IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 19, 2020, accepted March 7, 2020, date of publication March 12, 2020, date of current version March 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2980322

Edge Computing-Based Adaptable Trajectory Transmission Policy for Vessels Monitoring Systems of Marine Fishery

JIE HUANG[1,2], JIAN WAN[1], JIANJUN YU[3], FENGWEI ZHU[2], AND YONGJIAN REN[2]

[1]School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China
[2]School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China
[3]Software Industry Solutions Center, AVEVA Group Plc (Schneider Electric Software Business), Singapore 486057

Corresponding author: Jian Wan (wanjian@hdu.edu.cn)

This work was supported in part by the Key Research and Development Project in Zhejiang Province under Grant 2017C03024.

ABSTRACT Vessel Monitoring Systems (VMS) are extensively used in the world to provide information on the vessel's spatiotemporal distribution, monitor the fishing activities of marine fishery and manage the safety of vessel navigation. In the traditional VMS, there are some deficiencies in the interaction and real-time communication between the land system and marine vessel. We present an edge computing-based adaptable trajectory transmission policy (EC-ATT) for VMS to improve the communication efficiency in this paper. Firstly, a novel VMS framework named EC-VMS is proposed which composed of four layers. Each vessel has an edge computing intelligent node to collect data, process and transmit data. Meanwhile, the edge computing server is set up to enhance collaborative computing between the cloud and the edge, that transmits data through the Beidou navigation satellite system. Secondly, the EC-ATT utilizes the computing power of edge nodes to establish an adaptive data transmission mechanism, which reduces redundant data and satellite communication frequency. Besides, the packet loss feedback mechanism and error checking strategy are used to ensure the reliability of data transmission. The experimental results show that EC-ATT has better performance in typical cases, which not only reduces the average communication time but also strengthens the real-time availability of the VMS.

INDEX TERMS Vessels monitoring systems, edge computing, marine communication, vessel trajectory.

I. INTRODUCTION

Vessel Monitoring Systems (VMS) enable fishery managers to control and monitor fishing activities, which are widely used in many countries over the world. In a typical VMS, the electronic module installed onboard vessels automatically sends data to the land monitoring center through satellite communication. The fishery monitoring center receives and processes the transmitted data to obtain navigation data and relevant information. Utilizing the information of the vessel locations along with the vessel movement characteristics in near-real-time offers many benefits. For examples, it will improve the quality of logbooks for recovery, facilitate estimation of fishery-independent fishing efforts, improve the ability of vessel safety protection, enable effective regional management with an improved understanding of individual vessel behavior and fleet dynamics, and prompt catch/effort reporting [1]. One of the most common, large, and valuable data in VMS is trajectory data. The collection, tracking, estimation, anomaly detection and prediction of marine vessel trajectory are fundamental functions for navigation systems as well as the VMS to improve safety, security, and survivability in marine navigation.

Nevertheless, there are still some deficiencies in the existing VMS, especially in real-time performance and maritime communication, which leads to the inefficient of the system, and the value of vessel trajectory data cannot be mined out to meet some advanced application needs. The existing VMS adopts the centralized computing model, which transmits all the data of fishing vessels to the monitoring center, and then processes and analyzes the data. However, due to the lack of marine communication resources, the data collected by the terminal equipment cannot be all transmitted to the monitoring center, and the communication delay is high, which cannot meet the needs of high real-time applications. For instances:

The associate editor coordinating the review of this manuscript and approving it for publication was Ying Li.

(1) The advancement of marine communication falls behind that on land. The traditional marine communication system provides limited essential services (e.g. ship identification, speed, tonnage, destination, heading, course, location, and positioning, etc.) using VHF radio frequencies such as AIS (Automatic Identification System). It is much more expensive to use inter-ship satellite communication than traditional wireless communication. Therefore, for most small and medium-sized vessels, it is unaffordable [2]. Sensors on vessels will produce a lot of trajectory-dependent data, however, these data cannot be well utilized by cloud centers because of the restrictions on transmission.

(2) To detect vessels in the commission of plausible infringements, fishing activities need to be monitored. This requires near real-time trajectory data acquisition and transmission so that such data can be processed in the cloud immediately to identify suspected infringements in time [3].

To address the aforementioned problems, this paper proposes an edge computing-based adaptable trajectory transmission policy (EC-ATT) for marine fishery vessel monitoring systems.

Firstly, a system framework of intelligent VMS based on edge computing (EC-VMS) is introduced. Each vessel is equipped with an intelligent node to collect data from the various vessel terminals. It can send the data to the server or process data in real-time. EC-VMS uses the BeiDou navigation satellite system (BDS), which has been developed by China, for communication via short messages. Besides, a local edge computing server is set up on some vessels to process the data from nearby vessels. This helps in responding quickly to anomalies at the edge. Meanwhile, the onshore managers can schedule jobs and find anomalies through these edge servers.

Based on the EC-VMS framework, EC-ATT can not only improve the efficiency of vessel trajectory data transmission but also improve the quality of these data received in the cloud. EC-ATT utilizes the computing and communication capabilities of edge nodes to establish a unified adaptive trajectory transmission mechanism in the edge layer and cloud layer. Relevant algorithms are used to reduce redundant data and the number of satellite communications. Packet loss feedback mechanism and error checking strategy are used to ensure the reliability of data transmission.

Finally, an experimental system is set up based on the proposed policy and framework, which proves its effectiveness.

The following three points are the contributions of this paper:

(1) To design a new VMS framework based on edge computing (EC-VMS) to improve the system flexibility and communication efficiency of traditional VMS.

(2) To propose a novel edge computing-based adaptable trajectory transmission policy (EC-ATT) to improve the VMS capability in real-time trajectory data transition and analytics in the context of limited communication.

(3) Achievement of the better performance of EC-ATT in comparison to current ones.

II. RELATED WORK

VMS can provide detailed information on the spatiotemporal distribution and activities of managed vessels. Some countries have mandatory regulations on the use of VMS. In Europe, fishing activities are subject to strict monitoring by law, and vessels over 15 meters in length must provide location information every two hours or less to compare the data transmitted from remote animal sensing [1].

Data transmission delay is an obvious disadvantage of the existing VMS. The sensors on the vessel generate plenty of data, but it is impossible to transmit all the data to the shore in time. So, most previous studies regarding VMS are limited to post data analytics, for examples, to improve fishing efficiency [4], to identify and characterize trips [5], to calculate indices of fishery abundance [6], to detect fishing behavior [7], and to differentiate the type of fishing gear used [8]. Another research topic is the assimilation of various data sources (e.g. vessel detection system data, satellite AIS tracking data, and space-borne high-resolution remote sensing satellite data) from to improve the uniformity of VMS data [9]. These previous works have limitations in the calculation and processing of the shipborne terminal. The performance of VMS, in using satellite communication, remains the challenge due to the limitation of real-time data communication capabilities.

The recent progress of low-bandwidth communication in satellite communication and satellite positioning [10] has shed light on addressing the challenges. For example, BDS, developed by China, possesses the capabilities of time services, short message communication, and high precision positioning, etc. Because communication via short message is relatively cheap, BDS is widely used for marine fishing vessel monitoring in China [11]. Although there are many methods to be used in marine communication, However, the real-time performance and flexibility of the marine information systems still have defects, which lags behind the systems onshore [11]–[14].

Therefore, it is a new method to transfer computing from cloud to edge. This new computing mode is called edge computing [15]. It performs some intelligent processing on the edge of the network, and the calculation takes place near the data source. Therefore, compared with cloud computing, it has some advantages in response time, privacy, data security, bandwidth cost and energy consumption [16]. Some research results have proved the advantages of edge computing [17]–[21]. So, in our EC-VMS, the new technology of edge computing is introduced, which can help us reduce the traffic and speed up the response.

The advances of the shipborne positioning systems have generated a lot of spatiotemporal trajectory data, which represent the moving characteristics of vessels [22]. Many studies for processing [23]–[25], managing [26]–[28], and analysis [29]–[31] trajectory data have been proposed recently, which has promoted the wide application of trajectory data [32]–[36]. Trajectory compression can reduce the data

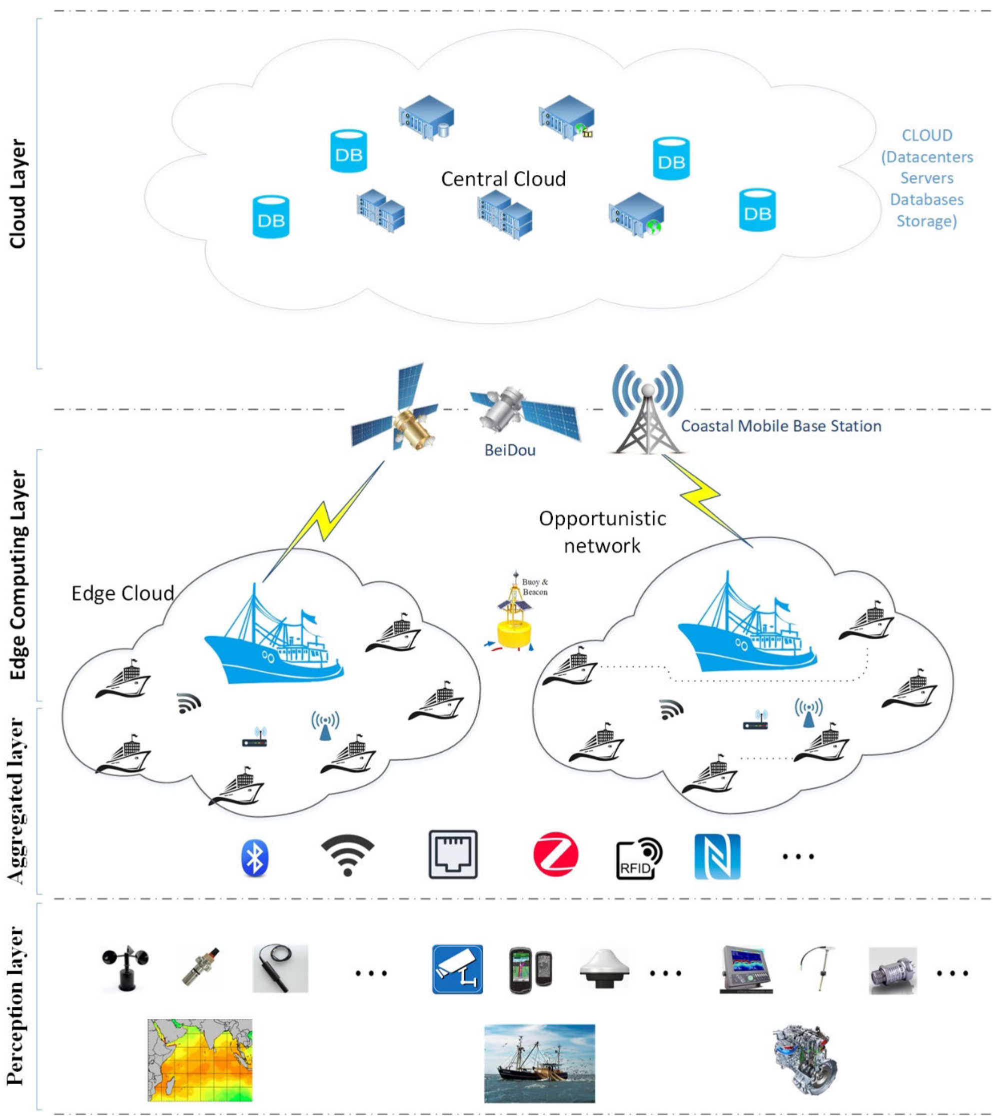


FIGURE 1. The EC-VMS architecture.

traffic of VMS. One trajectory compression strategy is offline compression, which compresses the trajectory after completely obtained. Another strategy is online compression, that is to compress the data immediately when the object moves, to determine whether the newly acquired point should remain in the trajectory [22]. Online trajectory compression considers some key factors such as speed and directions, which is important in our solution. But none of these previous studies used edge computing to improve the utilization of these data in VMS.

In this paper, we propose the EC-ATT strategy and EC-VMS framework, using the latest edge computing and related technologies, making the VMS more intelligent and real-time [38].

III. SYSTEM ARCHITECTURE

A. ARCHITECTURE OVERVIEW

As shown in Figure 1, we designed a four layers architecture of EC-VMS.

1) PERCEPTION LAYER

This layer covers the sensors with their operation systems on board. The vessels have many heterogeneous sensors, some of which have high acquisition frequency and can produce numerous data. The perception layer collects the data of vessel's state, trajectory and ocean environment through these sensors.

2) AGGREGATED LAYER

This layer integrates the data acquired by the perception layer, transforms, preprocesses and centrally stores the data. It is connected with nodes of the perception layer through wired or wireless and adapts to different data providers.

3) EDGE COMPUTING LAYER

This layer establishes an intelligent computing layer on the vessels. It can not only run on a single vessel, but also in the marine ad hoc network. It determines whether the data

TABLE 1. **The data collected in the perception layer.**

Classification	Data content
Fishery production	Materials, fish catch, fishing gear, fishing conditions, etc.
Marine environment	Meteorological, ocean depth, humidity, and salinity, hydrological, SST, etc.
Equipment conditions	Internal network, oil quantity, and engine condition, etc.
Video surveillance	Engine room, deck, pilothouse, etc.
Navigation & Position	Latitude, longitude, speed, heading, time, etc.

received by the aggregation layer is processed locally or forwarded to the cloud.

4) CLOUD LAYER

An onshore management system is deployed in the cloud center, which is defined as cloud layer. It is used to manage all received vessel data form the edge computing layer. All the vessels will be tracked in real-time. It is also used for aid decision making and emergency response.

B. PERCEPTION LAYER

Table 1 lists the data from different sensors and devices in the perception layer. It mainly includes the data of fishery production, marine environment, equipment conditions, video surveillance, and navigation. Various types of sensors can actively perceive data, and their associated back-end has certain data conversion and processing capabilities, but the functions are relatively weak, the platforms cannot communicate with each other, the data standards are not unified, and data integration is difficult. Besides, the RFID technology is used to report the states of the monitored objects with its capability of self-perception.

C. AGGREGATION LAYER

The aggregation layer is a data channel, which gathers all the perception layer data and connects the sensors data through the adapter. It can be used for local network management, data caching, data transmission, node initialization and sensor configuration. In modern vessel, sensor data can be shared through Ethernet network. Data can also be sent through various wireless protocols such as UWB, ZigBee, Bluetooth and Wi-Fi.

The aggregation layer provides the caching, which is used for the temporary storage of data and distributed communication optimization with the perception layer. The aggregated caching receives, processes, and stores the raw sample data from the sensor, and then forwards it when a data request is received. Some other basic vessel data, such as fishery facilities data, marine GIS data, logbook data, crew data and other data, can also be accessed into the aggregation layer if allowed. Besides, the multi-source heterogeneous data (such as structured points from GPS, unstructured data from video monitoring, etc.) processing scheme is established. Data can be exchanged and shared among aggregation layers of different vessels, which provides flexible data support.

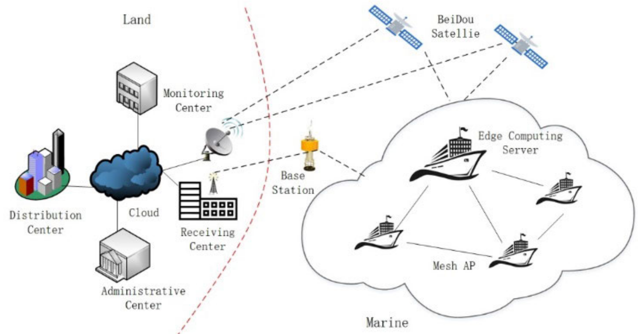


FIGURE 2. **The network of the system.**

D. EDGE COMPUTING LAYER

A series of onboard devices with computing resources, storage resources, and communication resources make up the edge computing layer, which can be seen as a big virtual computer. Based on the aggregation layer, the edge computing layer makes nodes more autonomous and intelligent. Aggregation layer and edge computing layer can coexist in a single vessel or vessels network, and they can have functional overlap, so they can easily establish a connection through the local network. In the EC-VMS, the edge computing layer stores, processes, resamples, calculates and analyzes the received data, and then gives a fast feedback to the aggregation layer.

A Vessels Edge Computing Server (VECS) is designed as the larger edge computing layer, it can specify one vessel as the primary node in the local network, and then manage the other edge computing nodes in the network. It interacts with the edge computing layer of each vessel and performs some advanced tasks. Under the coordination of VECS, vessels can exchange information with each other and perform specific computing tasks cooperatively [38]. VECS is responsible for the determination of the tasks going to the cloud center or the local vessel.

In the EC-VMS (see Figure 2), only a small amount of raw sensor data is transmitted to the land directly. To increase system availability, the edge computing layer relies more on local data processing and analytics. When the vessel encounters an emergency and the cloud center is unable to make the abnormal judgment, danger warning and action instruction in time, the edge layer can assist decision support and replace part of the work of the cloud center.

E. CLOUD LAYER

A cloud layer is used for central processing with the advantage of low cost in computing power and storage. However, the delay caused by inefficient processing and communication links cannot be ignored as the cloud computing systems are deployed onshore. A local server is set up specifically to run the management system to speed up the responses with a reduction in communication delay [39].

When the edge node is abnormal, the cloud layer should respond in time, which is a very important management task. For example, when the vessel makes operations contrary to the normal, or the communication equipment is invalid, the cloud layer can detect and send an alarm in time through the anomaly detection mechanism.

A GIS is used to visualize the spatiotemporal distribution of all vessels in the EC-VMS, as well as corresponding properties including the name, status, location, unique ID, etc. Bright colors are used to indicate that the vessel may be in an abnormal state so that the system manager can quickly find the unusual vessel. Also, the vessel status is updated in time. When a vessel is lost contact within a specified period, it will be highlighted in the GIS showing it's out of touch status. It is similar for a vessel sailing into the fishing prohibited areas, it will be blinked synchronously on the map the warning message will be reported to relevant staff.

IV. EDGE COMPUTING-BASED ADAPTABLE TRAJECTORY TRANSMISSION POLICY (EC-ATT)

Based on EC-VMS architecture, we adopt a transmission policy, named Edge Computing-based Adaptable Trajectory Transmission Policy (EC-ATT) to establish a VMS communication mechanism, which combines the SQUISH (Spatial Quality Simplification Heuristic) trajectory compression algorithm [40], LDR (Linear Dead Reckoning) algorithm [41] and reliable transmission strategy.

Definition 1 (Trajectory Point): The trajectory point P consists of longitude, latitude, and timestamp denoted as $P = (x, y, t)$.

Definition 2 (Observation Trajectory): The Observation trajectory $TR = \{P_1, P_2, \ldots, P_i, \ldots, P_{n-1}, P_n\}\ 1 \le i \le n$ represents the collection of points in chronological order.

Definition 3 (Approximate Trajectory): The approximate trajectory $\boldsymbol{TR'} = \left\{\boldsymbol{P_{c_1}}, \boldsymbol{P_{c_2}}, \ldots, \boldsymbol{P_{c_{m-1}}}, \boldsymbol{P_{c_m}}\right\}, \boldsymbol{1} \le \boldsymbol{c_1} < \boldsymbol{c_2} < \ldots < \boldsymbol{c_m} \le \boldsymbol{n}$ represents the trajectory sequence after simplification. It is a subset of the observed trajectory $\boldsymbol{TR}$.

Definition 4 (Euclidean Distance (ED): Euclidean distance between points $\boldsymbol{P_1}$ and $\boldsymbol{P_2}$ is calculated using Equation 1, as following:

$$ED\,(P_1, P_2) = \sqrt{(P_1 \centerdot y - P_2 \centerdot y)^2 + (P_1 \centerdot x - P_2 \centerdot x)^2} \quad (1)$$

Definition 5 (Synchronous Euclidean Distance (SED)): B' represents the same time mapping point of B on trajectory vector AC, as shown in Figure 3, SED is the Euclidean distance between B and B', which is calculated using Equation 2:

$$\begin{cases} SED(B, \overrightarrow{AC}) = ED\left(B, B'\right) \\ B' \centerdot x = A \centerdot x + \dfrac{B \centerdot t - A \centerdot t}{C \centerdot t - A \centerdot t}(C \centerdot x - A \centerdot x) \\ B' \centerdot y = A \centerdot y + \dfrac{B \centerdot t - A \centerdot t}{C \centerdot t - A \centerdot t}(C \centerdot y - A \centerdot y) \end{cases} \quad (2)$$

SQUISH is an online space quality heuristic trajectory compression algorithm [40]. Different from error-based trajectory compression algorithms such as Sliding Window [42] and Threshold-Guided Sampling [43], SQUISH is based on satisfying compression ratio to keep tracking feature information as much as possible.

SQUISH select the optimal subset tracking points using the strategy of local optimization, and delete redundant ones

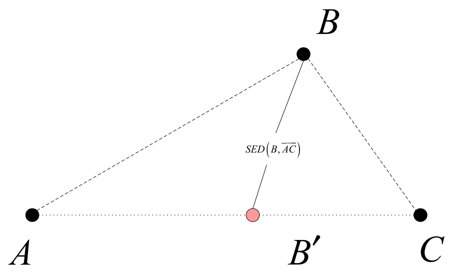


FIGURE 3. Synchronized euclidean distance.

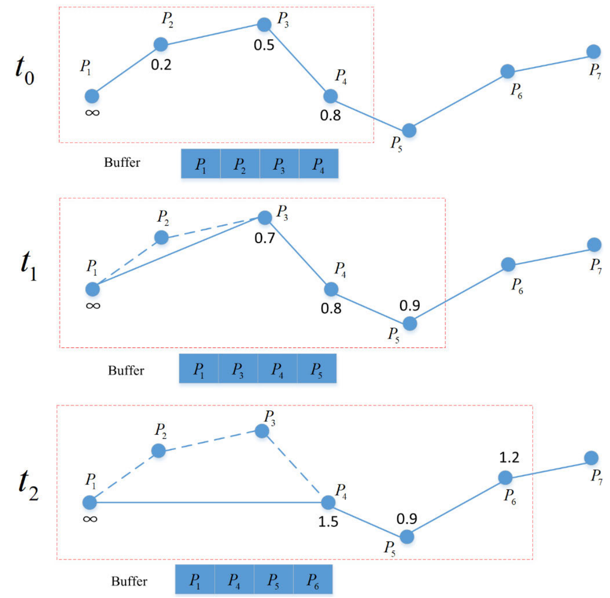


FIGURE 4. The compression process of the SQUISH algorithm.

from the track [40]. Figure 4 demonstrates the compression process ($t_0 \sim t_2$) in using the SQUISH algorithm. The dotted line box indicates the tracking points under processing, with a figure showing the priority of the corresponding point. To determine the priority, every two adjacent points are connected to form line segments. The SED from the point to the segment is then calculated, for example, the priority of P_2 equals to $SED(P_2, \overrightarrow{P_1P_3})$. The smaller the priority, the smaller the SED error caused by deleting the point. The priority of the endpoint is set to infinity to avoid deletion when executing the algorithm. EC-ATT uses the SQUISH to compress the trajectory, because SQUISH can limit the length of the simplified trajectory by setting the buffer size and has good performance.

The LDR algorithm is a kind of navigation technology that can predict the future time position coordinates according to the current time position coordinates, heading, speed and time of the moving object [41]. It is widely adopted in location-based related services such as vehicles, pedestrians, and ships. The dead reckoning can effectively reduce the communication between the mobile object and the server, thereby save communication resources.

The pseudocode of the algorithm about EC-ATT in the edge computing cloud layer is shown in Algorithm 1.

Algorithm 1 EC-ATT (Edge)

```
Input:
    observation trajectory point P_t
    error threshold θ_d
Function:
    send messages
Begin
initial sending queue;
initial uncompressed queue;
while (received data)
    if received a retransmit signal then
        adding missing messages to the sending queue
        based on message number;
    if received the observation trajectory points then
        if the uncompressed queue is empty then
            estimate trajectory points by LDR;
            if estimated value greater than threshold then
                add observation point to uncompressed
                queue;
        else add observation point to compressed queue;
    if it's time window for data transmission then
        if sending queue is not empty then
            send message;
        else if uncompressed queue is not empty then
            compress trajectory by SQUISH;
            generate message into sending queue;
            send message [37];
End
```

The edge computing layer of EC-ATT consists of three modules: trajectory simplification, position tracking, and data retransmission. The position tracking and trajectory simplification are carried out synchronously, to ensure the processed data can be sent in time [44]. The fishing vessel operation has certain randomness. The crew will change the fishing route according to their own experience and the surrounding environment. Therefore, the Gaussian Regression Process, Neural Networks, and other machine learning algorithms are not applicable in the current scene. LDR algorithm can predict the trajectory only according to the base point and velocity vector, which is simple and efficient. It is used in the edge computing layer to predict the fishing vessel positions, written as:

$$\vec{\mathrm{l}}\,(\mathrm{t}) : \mathrm{t} \to \mathrm{l_b} \,.\, \vec{\mathrm{p}} + (\mathrm{t} - \mathrm{l_b} \,.\, \mathrm{t})\vec{\mathrm{l_V}} \quad (3)$$

where $\vec{l_V}$ is the velocity vector, l_b is the prediction base point [45]. Suppose P_t is the prediction point and P_t' is the observation point, as long as their distance does not exceed the given error threshold θ_d, denoted as $\mathrm{ED}\left(P_t, P_t'\right) < \theta_d$, no update message will be generated in the edge layer. The monitoring center on-shore uses the predicted trajectories to replace the observed ones. The system does not transmit any data through the satellite meanwhile. Otherwise, it is considered that the prediction is wrong and the velocity vector and prediction base point will be revised.

To deal with transmission failure or data distortion caused by the potential unreliability of Beidou satellite transmission, the edge layer is required to verify the communication receipt from the monitoring center. In case of failure, the edge layer will retransmit the lost packet according to the sent message sequence number and that in the communication receipt as well. If the number of the message in the communication receipt is k, and the number of the transmitted message is i ($i >= k$), the lost messages $k, k+1, \ldots, i$ will be retransmitted. The system adopts the FIFO strategy to queue up the messages under retransmission. When the time window for sending messages appears (i.e. the time has passed a minimum communication interval), the message is taken out from the queue and transmission. If the communication receipt is not received (i.e. the aforementioned situation does not occur) the recent generated update message will be transmitted directly.

When the observation trajectory *TR* changes frequently, the SQUISH is used to extract a sub-trajectory *TR'* from *TR*, and the length of *TR'* should match the BDS packet. The update message along with the *TR'* will be transmitted to the cloud layer when the transmission condition is satisfied.

The pseudocode of the algorithm about EC-ATT in the cloud layer is shown in Algorithm 2.

Algorithm 2 EC-ATT (Cloud)

```
Input:
    message m
Function:
    correction trajectory
Begin:
if message distortion then
    send a receipt
else message decoding
if network packet loss then
    send a receipt
else
    correction trajectory
    update (l_b, l_V)
End
```

The cloud layer mainly adopts both error checking strategy and packet loss feedback mechanism to ensure the reliability of Beidou satellite communication.

(1) Error check strategy: When the message is received, the cloud layer will firstly run XOR processing on all bytes except for the check digit, and then compare it with the check code. If they are the same, the transmission is fidelity. Otherwise, the transmission is discarded and the edge layer will be notified to retransmit the message.

(2) Packet loss feedback mechanism: For transmitting the fidelity data, the cloud layer compares the received message sequence number with the current expected one. If they are the same, it is considered that there is no packet loss. Otherwise, the edge layer will be notified to retransmit the data.

The number of satellite communications will be reduced to save satellite communication resources. The cloud layer

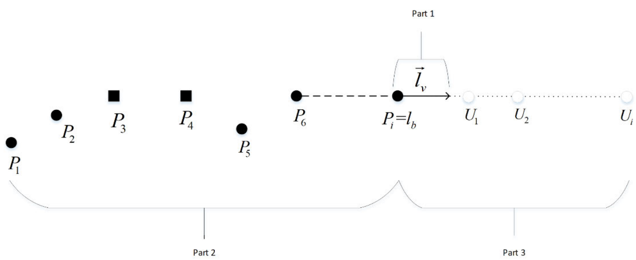


FIGURE 5. The cloud layer storage content.

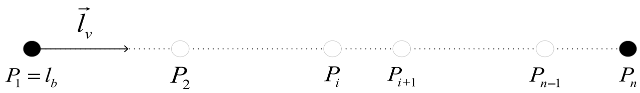


FIGURE 6. Prediction trajectory simplification.

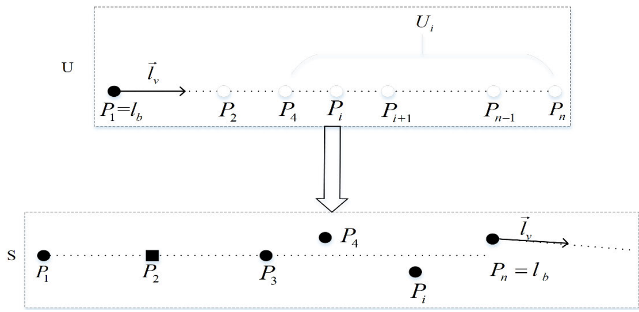


FIGURE 7. Prediction trajectory correction process.

will check the occurrence of the above transmission abnormal situations. If anyone occurs, the sequence number of the message that is currently expected to be received will be sent to the edge, so the message will be re-sent. Otherwise, the cloud layer will correct the stored trajectory data according to the content of the message. As shown in Figure 5, the simplified trajectory stored in the cloud layer consists of three parts, as follows:

(1) The predicted base point l_b and the velocity vector $\overrightarrow{l_V}$ constitute the first portion of the stored content. The cloud layer needs to update l_b and $\overrightarrow{l_V}$ according to the received message, and perform trajectory prediction based on the latest l_b and $\overrightarrow{l_V}$.

(2) The approximate trajectory TR' composed of $\{P_1, P_2 \ldots, P_i, \ldots, P_n\}$ constitutes the second part of the stored content. Specifically, TR' represents a set of approximate trajectory points arranged chronologically $(P_i \centerdot t \geq P_{i-1} \centerdot t)$ since the fishing vessel was in operation. The black dots indicate the trajectory points compressed by the SQUISH algorithm in the edge computing layer, and the black square is trajectory point calculated by the LDR algorithm in the cloud layer. Different from the predicted trajectory points of the third part, these trajectory points are corrected, that is, the prediction error is assumed to be less than the specified threshold θ_d. At the same time, since the continuous predicted trajectory points are calculated according to the same l_b and $\overrightarrow{l_V}$, only the first and last trajectory points are stored to reduce the redundant information and improve the query speed. The deleted trajectory points are possible to be restored according to the first and last trajectory points and time without any loss of precision (see Figure 6).

(3) The predicted trajectory S composed of $\{U_1, U_2 \ldots, U_i, \ldots, U_n\}$ constitutes the third portion of the stored content. U_i is a trajectory sequence calculated based on the latest predicted base point l_b and velocity vector $\overrightarrow{l_V}$. and. As shown in Figure 7, it is necessary to correct S according to the content of the message sent from the edge layer considering the cloud layer cannot guarantee the accuracy of U_i.

A real-time query function of the trajectory is developed to better supervise the fishing vessels. The cloud layer can query the position of all the fishing vessels at any time t. According to the predicted base time $P_n \centerdot t$ and the query time t, the query can be performed in the following two ways:

(1) $t \leq P_n \centerdot t$: obtained by linear interpolation of trajectory points P_i and P_{i+1}, where $P_i \centerdot t \leq t \leq P_{i+1} \centerdot t$.

(2) $t \leq P_n \centerdot t$: obtained according to LDR prediction.

V. RESULTS ANALYSIS

A. EXPERIMENTAL SETUP

The experimental data comes from the trajectories of fishing vessels near the East China Sea collected in a VMS, which manages more than 5000 vessels and is deployed in Zhoushan City, Zhejiang Province, China. The sampling time of the trajectory data is 30 seconds, which is generated by the shipboard BDS intelligent terminal. The data transmission frequency limit of BDS is 60 seconds. The trajectory mainly contains information on latitude, longitude, time, and device number. In this VMS, we installed edge computing nodes on four vessels and collected the trajectories data from Mar. to May. 2018, a total of 1,018,412 points. From figure 8, we can see the distribution characteristics of the four trajectories data.

Because of the instability of the shipborne positioning system in the marine environment, there may be abnormal points in the trajectory data. In our experiment, the mean speed of the fishing vessels is obtained by calculating the distance between adjacent trajectory points, and the trajectory points whose mean speed is greater than 20 knots are deleted. At the same time, due to the unexpected factors in the transmission and storage system, the trajectory data set will contain duplicate data, and these duplicate trajectory points will also be deleted in our experiment.

This experiment verifies the EC-ATT algorithm through the Beidou simulator (BDSim), which simulates the restrictions on minimum transmission interval and message length of Beidou protocol. When the length of data to be sent oversteps the protocol, some data will be discarded. Meanwhile, messages are not allowed to be sent when the minimum transmission interval is not reached. EC-ATT uses an error checking strategy and packet loss feedback mechanism to ensure the reliable transmission of data. For the possible abnormal situations, our experiment simulates data distortion and packet loss respectively.

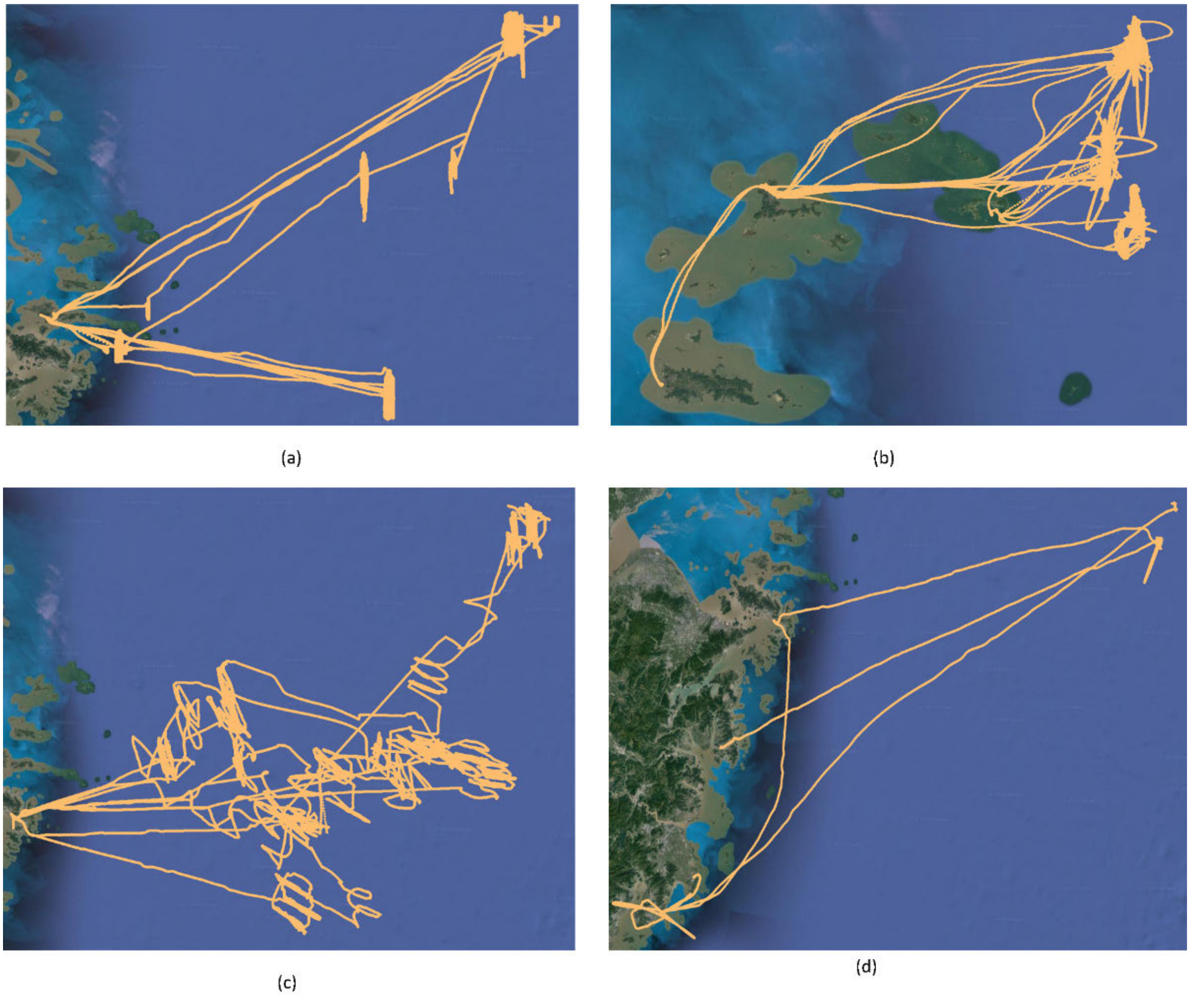


FIGURE 8. **Spatial distribution of four vessels' trajectory.**

B. EXPERIMENTAL RESULTS

Figure 9 shows the process of applying EC-ATT to correct the trajectory data when an abnormal transmission occurs. The blue line indicates the complete trajectory data collected by the shipborne terminal equipment. The green line indicates the trajectory data stored by the cloud layer under normal communication conditions. The red line indicates the trajectory data stored under the abnormal communication condition. Our experiment simulates data distortion during the second transmission and simulates packet loss during the fourth transmission. It is obvious that due to an abnormality in the transmission process, the edge layer cannot transmit the correct update information in time. Hence, the cloud layer continues to perform trajectory prediction based on invalid parameters, thereby causing the red trajectory to deviate from the correct one. As communication is gradually becoming a normal state, the edge layer retransmits the lost data, and finally, the red trajectory gradually approaches the blue trajectory. In the case of normal communication, since the cloud center can update the data in time, the green track always approximates the blue one.

The experiment analyzes the trajectory data from three aspects: trajectory quality, transmission times and real-time performance. The traditional Fixed-interval Trajectory Transmission Policy (FITT) of VMS transmits data at fixed time intervals. The comparison of EC-ATT transmission time with FITT is shown in Figure 10 under the conditions of 30, 50, and 70 meters threshold. The threshold refers to θ_d in the above algorithm 1. The vertical axis is communications number, and the horizontal axis is minimum communication intervals. With the increasing in communication intervals, the communication number of FITT and EC-ATT will decrease.

When the communication interval increases, the probability of EC-ATT to accurately predict all observation trajectory points will be reduced. Therefore, when reaching the communication window, EC-ATT needs to transmission, so the number of communications will close to that of FITT. If the ED from prediction point to observation one is within a given error threshold, the prediction is successful, therefore the prediction success rate goes up with the increasing error threshold, and the number of communications will decrease. Under different error thresholds and

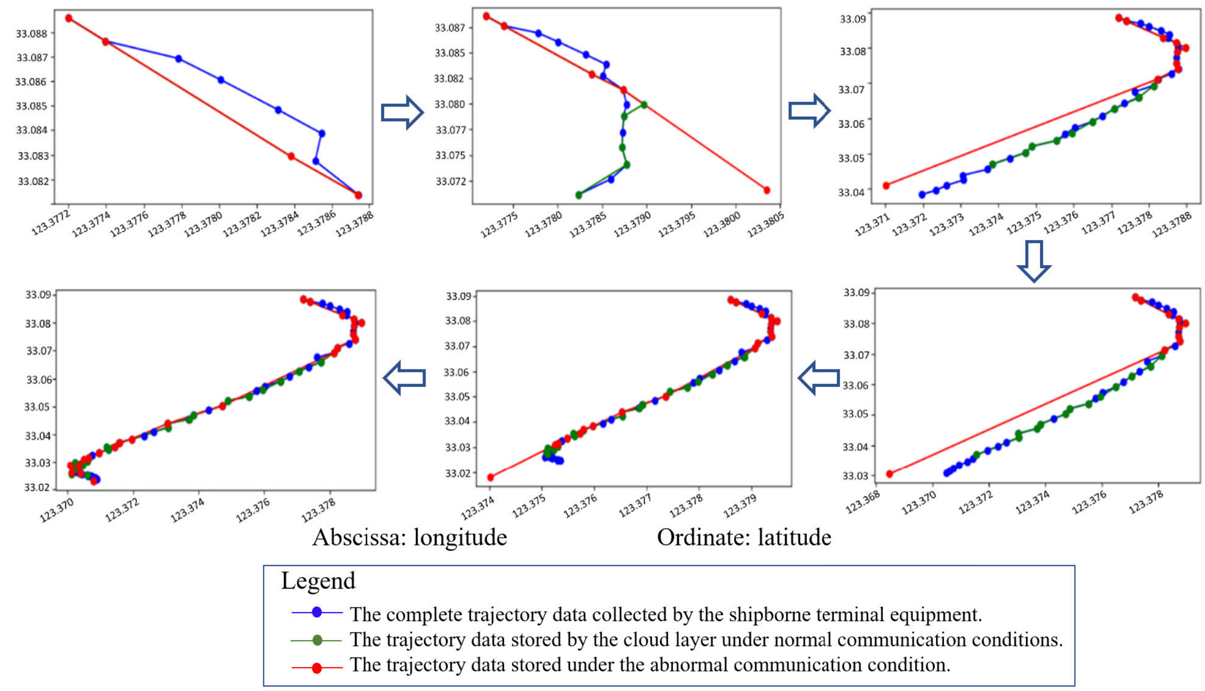


FIGURE 9. Transmission exception correction process.

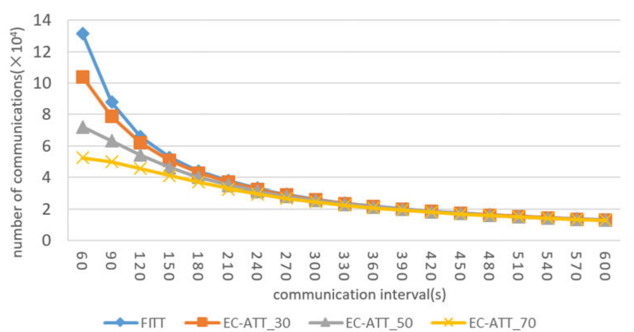


FIGURE 10. Comparison of transmission times.

communication intervals, EC-ATT has fewer communication times than FITT. It shows EC-ATT is superior to FITT in saving communication resources. Especially in the case of commonly used 50-meter threshold and 60-second transmission interval, the communication traffic is decreased by 45.22%.

We also compare EC-VMS with FITT in real-time performance. Take trajectory query as an example, FITT transmits data at regular intervals. If the cloud layer receives data at time t1, it needs to receive data at time t1+1 before querying the trajectory points within the two transmission intervals. Therefore, the minimum delay time of FITT query is 0 second, the maximum delay time is the transmission time interval Δt, and the average delay time is $\Delta t/2$ seconds. EC-VMS adopts a different operation mechanism, which can be used for real-time queries. The disadvantage is that when the cloud layer received the update packet, the predicted trajectory will be corrected, resulting in different query results before and after, but this error is acceptable.

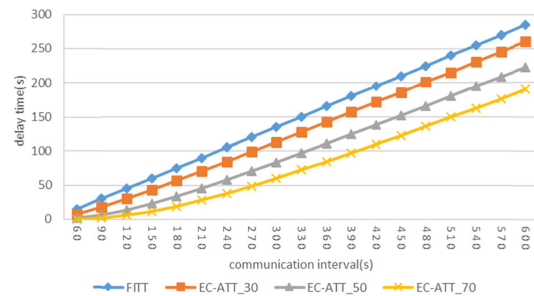


FIGURE 11. Comparison of real-time performance.

To further illustrate the real-time performance of EC-VMS, we make statistics on the correction delay time. From Figure 11, the delay time of EC-ATT enlarges with the increasing of the minimum communication time interval. This is because when the interval is large, EC-ATT cannot send update packet in time, resulting in the longer delay. Under the same communication condition, if the error threshold is set higher, the real-time performance will be better, because the trajectory points that need to be corrected will be reduced. Considering the correction time of EC-ATT is significantly shorter than FITT. It can be concluded that the EC-ATT is better than FITT in real-time performance.

To compare the EC-ATT and FITT in data quality, the Average of Pairs Distance (APD) is used. Given trajectories $A\{a_1, a_2, \ldots, a_n\}$ and $B = \{b_1, b_2, \ldots, b_n\}$, APD represents the mean *ED* of corresponding points belonging to the two trajectories as following:

$$\mathrm{APD}\,(\mathrm{A}, \mathrm{B}) = \frac{1}{\mathrm{n}} \times \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{ED}(\mathrm{a}_1, \mathrm{b}_1) \tag{4}$$

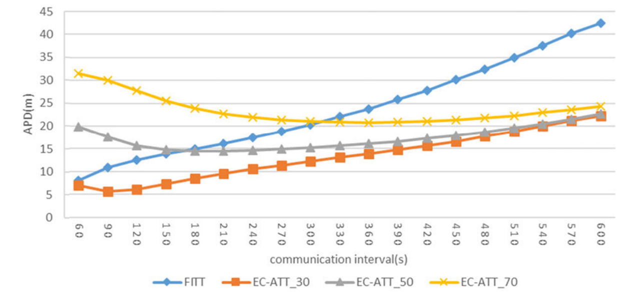


FIGURE 12. **Comparison of trajectory quality.**

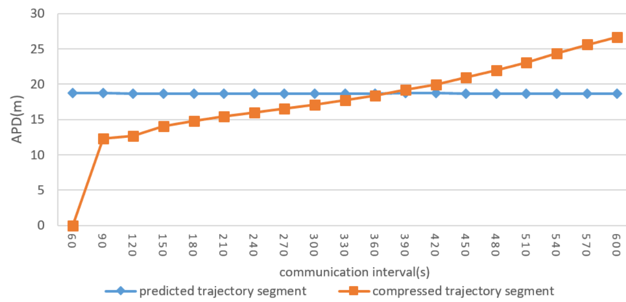


FIGURE 13. **Comparison of quality between predicted and compressed track segments under EC-ATT_30.**

In our experiment, A is the original observation trajectories and B is the trajectories queried in VMS.

We compared EC-ATT with FITT when the error thresholds were set to 30 meters, 50 meters, and 70 meters.

As shown in Figure 12, the APD of FITT enlarges gradually with the increase of minimum communication interval. This is because FIFM transmits data at a fixed communication interval, and the larger the interval, the fewer trajectory points are transmitted. The ADP of EC-ATT shows the characteristic of decreasing first and then increases gradually. The initial falling trend owing to lots of points are transmitted incorrectly when the communication interval is small, which requires to be corrected delaying the response. Therefore, in this stage, the frequency of LDR is much higher than that of SQUISH, which makes the error of LDR much greater than that of SQUISH.

As shown in Figure 13 and Figure 14, when the communication interval raises, the number of points produced by LDR goes down, which makes APD decrease. When the increasing of interval continued, the points generated by SQUISH goes up, and the ADP increase due to the error generated by compression goes up. Also, the error threshold of EC-ATT will affect the query error. The experiments show that when the error threshold is 30 meters, EC-ATT is significantly better than FITT in trajectory quality.

Figure 15 shows the comparison of normalized EC-ATT query error, number of communications, and delay time for different thresholds. The number of communications and delay times are negatively correlated with the increase of distance thresholds. It implies that the greater the threshold, the better the real-time performance and the more communication resources are saved. However, the query error and

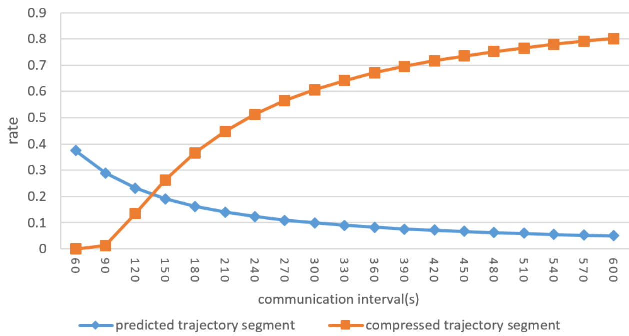


FIGURE 14. **Comparison of the proportion of predicted and compressed track segments under EC-ATT_30.**

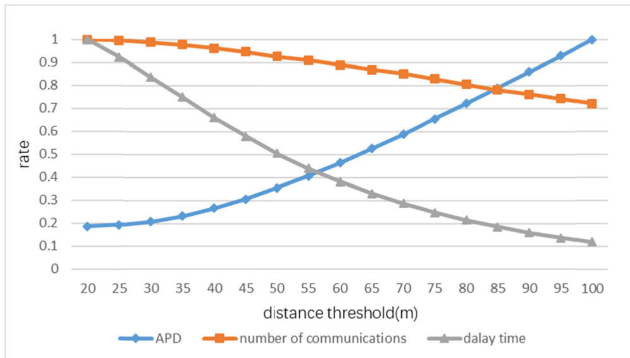


FIGURE 15. **The comprehensive situation of EC-ATT under different thresholds.**

the distance threshold have a positive correlation, that is, the smaller the threshold, the higher the trajectory quality.

VI. CONCLUSION

To improve the real-time performance and increase the marine communication efficiency of traditional VMS, an edge computing-based adaptable trajectory transmission policy EC-ATT is proposed. Firstly, a new VMS framework based on edge computing named EC-VMS is designed. EC-VMS establishes an intelligent node on each vessel to collect data, process and transmit data in real-time. Then, a VECS is established to process the data of all nodes in the jurisdiction in real-time, including the state value and position of the vessels. Hence, VECS enables faster collaborative computing between the cloud and the edge. Secondly, the EC-ATT is designed based on the SQUISH algorithm, LDR algorithm, and reliability transmission strategy, according to the characteristics of Beidou satellite transmission. We have installed some experimental equipment on an existing VMS, which runs in Zhoushan City, China. The experimental results demonstrate the EC-ATT is better than the original trajectory transmission policy of VMS in terms of real-time, usability, and efficiency. We will optimize the EC-VMS framework in more detail and explore more advanced analytics methods of vessel data in future work.

REFERENCES

[1] S. C. Votier, S. Bearhop, M. J. Witt, R. Inger, D. Thompson, and J. Newton, "Individual responses of seabirds to commercial Fisheries revealed using GPS tracking, stable isotopes and vessel monitoring systems," *J. Appl. Ecol.*, vol. 47, no. 2, pp. 487–497, Apr. 2010.

[2] J. Lee, A. B. South, and S. Jennings, "Developing reliable, repeatable, and accessible methods to provide high-resolution estimates of fishing-effort distributions from vessel monitoring system (VMS) data," *ICES J. Mar. Sci.*, vol. 67, no. 6, pp. 1260–1271, Sep. 2010.

[3] E. Ahmed and M. H. Rehmani, "Mobile edge computing: Opportunities, solutions, and challenges," *Future Gener. Comput. Syst.*, vol. 70, pp. 59–63, May 2017.

[4] J. T. Watson, A. C. Haynie, P. J. Sullivan, L. Perruso, S. O'Farrell, J. N. Sanchirico, and F. J. Mueter, "Vessel monitoring systems (VMS) reveal an increase in fishing efficiency following regulatory changes in a demersal longline Fishery," *Fisheries Res.*, vol. 207, pp. 85–94, Nov. 2018, doi: 10.1016/j.fishres.2018.06.006.

[5] J. T. Watson and A. C. Haynie, "Using vessel monitoring system data to identify and characterize trips made by fishing vessels in the united states north pacific," *PLoS ONE*, vol. 11, no. 10, Oct. 2016, Art. no. e0165173, doi: 10.1371/journal.pone.0165173.

[6] N. D. Ducharme-Barth, K. W. Shertzer, and R. N. M. Ahrens, "Indices of abundance in the gulf of mexico reef fish complex: A comparative approach using spatial data from vessel monitoring systems," *Fisheries Res.*, vol. 198, pp. 1–13, Feb. 2018, doi: 10.1016/j.fishres.2017.10.020.

[7] E. N. de Souza, K. Boerder, S. Matwin, and B. Worm, "Improving fishing pattern detection from satellite AIS using data mining and machine learning," *PLoS ONE*, vol. 11, no. 7, Jul. 2016, Art. no. e0158248, doi: 10.1371/journal.pone.0158248.

[8] M. I. Marzuki, P. Gaspar, R. Garello, V. Kerbaol, and R. Fablet, "Fishing gear identification from vessel-monitoring-system-based fishing vessel trajectories," *IEEE J. Ocean. Eng.*, vol. 43, no. 3, pp. 689–699, Jul. 2018.

[9] N. Longépé, G. Hajduch, R. Ardianto, R. D. Joux, B. Nhunfat, M. I. Marzuki, R. Fablet, I. Hermawan, O. Germain, B. A. Subki, R. Farhan, A. D. Muttaqin, and P. Gaspar, "Completing fishing monitoring with spaceborne vessel detection system (VDS) and automatic identification system (AIS) to assess illegal fishing in indonesia," *Mar. Pollut. Bull.*, vol. 131, pp. 33–39, Jun. 2018, doi: 10.1016/j.marpolbul.2017.10.016.

[10] R. Al-Zaidi, J. Woods, M. Al-Khalidi, K. M. A. Alheeti, and K. McDonald-Maier, "Next generation marine data networks in an IoT environment," in *Proc. 2nd Int. Conf. Fog Mobile Edge Comput. (FMEC)*, Valencia, Spain, May 2017, pp. 50–55.

[11] C. Lu, X. Li, T. Nilsson, T. Ning, R. Heinkelmann, M. Ge, S. Glaser, and H. Schuh, "Real-time retrieval of precipitable water vapor from GPS and BeiDou observations," *J. Geodesy*, vol. 89, no. 9, pp. 843–856, Sep. 2015.

[12] Y. Zhang, S. Chen, Z. Hong, Y. Han, B. Li, S. Yang, and J. Wang, "Feasibility of oil slick detection using BeiDou-R coastal simulation," *Math. Problems Eng.*, vol. 2017, Feb. 2017, Art. no. 8098029, doi: 10.1155/2017/8098029.

[13] F. Yu, X. Hu, S. Dong, G. Liu, Y. Zhao, and G. Chen, "Design of a low-cost oil spill tracking buoy," *J. Mar. Sci. Technol.*, vol. 23, no. 1, pp. 188–200, Mar. 2018.

[14] L. N. Wang, L. L. Li, and R. Qiu, "Edge computing-based differential positioning method for beidou navigation satellite system," *KSII Trans. Internet Inf. Syst.*, vol. 13, no. 1, pp. 69–85, Jan. 2019.

[15] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, Jan. 2017.

[16] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[17] E. Zeydan, E. Bastug, M. Bennis, M. A. Kader, I. A. Karatepe, A. S. Er, and M. Debbah, "Big data caching for networking: Moving from cloud to edge," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 36–42, Sep. 2016.

[18] H. H. Gao, Y. C. Duan, L. X. Shao, and X. B. Sun, "Transformation-based processing of typed resources for multimedia sources in the IoT environment," *Wireless Netw.*, vol. 2019, Nov. 2019, doi: 10.1007/s11276-019-02200-6.

[19] T. Taleb, S. Dutta, A. Ksentini, M. Iqbal, and H. Flinck, "Mobile edge computing potential in making cities smarter," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 38–43, Mar. 2017.

[20] Y. Y. Yin, "QoS prediction for service recommendation with deep feature learning in edge computing environment," *Mobile Netw. Appl.*, vol. 2019, pp. 1–9, Apr. 2019, doi: 10.1007/s11036-019-01241-7.

[21] G. Premsankar, M. Di Francesco, and T. Taleb, "Edge computing for the Internet of Things: A case study," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1275–1284, Apr. 2018.

[22] Y. Zheng, "Trajectory data mining: An overview," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, p. 29, May 2015.

[23] L. Zhao, G. Shi, and J. Yang, "Ship trajectories pre-processing based on AIS data," *J. Navigat.*, vol. 71, no. 5, pp. 1210–1230, Sep. 2018.

[24] L. Zhang, Q. Meng, Z. Xiao, and X. Fu, "A novel ship trajectory reconstruction approach using AIS data," *Ocean Eng.*, vol. 159, pp. 165–174, Jul. 2018.

[25] L.-Z. Sang, A. Wall, Z. Mao, X.-P. Yan, and J. Wang, "A novel method for restoring the trajectory of the inland waterway ship by using AIS data," *Ocean Eng.*, vol. 110, pp. 183–194, Dec. 2015.

[26] H. Jeung, H. Lu, S. Sathe, and M. L. Yiu, "Managing evolving uncertainty in trajectory databases," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 7, pp. 1692–1705, Jul. 2014.

[27] B. Zheng, H. Wang, K. Zheng, H. Su, K. Liu, and S. Shang, "SharkDB: An in-memory column-oriented storage for trajectory analysis," *World Wide Web*, vol. 21, no. 2, pp. 455–485, Mar. 2018.

[28] S. Ke, J. Gong, S. Li, Q. Zhu, X. Liu, and Y. Zhang, "A hybrid spatio-temporal data indexing method for trajectory databases," *Sensors*, vol. 14, no. 7, pp. 12990–13005, Jul. 2014.

[29] L. Zhao and G. Shi, "A trajectory clustering method based on douglas-peucker compression and density for marine traffic pattern recognition," *Ocean Eng.*, vol. 172, pp. 456–467, Jan. 2019.

[30] K. Sheng, Z. Liu, D. Zhou, A. He, and C. Feng, "Research on ship classification based on trajectory features," *J. Navigat.*, vol. 71, no. 1, pp. 100–116, Jan. 2018.

[31] R. Zhen, Y. Jin, Q. Hu, Z. Shao, and N. Nikitakos, "Maritime anomaly detection within coastal waters based on vessel trajectory clustering and Naïve Bayes classifier," *J. Navigat.*, vol. 70, no. 3, pp. 648–670, May 2017.

[32] R. Szlapczynski, "Evolutionary sets of safe ship trajectories: A new approach to collision avoidance," *J. Navigat.*, vol. 64, no. 1, pp. 169–181, Jan. 2011.

[33] T. Somers and G. A. Hollinger, "Human–robot planning and learning for marine data collection," *Auto. Robots*, vol. 40, no. 7, pp. 1123–1137, Oct. 2016.

[34] L. P. Perera, P. Oliveira, and C. Guedes Soares, "Maritime traffic monitoring based on vessel detection, tracking, state estimation, and trajectory prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1188–1200, Sep. 2012.

[35] H. Gao, W. Huang, and X. Yang, "Applying probabilistic model checking to path planning in an intelligent transportation system using mobility trajectories and their statistical data," *Intell. Autom. Soft Comput.*, vol. 25, no. 3, pp. 547–559, Sep. 2019.

[36] K. Patroumpas, E. Alevizos, A. Artikis, M. Vodas, N. Pelekis, and Y. Theodoridis, "Online event recognition from moving vessel trajectories," *GeoInformatica*, vol. 21, no. 2, pp. 389–427, Apr. 2017.

[37] F. W. Zhu, Y. J. Ren, J. Huang, J. Wan, and H. Zhang, "An edge computing-based framework for marine Fishery vessels monitoring systems," in *Proc. CollaborateCom*, London, U.K., 2019, pp. 201–214.

[38] Y. Yin, J. Xia, Y. Li, Y. Xu, W. Xu, and L. Yu, "Group-wise itinerary planning in temporary mobile social network," *IEEE Access*, vol. 7, pp. 83682–83693, 2019, doi: 10.1109/ACCESS.2019.2923459.

[39] G. Jia, G. Han, H. Rao, and L. Shu, "Edge computing-based intelligent manhole cover management system for smart cities," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1648–1656, Jun. 2018, doi: 10.1109/JIOT.2017.2786349.

[40] J. Muckell, J. Hwang, V. Patil, and C. T. Lawson, "SQUISH: An online approach for GPS trajectory compression," in *Proc. Geo*, Washington, DC, USA, May 2011, Art. no. 13.

[41] G. Trajcevski, H. Cao, P. Scheuermanny, O. Wolfsonz, and D. Vaccaro, "On-line data reduction and the quality of history in moving objects databases," in *Proc. 5th ACM Int. Workshop Data Eng. Wireless Mobile Access (MobiDE)*, Chicago, IL, USA, 2006, pp. 19–26.

[42] E. Keogh, S. Chu, D. Hart, and M. Pazzani, "An online algorithm for segmenting time series," in *Proc. IEEE Int. Conf. Data Mining*, San Jose, USA, 2001, pp. 289–296.

[43] M. Potamias, K. Patroumpas, and T. Sellis, "Sampling trajectory streams with spatiotemporal criteria," in *Proc. 18th Int. Conf. Sci. Stat. Database Manage. (SSDBM)*, Vienna, Austria, 2006, pp. 275–284.

[44] H. Gao, Y. Xu, Y. Yin, W. Zhang, R. Li, and X. Wang, "Context-aware QoS prediction with neural collaborative filtering for Internet-of-Things services," *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2019.2956827.

[45] R. Lange, T. Farrell, F. Durr, and K. Rothermel, "Remote real-time trajectory simplification," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun.*, Galveston, TX, USA, Mar. 2009, pp. 1–10.

JIE HUANG received the Ph.D. degree in geography information science from Zhejiang University, Hangzhou, China, in 2008.

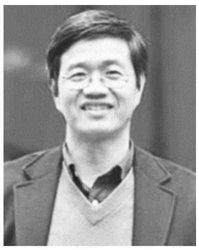
He is currently an Assistant Professor of software engineering with the School of Computer, Hangzhou Dianzi University, and the Zhejiang University of Science and Technology. He is the author of two books, more than 20 articles, and more than ten patents. His research interests include cloud computing, spatiotemporal analysis, and oceanographic.

JIAN WAN received the Ph.D. degree in computer application technology from Zhejiang University, Zhejiang, China, in 1989.

He is currently a Professor of software engineering with the School of Computer Science and Technology, Hangzhou Dianzi University, and the Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education. His research interests include grid computing, service computing, and cloud computing.

JIANJUN YU received the Ph.D. degree in environmental and water resources engineering from Nanyang Technological University, Singapore, in 2014, and the M.Sc. degree in geographic information system (GIS) from Zhejiang University, China, in 2005. He carried out his Postdoctoral research in water scarcity risk assessment at the University of Oxford, from 2015 to 2017. He was recently worked as a Smart Water Scientist at the Singapore Industry Solutions Research and Development Center, AVEVA Group Plc., focusing on automation of municipal water supply. His research focuses on water and environmental informatics, spatial-temporal data analytics, and GIS.

FENGWEI ZHU is currently pursuing the M.S. degree with the School Computer Science and Technology, Hangzhou Dianzi University, China. His research interests include data mining and edge computing.



YONGJIAN REN received the Ph.D. degree in engineering from Zhejiang University, Hangzhou, China, in 1989. He is currently a Distinguished Professor with Hangzhou Dianzi University. His research interests include mass storage and cloud computing.