Trajectory Clustering Based Oceanic Anomaly Detection Using Argo Profile Floats

Wen-Yu Cai[1], Zi-Qiang Liu[1], and Mei-Yan Zhang[2(✉)]

[1] College of Electronics and Information, Hangzhou Dianzi University, Hangzhou, China
[2] School of Electrical Engineering, Zhejiang University of Water Resources and Electric Power, Hangzhou, China
dreampp2000@163.com

Abstract. The observation data of Argo profile floats are very crucial for long-term climate change and natural variability, which reflect three-dimensional distribution of temperature and salinity in the sea. In order to solve the anomalies in the profile caused by uncertainties factors, this paper proposes a novel anomaly detection method for Argo profile floats using an improved trajectory clustering method to discriminate normal and abnormal. The proposed algorithm partitions Argo data into a set of line segments, and then clusters line segments to get rid of noisy data, finally recovers the line segments to the raw data accordingly. As a result, the proposed oceanic anomaly detection method subtly converts the sequence data into line segments for anomaly detection, which considers both positional relationship and trend of data source. Extensive experiments on real dataset from Argo floats verify that our method has better results under different conditions compared to existing methods such as LOF and DBSCAN.

Keywords: Anomaly detection · Trajectory clustering · Oceanic observation data · Argo profile floats

1 Introduction

With the increasingly concerned about global change and its regional impacts, practical oceanic observation data are becoming more and more crucial. To combat historical lack of data, an innovative project named Argo was taken by scientists to greatly improve the collection of observations inside the ocean through increased sampling of old and new quantities and increased coverage in terms of time and area. Argo is a global array of 3,800 free-drifting profiling floats that measure the temperature and salinity of the upper 2000 m of the ocean. The Argo project allows a long-term continuous monitoring of the temperature, salinity, and velocity of the upper ocean, with all data being relayed and made publicly available within hours after collection [1]. However, the observation data of Argo profile floats are very valuable, which reflect the three-dimensional distribution of temperature and salinity of the sea and offer a database.

The data acquisition period of Argo profile floats is generally 10 days. They dive to a depth of 1000 m and drift for 9 days, and then descend to 2,000 m. The conductivity,

H. Gao et al. (Eds.): ChinaCom 2019, LNICST 312, pp. 498–508, 2020.
https://doi.org/10.1007/978-3-030-41114-5_37

temperature, depth (CTD) sensors are assembled on the Argo floats to record the ocean profile during the ascent. They transmit the collected profile to the satellite when back to sea surface. Due to the limited battery carried, Argo floats have a typical life cycle of 4–5 years. The positional uncertainty caused by the free-drifting of the floats makes maintenance and calibration very difficult. Argo profile floats work in the sea for a long time and the positional uncertainty caused by the free-drifting of the floats makes maintenance and calibration very difficult. Since local sensory data of Argo floats are prone to occur exceptions when sensors encounter a water mass abrupt layer or data transmission failure occurs. Moreover, sensors equipped in the floats are vulnerable by marine organisms or pollutants when it has been working for a long time, as a result the entire profile collected by underwater sensors may shift.

These anomalous data will affect some of the relevant research conducted by scientific researchers if they cannot be effectively identified. One of the most prominent being is the deviation of data from the adjacent data would not in the normal range. Several abnormal data detection methods in the marine field have been proposed. In order to guarantee the quality of data, Yusheng et al. [2] use a sliding window and improved AutoRegressive Integrated Moving Average model to detect and fill up the missing or suspect data. This simple method works well for continuously observed data, but it has some defects for non-continuous data. Andreas et al. [3] propose a robust and fast anomaly detection framework consists of cluster based auto associative kernel regression and sequential probability ratio test, which reconstructs the data and perform residual analysis to determine whether it is abnormal. Yosuke et al. [4] propose a new method for error detection in Argo observation data using conditional random field to realize an automatic QC (Quality Control) with high accuracy equal to human experts, but it has to consider the surrounding labels when assigns QC labels. Besides, a series of test methods for CTD data including location test, speed test, spike test, stuck value test, density inversion and etc. have been applied [5]. Furthermore, Jae-Gil et al. [6] propose a trajectory clustering method, which divides the complete trajectory into multiple line segments and then clusters the line segments to obtain a similar line segment set. Since it has many good features, we are devoting to improving and applying this idea to the filed of oceanic anomaly detection in this paper.

The rest of the paper is organized as follows. Section 2 presents our trajectory clustering based anomaly detection algorithm. Experiments and results are carried out in Sect. 3. Section 4 concludes this paper.

2 Trajectory Clustering Based Anomaly Detection

In this paper, we propose the clustering and restore framework to detect anomaly of Argo data. The proposed method consists of three main phases. Firstly, the characteristic data in the original data of Argo profile floats are extracted in the pre-processing phase. Afterwards, in the clustering phase, an improved Line Segment Clustering algorithm is used to cluster the line segments which are composed of adjacent two characteristic data. The line segments with the classification labels are reconstructed to the original data in the restore phase, and the raw data corresponding to noise segments will be determined as abnormal data.

2.1 Pre-processing Module

The Argo profile floats spread all over the world, and the profiles of different regions and time are quite different. Therefore, the adjacent similar profiles in time and space are selected as the basic dataset $T = \{TR_1, TR_2, \ldots, TR_{numtra}\}$, where num_{tra} is total number of profiles, and we treat each profile as a single trajectory and each data as a data point. A trajectory consists of many sequential data points denoted as $TR_i = \{p_1, p_2, \ldots, p_j \ldots, p_{leni}\}$, where the point p_j is three-dimensional data, $p_j = \{Pressure_j, Sensor\text{-}Data_j, Global\text{-}Index_j\}$. The first two dimensions of data are pressure value and sensory data using for segmentation and clustering. The global-index indicates the order for data point p_j in the basic dataset, used only in the restore section.

It is necessary to scale the points before partitioning and clustering sections, because the value difference in scale of different sensor is not the same, e.g., the pressure range is generally between 0–2000 dbar, but the salinity in range 2 to 41 PSU. Data can be normalized using Min-Max feature scaling according to the standard variation range of the data, and it can avoid the abnormal extremes effectively. The normalized formula is defined as Eq. (1). It is well known that the pressure range of Argo profile floats is between 0 to 2000 dbar, so we set X_{min} to 0 and X_{max} to 2000 when normalizing pressure data. The variation range of sensory data is not fixed, and it varies with the sea area. First, we find the maximal and the minimal sensory data in each trajectory TR_i, then calculate the difference between them to get the variation range V_{ri}. The median value from the V is chosen as the standard range M_r, $V = \{V_{r1}, V_{r2}, \ldots, V_{rnumtra}\}$. Finally, the median value is calculated from all sensory data in the dataset T as M_d, so X_{min} is M_d-M_r/2 and X_{max} is $M_d + M_r$/2.

$$Z_i = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}} \times 10 \tag{1}$$

Herein, the data characteristic points in each trajectory are extracted after normalizing raw dataset. The line segment consists of adjacent characteristic points that not only represent all data points, but also reflect the trend of these points, so as to reduce the number of data and operation consumption in the clustering process. The method of Approximate Trajectory Partitioning [6] is applied to find characteristic points in the data trajectory.

This proposed algorithm mainly uses the Minimum Description Length (MDL) principle to find the optimal trade-off between precision and simplicity. The input of this algorithm is a trajectory $TR_i = \{p_1, p_2, \ldots, p_j, \ldots, p_{leni}\}$, and the output is a set of characteristic points $CP_i = \{p_{c1}, p_{c2}, \ldots, p_{lenci}\}$. Two adjacent characteristic points in the set constitute a sub-trajectory (or line segment) $L_i = \{p_{ci}, p_{ci+1}\}$. Moreover, the trajectory and the line segment set can be represented by $TRL_i = \{L_1, L_2, ..., L_{lenci\text{-}1}\}$ and $\mathcal{D} = \{TRL_1, TRL_2, \ldots, TRL_{numtra}\}$ respectively.

2.2 Clustering Module

In this module, we propose a line segment clustering algorithm with noise interference, which is derived from Line Segment Clustering algorithm [6]. In line segment clustering process, we try compare two line segments instead of two data points, so the distance

definition between two line segments is crucial to performance. The process consists of three main parts: perpendicular distance ($d_\perp$), parallel distance ($d_{//}$), and angle distance (d_θ). Let $L_i = \{s_i, e_i\}$, $L_j = \{s_j, e_j\}$, where L_i is longer than L_j, p_s and p_e are the projections of points s_j and e_j on line segment L_i, which are illustrated in Fig. 1.

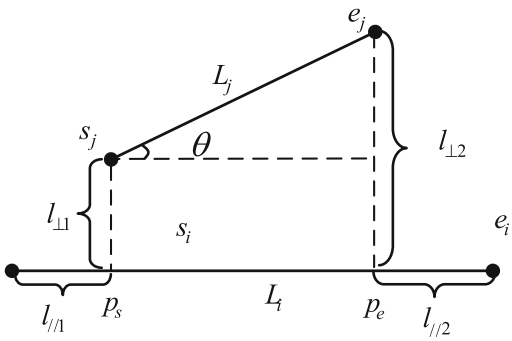


Fig. 1. The distance between two line segments.

Definition 1: The perpendicular distance between L_i and L_j is defined as Eq. (2). $l_{\perp 1}$ denotes the distance from point s_j to line L_i, which is the Euclidean distance from s_j to p_s, $l_{\perp 2}$ is that between e_j and p_e.

$$d_\perp = \frac{l_{\perp 1}^2 + l_{\perp 2}^2}{l_{\perp 1} + l_{\perp 2}} \tag{2}$$

Definition 2: The parallel distance between L_i and L_j is defined as Eq. (3). $l_{//1}$ is the minimum of the Euclidean distance between point p_s to s_i or e_i. $l_{//2}$ is the minimum of the Euclidean distance between point p_e to s_i or e_i.

$$d_{//} = \text{MIN}(l_{//1}, l_{//2}) \tag{3}$$

Definition 3: The angle distance between L_i and L_j is defined as Eq. (4). $\|L_i\|$ is the length of L_i, and θ is the angle between L_i and L_j. In the literature [6], the angle distance is defined as $\|L_j\| \times \sin\theta$, but we notice it is difficult to detect the angular abnormal line segment because the long abnormal line segment will be affected by the short line segment, even if the angle difference is large. $\|L_j\|$ is small, the angle distance is also small. Hence we improve the definition as follows.

$$d_\theta = \begin{cases} \|L_i\| \times \sin\theta & 0° \le \theta \le 90° \\ \|L_i\| & 90° \le \theta \le 180° \end{cases} \tag{4}$$

Definition 4: The total distance between the two line segments is defined as Eq. (5), w denotes the weighting value, and usually all of the weights are taken as 1.

$$\mathrm{dist}(L_i, L_j) = w_\theta \cdot d_\theta(L_i, L_j) + w_{//} \cdot d_{//}(L_i, L_j) + w_\perp \cdot d_\perp(L_i, L_j) \quad (5)$$

Up to now, we can discuss our trajectory anomaly detection algorithm. The proposed algorithm requires a line segment set $\mathcal{D}$and two parameters $\mathfrak{E}$ and *MinLns,* then we can get a set of O of clusters. As the algorithm in literature [6] does not define noise line segments, we define novel noise line segment concept as below definition.

Definition 5: There are several clusters $C_1, C_2, \ldots C_k \subseteq \mathcal{D}$w.r.t. $\mathfrak{E}$ and *MinLns*. A line segment $L_j \in \mathcal{D}$is noise if L_j does not belong to any cluster C_i, $1 \leq i \leq k$.

We regard the line segments that do not belong to O as a noise line segment after clustering, so a noise line segment set AL will be obtained with such operations.

2.3 Restore Module

The objective of this process is to restore from clustered line segments to raw data points so as to distinguish which points are normal or abnormal. Each line segment $L_i = \{p_{ci}, p_{ci+1}\}$ consists of two characteristic points, and the line segment can represent all the points between the two characteristic points. These two points are not necessarily adjacent in the basic dataset T, so we need Global-Index to indicate the position of the characteristic point in the basic dataset T. The raw points corresponding to the noise line segment set AL are marked as abnormal points. Due to the sensor drift phenomenon after being used for a long time, there may be a few entire profile anomalies in Argos. Hence, we need to check all of the trajectories, the entire anomalous trajectory is defined as Definition 6.

Definition 6: The points set of an anomalous trajectory are defined by $TRn_i = \{p \in TR_i |$, p denotes an abnormal point$\}$. $|TRn_i|$ denotes the number of points in TRn_i. The entire trajectory is abnormal if it satisfies Eq. (6), where S denotes a parameter given by a user.

$$\frac{|TRn_i|}{|TR_i|} \geq S \quad (6)$$

Usually, we regard a trajectory containing more than 70% of the erroneous points as an abnormal trajectory, i.e., the points contained therein are all abnormal points. So here we set S to 0.7. The total process of abnormal detection algorithm is illustrated in Fig. 2.

3 Experiments

In this section, we evaluate the proposed method using real dataset and compare it with other well-known approaches in the field of anomaly detection.

Algorithm Abnormal Detection based on Line Segment Clustering

Input: A set of trajectories $T=\{TR_1, TR_2, ..., TR_{numtra}\}$
Output: A set of abnormal $Ap=\{ p_1, p_2,..., p_{numAp} \}$
Algorithm:
/*Pre-processing Phase*/
01: for each TR in T do
02: $V_r = \max(TR._{Sensor\_Data}) - \min(TR._{Sensor\_Data})$
03: $V = V \cup V_r$
04: end
05: $M_r = \text{media}(V)$
06: $M_d = \text{media}(T._{Sensor\_Data})$
07: for each p in T do
08: $p._{Pressure} = p._{Pressure} * 10 / 2000$
09: $p._{Sensor\_Data} = (p._{Sensor\_Data} - M_d + M_r / 2) * 10 / M_r$
normalize the $p._{Pressure}$ and $p._{Sensor\text{-}Data}$;
10: end
11: for each TR in T do
12: Execute Approximate Trajectory Partitioning;
Get a set L of line segments;
13: end
/*Clustering Phase*/
14: Execute Line Segment Clustering for $\mathcal{D}$;
Get a set of O of clusters as the result;
15: for each L in $\mathcal{D}$ do
16: if $L \notin O$
17: $AL = AL \cup L$
18: end
19: end
/*Restore Phase*/
20: for each L in AL do
21: Mark the raw points in T as abnormal points according to the range between the $p_{c}._{Global\text{-}Index}$ And $p_{c+1}._{Global\text{-}Index}$ in L as the result;
22: end
23: for each TR in T do
24: if $|TRn| / |TR| \geq S$
25: $Ap = Ap \cup TR$
TR is an abnormal trajectory;
26: else
27: $Ap = Ap \cup TRn$
28: end
29: end

Fig. 2. Abnormal Detection Algorithm based on Line Segment Clustering.

3.1 Experimental Setting

The experimental data are derived from the Argo dataset in Argo China [7]. We randomly choose the profile data from January to March 2017 with longitude from 25 to 28 and latitude from 32.5 to 35.5. There are 224 profiles consisting of 69,499 points, and the salinity data is selected as the sensor data. Our experiments are conducted on Intel i5

3.0 GHz PC with 4 G Byte of main memory, and the simulation software is Matlab on Windows7 OS.

In order to investigate the validity of the proposed method, the unsupervised methods DBSCAN [8] and LOF [9] are used for comparisons. We use these three methods to calculate the same dataset, and compare the calculated predicted value with the real one. Four metrics are introduced to compare the difference between predicted value and real value. True Positive (*TP*) denotes the number of real abnormal points that are correctly identified as the abnormal points, False Negative (*FN*) denotes the number of real abnormal points that are not recognized, False Positive (*FP*) denotes the number of real normal points that are predicted as the abnormal points, and True Negative (*TN*) denotes the number of real normal points that prediction correct.

The FPR (false positive rate) metric indicates the ratio of number of normal points identified by the error to the number of all normal points, which is defined as,

$$\text{FPR} = \frac{FP}{FP + TN} \tag{7}$$

The TPR (true positive rate) metric represents the ratio of points correctly identified as abnormal with real abnormal points, which is defined as,

$$\text{TPR} = \frac{TP}{TP + FN} \tag{8}$$

3.2 Simulation Results

In order to determine the influence of parameters Ɛ (Eps) and *MinLns* on the final results, we compare results with different values. First, we fix the value *MinLns*, and then we calculate the corresponding prediction accuracy by taking different Ɛ value, finally we change *MinLns* value sequentially to obtain multiple curves. The results are illustrated in Fig. 3.

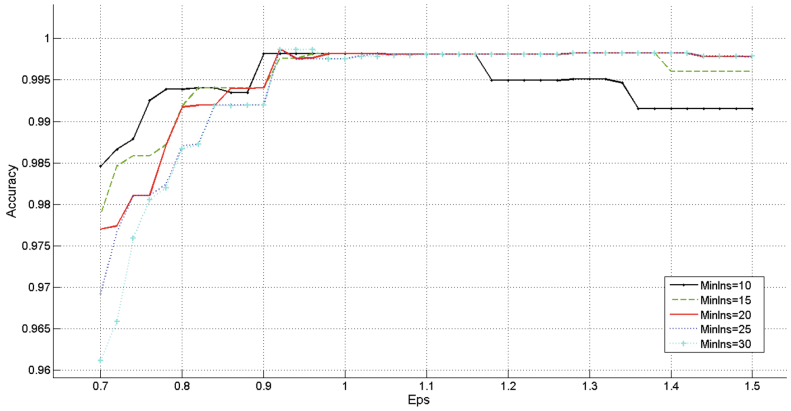


Fig. 3. Prediction accuracy with different *MinLns* and Eps.

When *MinLns* is at 10, the accuracy increases first and then decreases with increasement of Ɛ. But the curve gradually becomes stable as the value of *MinLns* increases. When Ɛ is between 1 and 1.16, the accuracy of different *MinLns* values is same approximately, so herein we set *MinLns* and Ɛ to 20 and 1.1 respectively. In order to compare ROC (Receiver Operating Characteristic) curves of different methods, we set *MinLns* to 20 and change the value of Ɛ to figure ROC curves. Figure 4 illustrates the comparison of ROC curves for LOF, DBSCAN and our proposed method. The area AUC (Area Under Curve) under the ROC curve of LOF, DBSCAN and proposed method are 0.8924, 0.9424 and 0.9783, respectively. As larger the AUC value is, the better the performance of classifier. Therefore, the proposed method has better performance than the other two methods.

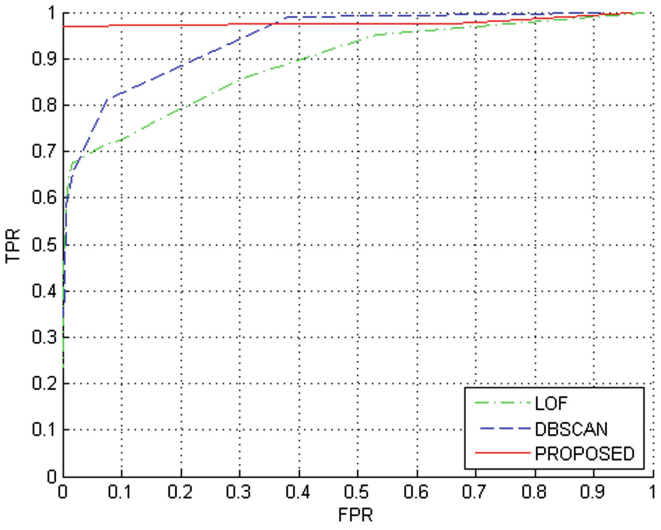
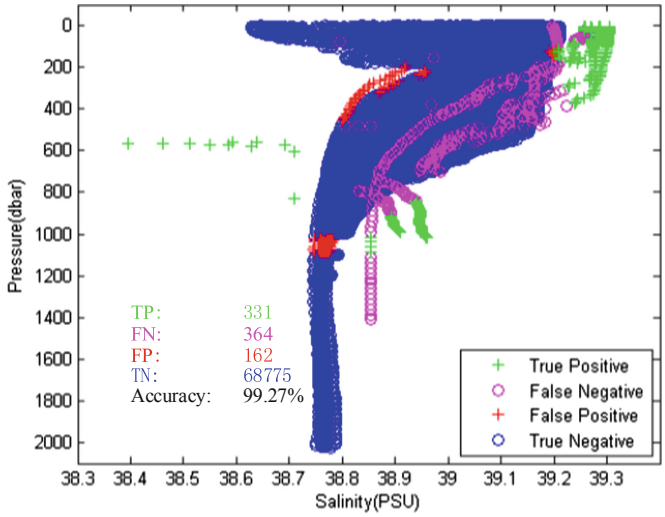


Fig. 4. ROC curve for DBSCAN, LOF and proposed method.

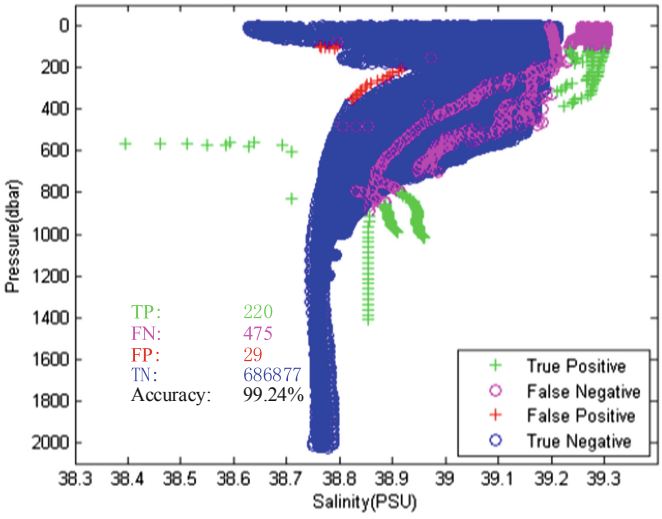
The highest accuracy of the results of LOF and DBSCAN methods are compared in Fig. 5, where green '+' are *TP* points, magenta 'o' are *FN* points, red '+' are *FP* points, and blue 'o' are *TN* points.

There are some anomaly profiles such as entire anomaly profiles and local anomaly profiles in the dataset. In Fig. 5(a), the LOF method can detect some abnormal data points but cannot identify the entire abnormal profile, e.g., there are 162 normal data points are recognized as erroneous points. The DBSCAN method can only detect 220 abnormal data points in Fig. 5(b), which is lower than that of LOF method, but the number of normal data points recognized as errors is only 29. Both of the above methods are directly processed with raw data points, so they are easily affected by the local density of data points. Moreover, change trend of data points has not been considered together.

Our results are better than that of two methods because it not only can identify the entire profile anomaly, but also detect some local anomaly profile hidden in raw dataset according to the change trend of data points so as to improve detection accuracy.

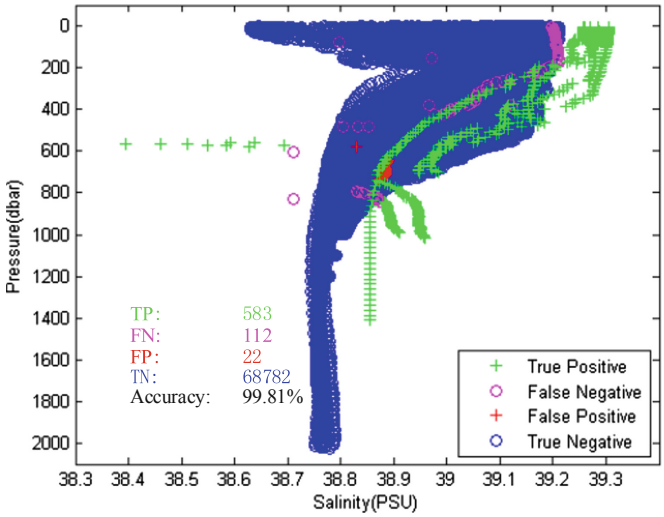


(a) LOF method



(b) DBSCAN method

Fig. 5. Detection results comparison (Color figure online)



(c) Proposed method

Fig. 5. (*continued*)

4 Conclusion

To solve the anomalies in the profile which caused by uncertainties factors, we propose an improved trajectory clustering method to discriminate Argo data' normal and abnormal. We take the trend of data points into account, and data points with the same trend are regarded as a line segment, so it can detect the abnormal points hidden in total raw dataset very well. Compared with LOF and DBSCAN methods, the proposed algorithm has better performance in terms of detection accuracy. However, it is not particularly effective for some of the mutation points, because these points are not used as characteristic points in the partitioning step and are replaced by other normal line segments. In our future works, we will apply different trajectory partition methods to improve detection accuracy. Moreover, more practical data will be used to measure our methods.

Acknowledgment. The authors would like to thank the anonymous reviewers for their helpful and constructive comments that greatly contributed to improving the final version of the paper. These data were collected and made freely available by the International Argo Program and the national initiatives that contribute to it (http://www.argo.net). This research has been partially supported by National Natural Science Foundation of China (No. 61871163 and No. 61801431), Zhejiang Public Welfare Technology Research Project (No. LGF20F010005) and Key Research and Development Program of Hainan Province (ZDYF2017006). Natural Science Foundation of Zhejiang Province (No. LY18F030006) and Open funding of Zhejiang Provincial Key Lab of Equipment Electronics.

References

1. Argo Profile Floats, 10 June 2019. http://www.argo.ucsd.edu [BL/OL]
2. Zhou, Y., Qin, R., Xu, H., Sadiq, S., Yu, Y.: A data quality control method for seafloor observatories: the application of observed time series data in the East China Sea. Sensors **18**, 2628 (2018)
3. Brandsæter, A., Vanem, E., Glad, I.K.: Cluster based anomaly detection with applications in the maritime industry. In: 2017 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC), Shanghai, pp. 328–333 (2017)
4. Kamikawaji, Y., Matsuyama, H., Fukui, K., Hosoda, S., Ono, S.: Decision tree-based feature function design in conditional random field applied to error detection of ocean observation data. In: 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, pp. 1–8 (2016)
5. Wong, A., Keeley, R.: Thierry Carval and the Argo Data Management Team. Argo Quality Control Manual for CTD and Trajectory Data (2019). http://dx.doi.org/10.13155/33951
6. Lee, J.G., Han, J., Wang, K.Y.: Trajectory clustering: a partition-and-group framework. In: Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, pp. 593–604 (2007)
7. Argo Data 10 June 2019. ftp://ftp.argo.org.cn/pub/ARGO/global/core/ [BL/OL]
8. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. KDD **96**, 226–231 (1996)
9. Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. In: Proceedings of the SIGMOD Conference, pp. 93–104 (2000)