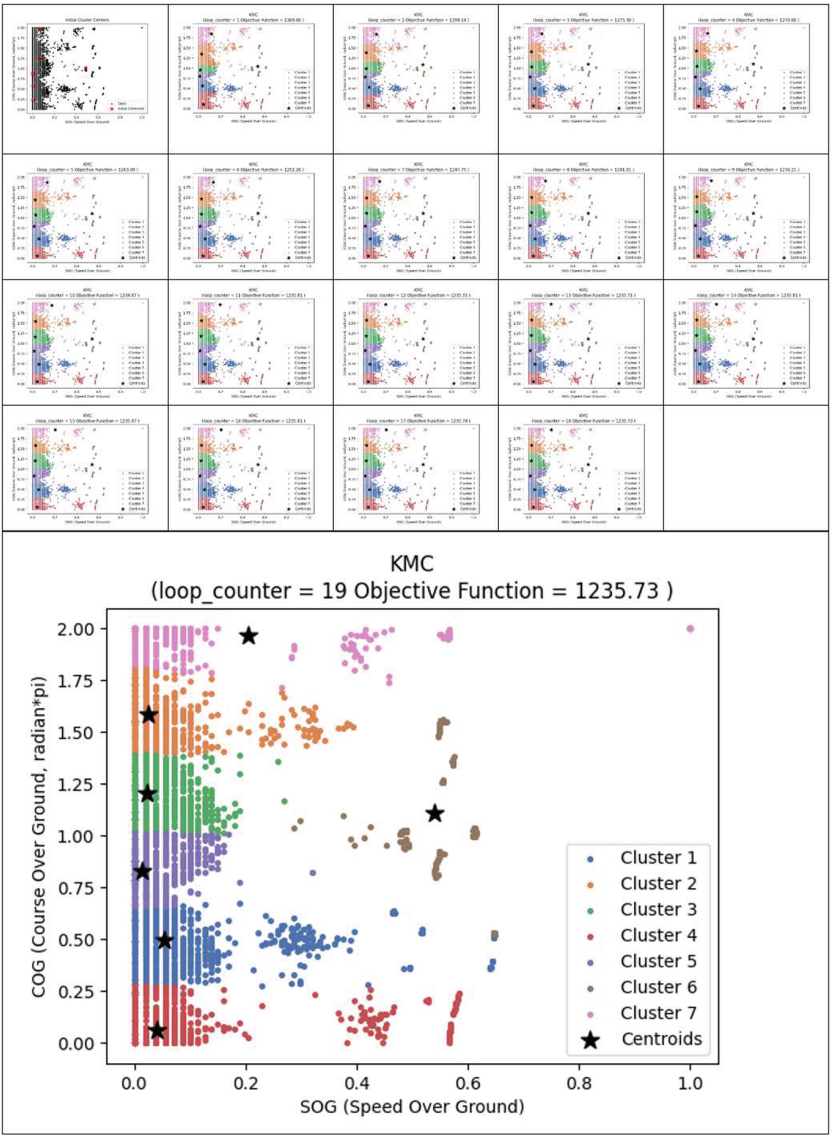

Fig. 12. Normalized the KMC results with $k = 7$.

5 Conclusion

In this paper, we have performed the analysis of maritime big data by MRM with KMC. This enables us to process the trajectory characteristics (standard deviation of pairs of vectors SOG and COG) for each ship from the AIS data collected and sent to the operating system center to analyse the maritime traffic

data in South Vietnam Sea. The proposed method, which evaluates the stability of ship's traffic, is also used to detect the anomaly in the navigation of ships based on the characteristics of the cluster. However, the information collected from AIS data still has many features that we have not fully exploited, and the extracted features need to be enhanced in our future works.

References

1. Fiorini, M., Capata, A., Bloisi, D.D.: AIS data visualization for maritime spatial planning (MSP). Int. J. e-Navigation Maritime Econ. **5**, 45–60 (2016)
2. Le, V.T., Dang, X.K., Nguyen, D., Pham, T.D.A.: A novel maritime risk assessment model of waterway transportation based on Takagi-Sugeno fuzzy logic: Vietnam case study. IOP Conf. Ser. Earth Environ. Sci. **527**(1), 1–8 (2020)
3. Alba, J.M.M., Dy, G.C., Viriña, N.I.M., Samonte, M.J.C., Cruz, F.R.G.: Localized monitoring mobile application for automatic identification system (AIS) for sea vessels. In: Proceedings of IEEE International Conference on Industrial Engineering and Applications, Bangkok, Thailand, pp. 790–794, April 2020
4. Wakabayashi, N., Jurdana, I.: Maritime communications and remote voyage monitoring. In: Proceedings of International Conference on Broadband Communications for Next Generation Networks and Multimedia Applications, Graz, Austria, pp. 1–8, July 2020
5. Han, J., Kamber, M., Pei, J.: DataMining: Concepts and Techniques, 3rd edn. Morgan Kaufmann, Burlington (2011)
6. Li, Y., Liu, Z., Zheng, Z.: Study on complexity of marine traffic based on traffic intrinsic features and data mining. J. Comput. Methods Sci. Eng. **19**(3), 619–633 (2019)
7. Aji, A., et al.: Hadoop GIS: a high performance spatial data warehousing system over mapreduce. VLDB Endow. **6**(11), 1009–1020 (2013)
8. Mujeeb, S., Sam, R.P., Madhavi, K.: Adaptive hybrid optimization enabled stack autoencoder-based Mapreduce framework for big data classification. In: Proceedings of International Conference on Emerging Trends in Information Technology and Engineering, Vellore, India, pp. 1–5, February 2020
9. Hadoop: Open source implementation of Mapreduce (2021). https://hadoop.apache.org
10. Wang, Z., Xu, A., Zhang, Z., Wang, C., Liu, A., Hu, X.: The parallelization and optimization of k-means algorithm based on spark. In: Proceedings of International Conference on Computer Science & Education, Delft, Netherlands, pp. 457–462, August 2020
11. Lee, S.G., Lee, C.: Developing an improved fingerprint positioning radio map using the K-means clustering algorithm. In: Proceedings of International Conference on Information Networking, Barcelona, Spain, pp. 761–765, January 2020
12. Ng, Y., Pereira, J.M., Garagic, D., Tarokh, V.: Robust marine buoy placement for ship detection using dropout K-means. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Barcelona, Spain, pp. 3757–3761, May 2020
13. Shen, H., Duan, Z.: Application research of clustering algorithm based on K-means in data mining. In: Proceedings of International Conference on Computer Information and Big Data Applications, Guiyang, China, pp. 66–69, April 2020

14. Cui, X., Zhu, P., Yang, X., Li, K., Ji, C.: Optimized big data K-means clustering using Mapreduce. J. Supercomput. **70**, 1249–1259 (2014)
15. Hartigan, J.A., Wong, M.A.: Algorithm AS 136: a K-means clustering algorithm. J. Roy. Stat. Soc.: Ser. C (Appl. Stat.) **28**(1), 100–108 (1979)
16. Lin, Y., Ma, K., Sun, R., Abraham, A.: Toward a MapReduce-based K-means method for multi-dimensional time serial data clustering. In: Abraham, A., Muhuri, P.K., Muda, A.K., Gandhi, N. (eds.) ISDA 2017. AISC, vol. 736, pp. 816–825. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-76348-4_78
17. Zhao, W., Ma, H., He, Q.: Parallel K-means clustering based on Mapreduce. In: Proceedings of International Conference on Cloud Computing, Beijing, China, pp. 674–679, November 2009
18. Hanyang, Z., Xin, S., Zhenguo, Y.: Vessel sailing patterns analysis from S-AIS data dased on *k*-means clustering algorithm. In: Proceedings of IEEE International Conference on Big Data Analytics, Suzhou, China, pp. 10–13, March 2019
19. IMO: Regulations for carriage of AIS, (I. M. Organization, Producer), AIS transponders (2021). https://www.imo.org