Article

Deep Learning in Unmanned Surface Vehicles Collision-Avoidance Pattern Based on AIS Big Data with Double GRU-RNN

Jia-hui Shi and Zheng-jiang Liu *
Navigation College, Dalian Maritime University, Dalian 116026, China; shijiahui@dlmu.edu.cn
* Correspondence: liuzhengjiang@dlmu.edu.cn

Received: 4 August 2020; Accepted: 27 August 2020; Published: 4 September 2020

Abstract: There is a collection of a large amount of automatic identification system (AIS) data that contains ship encounter information, but mining the collision avoidance knowledge from AIS big data and carrying out effective machine learning is a difficult problem in current maritime field. Herein, first the Douglas–Peucker (DP) algorithm was used to preprocess the AIS data. Then, based on the ship domain the risk of collision was identified. Finally, a double-gated recurrent unit neural network (GRU-RNN) was constructed to learn unmanned surface vehicle (USV) collision avoidance decision from the extracted data of successful encounters of ships. The double GRU-RNN was trained on the 2015 Tianjin Port AIS dataset to realize the effective learning of ship encounter data. The results indicated that the double GRU-RNN could effectively learn the collision avoidance pattern hidden in AIS big data, and generate corresponding ship collision-avoidance decisions for different maritime navigation states. This study contributes significantly to the increased efficiency and safety of sea operations. The proposed method could be potentially applied to USV technology and intelligence collision avoidance.

Keywords: unmanned surface vehicles; ship collision avoidance; AIS; deep learning; double GRU-RNN

1. Introduction

With the development and the integration of the global economy, the marine world has become an important link for transportation and trade development in all countries in the world. However, the frequent occurrence of collisions at sea has caused not only great losses in terms of human lives and property but also has a wide coverage, posing a serious threat to marine ecology and the environment. These problems have gained increasing attention. In recent years, with the development of big data, artificial intelligence, cloud computing, and other technologies, the intelligence level of ships has continuously improved. Consequently, the development of unmanned ships has become an inevitable trend in future ship development. Due to their small size and intelligence, unmanned surface vehicles (USV) are often used to perform specific tasks, such as maritime rescue, marine surveying, and monitoring of dangerous goods. They have a wide range of applications worldwide. Although USV can serve to effectively reduce labor costs, the marine environment domain is complex, and therefore, to identify whether an USV can automatically avoid collisions is a prerequisite to successful completion of various tasks. As one of the core technologies used in USV, the automatic collision-avoidance technology includes not only a measure of the level of intelligence of unmanned ships but also the key to establish safe navigation of a USV in complex environments. Therefore, investigating the issue of collision-avoidance decision-making for unmanned ships has important practical significance to reduce or avoid maritime accidents.

In the maritime field, considering the large-scale popularization of AIS and the full coverage of shore-based equipment, the automatic identification system (AIS) big data is deemed as an effective
data source. In 2018, the fleet online HiFleet used its more than 3000 shore-based AIS base stations and 58 AIS satellites to receive an average of 150 million ship position report data from 838,044 ships (MSSI numbers between 200 million and 899,999,999) worldwide, every day Figure 1 shows the 2018 global liquefied natural gas (LNG) AIS trajectory data.

A large amount of AIS data contains extensive ship information, which requires certain screening and mining. Therefore, it is important to provide data support with the aim of establishing subsequent ship collision-avoidance, ship safety navigation, and ship behavior prediction. The maturity of algorithms, such as machine learning and big-data mining, has inspired new solutions to the previously mentioned problems in the marine field. Researchers received a new opportunity to easily access and analyze AIS data, to improve the intelligence of USV. Due to its easy access, multi-data integration, and other advantages, AIS has become a widely used big-data source in the field to perform prediction [1], clustering [2], reinforcement learning [3], pattern classification [4], etc.

Liu [5] focused on dynamic obstacle avoidance and considered the path-planning problem of a USV, using the ant colony algorithm (ACA) and the clustering algorithm (CA). Shi [6] introduced the concept of empirical mode decomposition (EMD) denoizing and Fermat’s spiral fitting, based on the AIS data. Lyu [7] combined the dynamic and static path planning approaches and proposed a set of real-time multi-ship adaptations for complex water automatic collision-avoidance decision-making and developing an effective path planning algorithm. Sun [8] suggested using 3D path planning for an autonomous underwater vehicle, (AUV) using a hierarchical deep Q-network combined with the prioritized experience replay. Pietrzykowski [9] realized a real-time implementation of an automotive collision mitigation system and evaluated the associated risk. Guo et al. [4] introduced deep reinforcement learning (DRL) to realize the intelligent route planning of unmanned ships in an unknown environment. Guardeño [10] focused on developing reactive static obstacle avoidance (SOA) methods, serving to increase the autonomy of USVs. Geng [11] developed velocity obstacle models, concerning both dynamic and static obstacles to represent a potential conflict-free region with regard to other objects. Xiu [12] implemented a maritime unmanned aerial vehicle (Mar-UAV) system, equipped with a high-resolution camera and an AIS. Azzeddine et al. [13] suggested considering an adaptive ship safety domain as a spatial risk function to identify both collision and grounding risks, based on motion and maneuverability conditions for all vessels.

Figure 1. 2018 global (liquefied natural gas) LNG automatic identification system (AIS) trajectory data.
The present study effectively combines the concepts of AIS and deep learning, to learn and solve the problem of USV’s maritime safety navigation and collision-avoidance. Figure 2 represents the overall framework of the proposed algorithm.

2. Materials and Methods

In this section, we first describe machine-learning training data preparation. In this study, we employed the Tianjin Port 2015 AIS trajectory dataset (Figure 3), as a source to specify the ship-encounter training data. In this study, extracting the successful cases of ship collision-avoidance from AIS big data could be divided into three steps—(a) ship AIS big data compression; (b) identification of the ship encounter situation data in historical data based on the ship domain; and (c) training data normalization. Data mining is used to extract training datasets that can be used to train a neural network. The preprocessing of training data could further improve the efficiency of the considered machine-learning technique, increase its pertinence, and solve the problem of blind learning.

Figure 2. Framework of the proposed algorithm.

Figure 3. Tianjin AIS data trajectories.
2.1. AIS Big Data Compression Based on the Douglas–Peucker Algorithm

As the high-frequency broadcast of AIS data accumulates a large number of data items in a period, there is a large amount of redundant information while defining the trajectory of a ship. The huge amount of AIS data might cause a system to perform a large number of irrelevant repetitive calculations and require considerable computational resources during data analysis. There are only a limited number of points in a ship trajectory that can actually characterize it, accounting for only 8% of the overall dataset. Therefore, in this study, we applied the Douglas–Peucker algorithm to compress and preprocess the AIS big data of a ship, to reduce the amount of the AIS big data, and to increase the value of the overall training data, which was beneficial for improving the efficiency of machine learning and did not require executing operations repeatedly.

While digitizing an AIS trajectory, it is necessary to sample a motion trajectory curve of a ship, i.e., to select a limited number of points on a curve and turn it into a polyline so that it can maintain the original shape to a certain extent. In this way, the training volume of a neural network could be reduced, and the learning efficiency might be improved.

The Douglas–Peucker algorithm [14] includes five steps, defined as follows (Figure 4):

1. Define a straight line (AB) between two points A and B at the beginning and end of the curve, corresponding to the chord of the curve.
2. Obtain (C) on the curve with the largest distance from the straight line segment and calculate the distance (d) from AB.
3. Compare the size of a distance using a predefined threshold. If it is less than the threshold, the straight line segment is regarded as an approximation of the curve, and the processing of the curve is completed.
4. If the distance is greater than the threshold, use C to divide the curve into two segments (AC and BC) and perform steps 1–3.
5. Upon completion of processing for all curves, a polyline formed by connecting each dividing point can be used as an approximation of a curve.

![Figure 4. Schematic of the proposed Douglas–Peucker algorithm.](image)

To facilitate the calculation and compression, we considered the recommended range of distance threshold selection for AIS data processing, under different scale charts (Table 1) [15].
Table 1. Recommended thresholds in different scales.

| Scale                      | Recommended Threshold/m | Compression Ratio/% |
|----------------------------|-------------------------|--------------------|
| <1:3,000,000               | 20                      | 97                 |
| 1:2,990,000~1:1,000,000    | 10~20                   | 94~97              |
| 1:990,000~1:2,000,000      | 4~10                    | 87~94              |
| 1:190,000~1:100,000        | 2~4                     | 78~87              |
| >1:100,000                 | 0.2~2                   | 36~78              |

Figure 5 displays the AIS trajectory and the compressed ship position, together, to show the compression effect of the DP algorithm. The DP algorithm shown in Figure 5 could reduce the amount of AIS data and enhance the processing efficiency of subsequent data, while retaining the characteristics of the ship’s trajectory.

Figure 5. Ship trajectory compression.

Figure 6 illustrates the comparison before and after the compression of the AIS data in the waters of Tianjin Port. The color bars of Figures 6 and 7, represent the density frequency values of the ship’s key feature points.

Figure 6. Before AIS data compression.
2.2. Identification of the Risk of Collision Based on the Ship Domain

After compressing AIS data, the ship encounter matching screening was performed, and the ship encounter data were further extracted. A successful case of ship collision-avoidance was considered as a training sample in the proposed double GRU-RNN. The conventional method requires the distance at Closest Point of Approach (DCPA) of the encounter between the two ships. Due to the difference in the positions of the other ships, a degree of danger, time to Closest Point of Approach (TCPA), was different. It was not sufficiently comprehensive to compare DCPA and TCPA alone, and therefore, in this study, we employed the shipping field as a method to identify ship encounters.

The ship field method was proposed by Fuji [16] et al. It implied defining an elliptical area in which the long axis was eight times of the length of a ship (a), and the short axis was 3.2 times of the length of the ship (b). This model could be described by an elliptical equation in a two-dimensional coordinated system. By calculating the value of lambda in a shipping area where the relative motion line of the other ship was tangent to the ship, whether the ship would encounter a collision risk, could be determined. Figure 8 represents a schematic diagram of the size change of the shipping area, with a step length of 0.1.

\[
\begin{align*}
    a' &= \lambda a \\
    b' &= \lambda b
\end{align*}
\]

where \(a'\) is the line of relative motion that is tangent to the long axis of the ellipse, \(b'\) is the line of relative motion that is tangent to the short axis of the ellipse.

In this way, the risk of collision could be successfully quantified as a single value (\(\lambda\)), which was convenient for the calculation and screening of ship encounter data. Usually, two-dimensional coordinates can be used to investigate the movements of ships in the sea. In the surface coordinate system, the origin of a coordinate is a fixed point on a surface; the y-axis is north-oriented, and the x-axis is east-oriented. In the ship motion coordinate system, the center of a ship is the coordinate origin. The ship length direction is the y-axis, and the ship width direction is the x-axis. The relationship between these two systems is represented in Figure 9.
In this way, the risk of collision could be successfully quantified as a single value ($\lambda$), which was convenient for the calculation and screening of ship encounter data. Usually, two-dimensional coordinates can be used to investigate the movements of ships in the sea. In the surface coordinate system, the origin of a coordinate is a fixed point on a surface; the y-axis is north-oriented, and the x-axis is east-oriented. In the ship motion coordinate system, the center of a ship is the coordinate origin. The ship length direction is the y-axis, and the ship width direction is the x-axis. The relationship between these two systems is represented in Figure 9.

**Figure 8.** Extent of the proposed approach based on the ship domain.

**Figure 9.** Relationship between own ship motion system and the target ship motion system in the earth coordinate system.
Transformation from the Earth's surface coordinate system \( XOY \) to the ship motion coordinate system \( xoy \) is defined as follows:

\[
\begin{align*}
x'_t &= x_t \cos \theta_o + y_t \sin \theta_o - x_o \cos \theta_o - y_o \sin \theta_o \\
y'_t &= -x_t \sin \theta_o + y_t \cos \theta_o + x_o \sin \theta_o - y_o \cos \theta_o
\end{align*}
\]

(2)

where \( a \) is the semi-axial length and \( b \) is the semi-axial width of the own ship domain, \( V_o \) is the speed of the own ship, \( V_t \) is the speed of the target ship, \( \theta_o \) is the course of the own ship, and \( \theta_t \) is the course of the target ship, \((x_o, y_o)\) is the own ship’s coordinates in Earth’s surface coordinate system, \((x_t, y_t)\) is the target ship coordinates in Earth’s surface coordinate system, \((x'_t, y'_t)\) is the target ship coordinates in the own ship motion coordinate system.

\[
\gamma = \alpha - \mathcal{C}
\]

(3)

\[
\begin{align*}
xx &= s \times \sin(c) - S \times \sin(C) \\
yy &= s \times \cos(c) - S \times \cos(C)
\end{align*}
\]

(4)

\[
\begin{align*}
x'_t &= d \times \sin(\gamma) \\
y'_t &= d \times \cos(\gamma)
\end{align*}
\]

(5)

\[
\begin{align*}
\left\{ \begin{array}{l}
y = (y'_t - x'_t \times (yy/xx))/((1 + 0.16 \times (yy/xx)^2)) \\
x = -((yy/xx) \times 0.16 \times y)
\end{array} \right.
, xx \neq 0 \text{ and } yy \neq 0 \\
y = 0 \\
x = x'_t \\
y = 0 \\
x = y'_t \\
y = x'_t \\
x = y'_t
\end{align*}
\]

(6)

\[
\lambda = \sqrt{(y/(4 \times L))^2 + (x/(1.6 \times L))^2}
\]

(7)

where \( S \) is the speed of the own ship; \( C \) is the course of the own ship; \( L \) is the captain of the own ship; \( s \) is the target speed; \( c \) is the target course; \( \alpha \) is the target azimuth angle; \( d \) is the distance; \( \gamma \) is the side angle; \( xx, yy \) are the component vectors of the relative speed on the x-axis and y-axis; \((x, y)\) is the tangent point coordinate of the ship domain.

The value of \( \lambda \) reflects how close the target ship is to the own ship (Figure 10). If \( \lambda > 1 \), the target ship did not enter its own spacecraft domain, and there was no risk of collision; however, if \( \lambda \leq 1 \), and no evasive action was taken, the target ship entered its own ship domain, and there was a risk of collision. Ships must take corresponding actions in accordance with COLREG requirements to avoid collisions.

To facilitate the calculation and comprehensiveness of collision avoidance, we visualized the collision risk of ships based on the shipping field. Considering a 360° incoming ship, two ships encountered a full-scale heading difference \((-180°-180°)\) with 0.1° being the step size. We generated a 3D view of the collision risk of a ship, as represented in Figure 11. The collision risk diagram could be used to intuitively judge the encounter state and the degree of danger of a ship.
To facilitate the calculation and comprehensiveness of collision avoidance, we visualized the collision risk of ships based on the shipping field. Considering a 360° incoming ship, two ships encountered a full-scale heading difference (−180~180°) with 0.1° being the step size. We generated a 3D view of the collision risk of a ship, as represented in Figure 11. The collision risk diagram could be used to intuitively judge the encounter state and the degree of danger of a ship.

It can be seen from Figure 11 that X, Y is the relative position coordinate of the target ship, and the Z-axis represents the heading difference between the target ship and own ship. The color (λ) could be used to indicate the current tangency between the relative motion line of the other ship and the ship domain of the own ship. The darker the color, the higher the risk, and the smaller the lambda value.
The ship encounter data were extracted from the massive ship AIS trajectories, as shown in Figure 12.

![Figure 12. Ship encounter AIS trajectory data.](image)

2.3. Training Data Normalization

To facilitate machine learning, it was necessary to process the extracted AIS encounter data and apply standardized conversion. We set the first position of a ship as the origin of coordinates and set the course of the first position of a ship as 000. There were two steps to transform a location point, a relative position, and an angle in each batch of data. (a) Translation—the first position of the own ship in the training group was employed as the reference to perform coordinate translation. (b) Rotation transformation—the heading of the first position of the own ship was utilized as the reference to perform coordinate rotation transformation. Translation and rotation transformation are illustrated in Figure 13.

\[
\begin{aligned}
\begin{cases}
x_0 = Ox_0 \\
y_0 = Oy_0 \\
h_0 = Oh_0
\end{cases}
\end{aligned}
\]

\[
\begin{aligned}
\begin{cases}
Ox_i = Ox_i - x_0 \\
Oy_i = Oy_i - y_0 \\
Oh_i = Oh_i - h_0 \\
Tx_i = Tx_i - x_0 \\
Ty_i = Ty_i - y_0 \\
Th_i = Th_i - h_0
\end{cases}
\end{aligned}
\]

\[
\begin{aligned}
\begin{cases}
x' = \cos(h_0) \times x - \sin(h_0) \times y \\
y' = \cos(h_0) \times y + \sin(h_0) \times x
\end{cases}
\end{aligned}
\]
2.4. Double GRU-RNN Model

In this section, we first present the structure of the proposed double-GRU-RNN deep learning model.

The RNN network introduces a directional loop, which can deal with the problems of correlation between inputs. Breaking the traditional neural network structure layer and the full connection between the layers, the transition state between each layer of nodes was no longer the traditional neural network model of input-hidden layer-output. AIS ship trajectory is a typical time-series data, and the RNN network has a high degree of fit for AIS data. In theory, RNN can process time-series data of any length.

There is no guarantee of a one-to-one correspondence between the other ships’ actions and the own ship. Under normal conditions, the trajectory feature points of a direction-maintaining ship is significantly smaller than those of a give-way ship. The output, the length of the input sequence, and the length of the output sequence cannot be predicted in advance, and therefore, we propose to develop two GRU-RNNs to learn collision-avoidance decisions of a USV. GRU-RNN_1 is employed as a neural network to process the navigation state of the target ship; GRU-RNN_2, as a prediction network for collision-avoidance decision making of a ship, can be used to effectively solve the problem of the unequal database length, and at the same time, adding a neural source into a GRU to solve the gradient mass problem caused by the long length of data. The structure of the proposed double GRU-RNN is illustrated in Figure 14.
Figure 14. Conversion of the proposed double double-gated recurrent unit neural network (GRU-RNN).

The activation function of RNN also has a soft saturation using the tanh function. As its output is centered at 0, the convergence speed is faster than the sigmoid function. The function image is shown in Figure 15.

\[
\tanh(x) = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}}
\]  

(11)

Figure 15. Tanh activation function image.

The input layer of ship behavior input data \(I(t)\) can be expressed as:

\[
I(t) = \{\text{lon}_1, \text{lat}_1, t_1, \text{lon}_2, \text{lat}_2, t_2, \ldots, \text{lon}_i, \text{lat}_i, t_i\}
\]  

(12)
ship position data of the output $O(t)$ layer is expressed as:

$$O(t + 2) = \{\text{lon}_1', \text{lat}_1', t_1', \text{lon}_2', \text{lat}_2', t_2', \ldots, \text{lon}_j', \text{lat}_j', t_j'\}$$  \hspace{1cm} (13)

For the loss function, we used the tensorflow function (tf.contrib.seq2seq.sequence_loss). The loss function applied in actual use was a softmax loss function.

$$L(W) = \frac{1}{N} \sum_{i=1}^{N} L_i(f(x_i, W), y_i) + \lambda R(W)$$  \hspace{1cm} (14)

### 2.4.1. GRU Cell Structure

A GRU neural network [17] is a variant of a long short-term memory (LSTM). Both are defined based on an RNN neural network. However, a GRU neural network is an improvement of an RNN neural network that has a short-term memory function. During back propagation, there is a problem with the gradient disappearance. Therefore, the RNN neural network is not friendly in terms of updating layers with small gradients. The GRU neural network sets up two gates (update gate and reset gate). The GRU neural network has fewer tensors and fewer parameters than the LSTM neural network and is much faster than the LSTM neural network in speed. The GRU neural network replaces the hidden unit of an RNN with a GRU module unit. The GRU neural network algorithm is an algorithm proposed by Cho et al. in 2014 [18]. The GRU unit structure is represented in Figure 16.

![Figure 16. GRU unit structure.](image_url)

In the GRU neural network algorithm, two gates were set, namely, the update gate and the reset gate. The reset gate was aimed to control the degree of ignorance of information in the previous stage; the function of the update gate was used to update the unit state in time and pass the hidden state and input state to the sigmoid function; the sigmoid function belonged to 0 or 1, and when the value was 0, it was discarded. When the value was 1, it was reserved. Using the GRU neural network algorithm, we could mitigate the problem of gradient disappearance [19]. The workflow of a GRU neural network was defined as follows:
Step 1: Solve the reset gate \((r_t)\) and the candidate state \((\tilde{h}_t)\):

\[
r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)
\]

\[
\tilde{h}_t = \tanh(W_r x_t + U_r (r_t \odot h_{t-1}) + b_r)
\]

Step 2: Solve the update gate \((z_t)\) and the current state \((h_t)\):

\[
z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)
\]

\[
h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t
\]

It could be seen that there were linear and nonlinear relationships between the current state \((h_t)\) and the historical state \((h_{t-1})\). When the update gate \((z_t = 0)\), and the reset gate \((r_t = 1)\), the current state \((\tilde{h}_t)\) was defined as below:

\[
\tilde{h}_t = \tanh(W_h x_t + U_h h_{t-1} + b_h)
\]

2.4.2. Double RNN Structure

(1) Target Ship State Handling

The variable-length trajectory feature points of other ships were directly input into GRU-RNN_1 for processing, and the semantic vector value \(C\) was obtained after completing the centralized processing as the input value of GRU-RNN_2 as follows:

\[
C = q(h_1, h_2, h_3, \ldots, h_t)
\]

where \(t\) represents the number of trajectory feature points in an input sequence. Then, we considered the output \(h_{t+1}\) of the last hidden layer as the semantic vector \(C\).

(2) Own Ship Collision-Avoidance Decision

Using \(C\) as the input value in GRU-RNN_2, we input the trajectory feature points of the collision-avoidance action undertaken by a ship in a predictive manner and used the input trajectory feature points as input. Finally, we aimed to obtain the complete avoidance trajectory feature point sequence of the ship \(Y_2, Y_{1-1}\):

\[
Y_t = \arg\max P(Y_t) = \prod_{i=1}^{T} p(Y_t|Y_1, Y_2, Y_3, \ldots, Y_{t-1})
\]

As a result of the continuous evolution of RNN, the structure of neural networks becomes more difficult to fit complex data structures, simplifying input data preprocessing and making the input information more complete and without loss. We utilized the overall target ship trajectory key point sequence as an input and the overall own ship trajectory key point sequence as an output. GRU-RNN_1 handled the target ship sequence to obtain semantic vector \(C\). GRU-RNN_2 was utilized to derive the corresponding avoidance decision sequence (Figure 17).

Concerning the two GRU-RNN networks, the one mainly completed the input of a time-series data sequence to generate a state vector, and the other one generated a time-series data sequence through the generated state vector. The two networks cooperated with each other to achieve better performance in learning ship collision-avoidance decision making.
3. Results

The AIS data used in this article was the full-year AIS data of the Tianjin Port in 2015, including the 22,349 ships occupying 8.3 GB of storage space, with the 1,511,504,900 trajectory points. The AIS data structure storage is shown in Table 2.

| MMSI     | Heading/° | Course/° | Speed/Kn | Time          | Lon/° | Lat/°     |
|----------|------------|----------|----------|---------------|-------|-----------|
| 636015XXX| 101        | 101.8    | 11.3     | 1440538896    | 117.911| 38.95156  |
| 636015XXX| 100        | 102.4    | 11.3     | 1440538986    | 117.9169| 38.95053  |
| 636015XXX| 100        | 101.5    | 11.3     | 1440539076    | 117.9229| 38.94955  |
| 636015XXX| 99         | 101.3    | 11.4     | 1440539154    | 117.9289| 38.94861  |
| 636015XXX| 100        | 101.3    | 11.3     | 1440539257    | 117.9348| 38.94771  |
| 636015XXX| 99         | 100.9    | 11.3     | 1440539346    | 117.9408| 38.94683  |
| 636015XXX| 99         | 101.2    | 11.3     | 1440539466    | 117.9487| 38.94538  |
| 636015XXX| 99         | 100      | 11.3     | 1440539598    | 117.9574| 38.94425  |
| 636015XXX| 99         | 100.4    | 11.3     | 1440539706    | 117.9646| 38.94327  |
| 636015XXX| 99         | 99.8     | 11.2     | 1440539826    | 117.9725| 38.94218  |

This article used Win10 x64 operating system with Intel(R) Core i7-9700F CPU, RAM 16 GB. The Double GRU-RNN was implemented in Python in a machine-learning framework provided by Google TensorFlow. The overall structure of the network training automatically generated using TensorBoard is depicted in Figure 18.

The double GRU-RNN had two input values—(a) target ship trajectory, and (b) own ship decision, where the generated decision and own ship decision were the optimized input of backpropagation training. Figure 19 illustrates the ship encounter data filtered by the AIS big data, which were used as the training data for the double GRU-RNN.
The double GRU-RNN had two input values—(a) target ship trajectory, and (b) own ship decision, where the generated decision and own ship decision were the optimized input of backpropagation training. Figure 19 illustrates the ship encounter data filtered by the AIS big data, which were used as the training data for the double GRU-RNN.

The trained double GRU-RNN network could process the trajectory of target ships and specify corresponding collision-avoidance operations in the own ship. The collision-avoidance results are represented in Figures 20 and 21.

---

**Figure 18.** Double GRU-RNN TensorFlow framework.

**Figure 19.** Double GRU-RNN training data preparation.
The trained double GRU-RNN network could process the trajectory of target ships and specify corresponding collision-avoidance operations in the own ship. The collision-avoidance results are represented in Figures 20 and 21.

Figure 20. Schematic diagram of the unmanned surface vehicle (USV) collision-avoidance process (1).

Figure 21. Schematic diagram of the USV collision-avoidance process (2).
4. Discussion

This research was aimed to provide a theoretical basis for designing innovative intelligent collision-avoidance systems for USV. In this paper, we proposed the double GRU-RNN to learn the effective ways of making appropriate collision-avoidance decisions, based on successful collision-avoidance case data. Using two GRU-RNN networks, we successfully processed own ship trajectory data and target ship trajectory data. The proposed method could freely process the trajectory data of different lengths, was deemed more flexible, and was capable of deriving more accurate ship collision-avoidance decisions. The proposed algorithm included big-data training and had a broad range of applications and high versatility to achieve timely strategy generation, timely response, and enhanced collision-avoidance security.

Author Contributions: Conceptualization, J.-h.S.; Software, J.-h.S.; Supervision, Z.-j.L.; Writing—original draft, J.-h.S.; Writing—review & editing, J.-h.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China, grant number (51579025, 51709165) and “The Fundamental Research Funds for the Central Universities” (3132020134, 3132020139).

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

References
1. Gao, M.; Shi, G.; Li, S. Online Prediction of Ship Behavior with Automatic Identification System Data Using Bidirectional Long Short-Term Memory Recurrent Neural Network. Sensors 2018, 18, 4211. [CrossRef] [PubMed]
2. Wang, J.; Zhu, C.; Zhou, Y.; Zhang, W. Vessel Spatio-temporal Knowledge Discovery with AIS Trajectories Using Co-clustering. J. Navig. 2017, 70, 1383–1400. [CrossRef]
3. Guo, S.S.; Zhang, X.X.; Zheng, Y.Y.; Du, Y. An Autonomous Path Planning Model for Unmanned Ships Based on Deep Reinforcement Learning. Sensors 2020, 20, 426. [CrossRef] [PubMed]
4. Pedroche, D.S.; Amigo, D.; García, J.; Molina, J.M. Architecture for Trajectory-Based Fishing Ship Classification with AIS Data. Sensors 2020, 20, 3782. [CrossRef]
5. Liu, X.X.; Li, Y.Y.; Zhang, J.J.; Zheng, J.J.; Yang, C. Self-Adaptive Dynamic Obstacle Avoidance and Path Planning for USV Under Complex Maritime Environment. IEEE Access 2019, 7, 114945–114954. [CrossRef]
6. Shi, B.B.; Su, Y.Y.; Zhang, D.; Wang, C.; AbouOmar, M.S. Research on Trajectory Reconstruction Method Using Automatic Identification System Data for Unmanned Surface Vessel. IEEE Access 2019, 7, 170374–170384. [CrossRef]
7. Lyu, H.; Yin, Y. COLREGS-Constrained Real-time Path Planning for Autonomous Ships Using Modified Artificial Potential Fields. J. Navig. 2019, 72, 588–608. [CrossRef]
8. Sun, Y.; Ran, X.; Zhang, G.; Xu, H.; Wang, X.-B. AUV 3D Path Planning Based on the Improved Hierarchical Deep Q Network. J. Mar. Sci. Eng. 2020, 8, 145. [CrossRef]
9. Pietrzykowski, Z.; Wolejsza, P.; Borkowski, P. Decision Support in Collision Situations at Sea. J. Navig. 2016, 70, 447–464. [CrossRef]
10. Guardeño, R.; López, M.J.; Sánchez, J.; Consegliere, A. AutoTuning Environment for Static Obstacle Avoidance Methods Applied to USVs. J. Mar. Sci. Eng. 2020, 8, 300. [CrossRef]
11. Geng, X.; Wang, Y.; Wang, P.; Zhang, B. Motion Plan of Maritime Autonomous Surface Ships by Dynamic Programming for Collision Avoidance and Speed Optimization. Sensors 2019, 19, 434. [CrossRef] [PubMed]
12. Xiu, S.; Wen, Y.; Yuan, H.; Xiao, C.; Zhan, W.; Zou, X.; Zhou, C.; Shah, S.C. A Multi-Feature and Multi-Level Matching Algorithm Using Aerial Image and AIS for Vessel Identification. Sensors 2019, 19, 1317. [CrossRef] [PubMed]
13. Bakdi, A.; Glad, I.; Vanem, E.; Engelhardtson, Ø. AIS-Based Multiple Vessel Collision and Grounding Risk Identification based on Adaptive Safety Domain. J. Mar. Sci. Eng. 2019, 8, 5. [CrossRef]
14. Mou, J.M.; Chen, P.F.; He, Y.X.; Zhang, X.J.; Zhu, J.F.; Rong, H. Fast self-tuning spectral clustering algorithm for AIS ship trajectory. J. Harbin Eng. Univ. 2018, 39, 428–432.
15. Wen, X.; Jiangqiang, H.; Jianchuan, Y.; Ke, L. Ship Automatic Collision Avoidance by Altering Course Based on Ship Dynamic Domain. In Proceedings of the 2016 IEEE Trustcom/BigDataSE/I SPA, Tianjin, China, 23–26 August 2016.

16. Li, W.; Yang, J.; Gao, X.; Yu, J. Modelling and simulation of intelligent collision avoidance based on ship domain. *Int. J. Simul. Process Model.* **2020**, *15*, 64–75. [CrossRef]

17. Wang, J.; Yang, Y.; Mao, J.; Huang, Z.; Huang, C.; Xu, W. CNN-RNN: A Unified Framework for Multi-label Image Classification. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.

18. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv* **2014**, *arXiv:1406.1078*.

19. Che, Z.; Purushotham, S.; Cho, K.; Sontag, D.; Liu, Y. Recurrent Neural Networks for Multivariate Time Series with Missing Values. *Sci. Rep.* **2018**, *8*, 53–62. [CrossRef] [PubMed]

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).