Article

Conflict Risk Assessment between Non-Cooperative Drones and Manned Aircraft in Airport Terminal Areas

Renwei Zhu [1,2], Zhao Yang [3,*] and Jun Chen [1]

1   School of Transportation, Southeast University, Nanjing 211189, China
2   CAUPD Beijing Planning & Design Consultants LTD, Nanjing 100044, China
3   College of General Aviation and Flight, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
*   Correspondence: yangzhao@nuaa.edu.cn

Abstract: Recent years have seen an increase in events of drone incursion into airport terminal areas, leading to safety concerns and disruptions to airline operations. It is of great importance to identify the potential conflict, especially for those non-cooperative drones, as their intentions are always unknown. For the safe operation of air traffic, this paper proposes a conflict risk assessment method between non-cooperative drones and manned aircraft in the terminal area. First, the trajectory data of manned aircraft and drones are obtained. Real-time cylindrical protection zones are established around manned aircraft according to the separation interval for safe operation between the drone and the manned aircraft at different altitudes. Secondly, trajectory predictions for the manned aircraft and the drone are conducted, respectively. A quartile regression bidirectional gate recurrent unit neural network is proposed in this research for the trajectory prediction of the drones. The model integrates the bidirectional gated recurrent unit structure and the quartile regression structure. The performance indicators confirm the superiority of the proposed model. Based on the trajectory prediction results, it is then determined whether there is a conflict risk between the drone and manned aircraft by comparing the position distribution of the drone as well as the real-time cylindrical protection zone of the manned aircraft. The conflict probability between the drone and the manned aircraft is then calculated. The prediction accuracy of conflict probability is estimated by Monte Carlo simulation methods. The collision probability prediction accuracy of manned aircraft and drones at different flight stages and altitudes ranges from 73% to 97%, which shows the reliability of the proposed method. Finally, the collision probability between the drone and the manned aircraft at the closest encountering point and the estimated time to reach the closest encountering point are calculated. This paper predicts the conflict risk between the drone and manned aircraft, thus providing theoretical support for the safe operation of air transport in low-altitude environments.

Keywords: non-cooperative drones; manned aircraft; trajectory prediction; machine learning; artificial neural networks; Monte Carlo simulation

1. Introduction

In recent years, incidents of drones invading the airspace around airports and interfering with civil manned aircraft have occurred frequently, resulting in the inbound flights of airports being diverted to other places and large-scale delays of outbound flights. Since most of the drones intruding at airports are non-cooperative drones, the flight information between the drone and the manned aircraft cannot be exchanged in real-time. When a drone is detected, it is difficult to determine the position of the drone at the next moment accurately and efficiently, as their intentions are always unknown. The conflict risk between the drone and the manned aircraft is uncertain and the risk severity is unclear. The risk assessment of conflict between drones and manned aircraft is an important means of airport safety management. The collision probability prediction and risk assessment between drones and manned aircraft in low-altitude environments can provide technical support for the early warning of conflict in the context of air traffic.

Until recently, a series of studies have been conducted concerning aircraft trajectory prediction. During the take-off, landing, and cruise phases, the manned aircraft basically follows the flight path, as they take off and land according to arrival and departure procedures. However, there is often a large temporal or spatial deviation for the specific track points as compared with the scheduled path, especially in complex air traffic environments [1]. In order to improve the safety of air traffic, it is necessary to predict the future trajectory points according to the historical trajectory of manned aircraft and drones, so as to assist the collision probability analysis between them.

The trajectory prediction models of traditional civil aircraft can be divided into prediction based on state estimation, prediction based on dynamics, and prediction based on machine learning [2]. Common predictions based on state estimation include the Kalman filter model [3], the particle filter algorithm [4], the hidden Markov model [5], etc. For example, the hidden Markov model is applied together with a Gaussian mixture model [6] or trajectory similarity [7] to represent the position and altitude transition patterns of the aircraft during flight operation. Since these models are only based on the state estimation of kinematic equations, they are only suitable for short-term prediction, and the prediction error is often large. Prediction based on dynamics relies on a large number of performance parameters under some ideal assumptions [8]. Sun et al. presented a set of methods for extracting different aircraft performance parameters for common aircraft types. The parametric models combined can be used to describe a complete flight that includes takeoff, initial climb, climb, cruise, descent, final approach, and landing [9]. However, the parameters vary significantly across different types of aircraft, i.e., the civil aircraft [10] and the general aviation aircraft [11]. The high dimensionality of the problem and nonlinearities in aircraft dynamics and control limits the use of common dynamic methods to obtain accurate prediction results.

On the other hand, prediction based on machine learning can mine information in historical trajectory data for effective prediction, which is often reported to produce satisfactory results. Machine learning technology has been widely used in autonomous driving, image recognition, etc., in which deep neural networks show good application ability. Moreover, research on deep neural network classifiers [12] and the detection of backdoor attacks [13] was reported to further improve recognition performance. In terms of the trajectory prediction of aircraft, the trajectory predictors could be enhanced by learning from historical data [14]. For example, Shi et al. proposed a constrained long short-term memory network for flight trajectory prediction. It is observed that the method outperforms the widely used long short-term memory network, Markov, weighted Markov, support-vector machine, and Kalman filter models [15]. Wu et al. proposed a 4D trajectory prediction model based on the backpropagation (BP) neural network. The results indicated that the predicted 4D trajectory is close to the real flight data, the time error at the crossing point is no more than 1 min, and the altitude error at the crossing point is no more than 50 m [16]. Gallego et al. developed a probabilistic horizontal interdependency measure between aircraft supported by machine learning algorithms, addressing time separations at crossing points [17].

In contrast to traditional civil aircraft or fixed-wing unmanned aerial vehicles (UAV), rotary-wing drones have high stability during vertical take-off and landing, hovering, and horizontal movement [18]. Due to the characteristics of strong maneuverability, high flexibility, and the changeable range of activities of rotary-wing drones, their intentions are not easy to capture, and their trajectories are more difficult to predict [19]. It is recognized that non-cooperative drones pose several safety and privacy concerns to the public [20]. Research has been conducted to identify target intent [21], some of which was conducted under the conditions of uncertain or incomplete information [22]. Existing trajectory prediction models for drones include models based on equations of kinematics and dynamics, such as the model predictive control model subject to aerodynamic disturbances [23], the motion-based trajectory paradigm shift that improves Flight Management System compatibility with tactical operations [8], the algorithm for the estimation, filtering, and prediction

of the trajectories of light aircraft and gliders using interacting multiple model filters [24], and machine learning methods based on historic trajectory data, such as recurrent neural networks (RNNs), long short-term memory (LSTM), gated recurrent unit (GRU), and their various improved algorithms [25]. In the authors' previous research, a novel procedure based on the bidirectional gated recurrent unit (D-GRU) method has been proposed for intention recognition and the short-term trajectory prediction of quadrotors [26].

When a non-cooperative drone invades the prohibited zone of an airport or the protected area of a manned aircraft, it may collide with the manned aircraft that is taking off or landing near the airport [27]. To ensure the safe, orderly, and efficient operation of an aircraft in low-altitude airspace, it is necessary to obtain the accurate trajectory prediction results of manned aircraft and drones in order to provide technical support for abnormal behavior detection, conflict warning, and flight situation awareness.

At present, the research on collision risk mainly focuses on the collision risk between manned aircraft or between drones, whereas there are a few works concentrated on the collision probability of intruding drones to manned aircraft. For example, a probabilistic model based on the stochastic kinematic model was developed to implement the collision risk evaluation. The cases covering different drones' initial positions, positions updates, and different collision zones are simulated and analyzed using the proposed collision-course based model [28]. A 3D Monte Carlo unmanned aircraft system (UAS) positional distribution model, based on flight dynamics of the UAS, was developed to help assess the risk posed by the UAS to aircraft operating inside the aerodrome. The 3D model was also used to carry out simulations that could help determine the buffer airspace needed for cooperative UASs operating inside the aerodrome [29].

Although few studies have been conducted with regard to the collision risk between manned aircraft and drones, most were based on kinematic models and simulation analysis under hypothetical assumptions. Due to the limitations of data acquisition, the conflict risk between drones and manned aircraft has seldom been estimated and the risk level is unclear. Furthermore, the reliability of the Monte Carlo simulation results needs to be further verified due to limited iterations. The purpose of this paper is to provide a conflict risk assessment method between manned aircraft and drones in the terminal area based on actual trajectory data of manned aircraft and drones. This method is oriented to the different flight stages and flight altitudes of manned aircraft. Based on the trajectory prediction results of manned aircraft and the drone, the collision probability is calculated.

The main contributions and innovations provided in this paper are as follows:

First, due to the flexibility and changeable range of activities of drones, their position at the next moment is uncertain. Therefore, the use of deterministic trajectory prediction may bring a large error to the collision prediction results. This paper proposes a quartile regression bidirectional gate recurrent unit neural network (QRDGRU) to predict the location of the drone. The model integrates the bidirectional gated recurrent unit structure and the quartile regression structure, which can obtain the position distribution of the drone. Considering the uncertainty of drone position, the collision probability prediction can be more accurate.

Second, most of the existing prediction methods regarding collision probability are based on the assumption that the error follows a normal distribution. The collision probability is calculated considering the error caused by the uncertainty of the predicted position. In this paper, a Monte-Carlo-based collision probability prediction method between manned aircraft and drones is proposed. Time to the closest distance at the encountering point and the collision probability at the closest point are calculated.

The rest of the paper is organized as follows. The data collection and preparation process are presented in Section 2. Section 3 exhibits the prediction method of manned aircraft trajectory in terminal areas. Section 4 provides the prediction of trajectory distribution for drones. The conflict risk assessment between drones and manned aircraft is presented in Section 5. Section 6 summarizes the conclusions and proposes future works.

2. Data Preparation

To meet the research objective, the trajectory data of both manned aircraft and drones were collected. The data collection and processing process is illustrated as follows.

2.1. Data Collection

For the manned aircraft, the Automatic Dependent Surveillance–Broadcast (ADS-B) trajectory data from Shanghai Hongqiao Airport (ZSSS) to Beijing Capital International Airport (ZBAA) were obtained, with flight data for a total of 9752 flights. The ADS-B data of each flight include the flight ID, aircraft type, longitude, latitude, altitude, ground speed, heading angle, monitor time, departure airport, landing airport, etc.

The main flight stages of manned aircraft can be divided into the take-off stage, cruise stage, and landing stage. Since drones can only collide with manned aircraft in the take-off and landing stages, the trajectory data of the manned aircraft in the two phases around the airport terminal area are selected for further analysis, which usually has an altitude of below 1000 m. Table 1 is an example of ADS-B trajectory data for manned aircraft, including aircraft type, longitude, latitude, altitude, ground speed, and monitoring time, as follows:

Table 1. Trajectory data illustration for manned aircraft.

Longitude (°)	Latitude (°)	Altitude (m)	Ground Speed (m/s)	Monitoring Time
121.3286514	31.18910980	7.62	209.21	18 November 2017 14:35:13
121.3322525	31.20203972	114.30	209.21	18 November 2017 14:35:19
121.3318024	31.20713043	167.64	209.21	18 November 2017 14:35:23
121.3292694	31.21495056	419.10	217.26	18 November 2017 14:35:38
121.3263092	31.21867943	495.30	217.26	18 November 2017 14:35:43
121.3179321	31.22800064	723.90	209.21	18 November 2017 14:36:04
121.3126831	31.23370934	853.44	205.00	18 November 2017 14:36:14
121.3089905	31.23778915	937.26	207.61	18 November 2017 14:36:24

Because the ADS-B data transmit various information to ground equipment through data link broadcasting, and are affected by various factors such as signals, the collected data might be inaccurate or missing. It is necessary to preprocess the data to improve the quality of the original data and delete the flight data with crucial missing or inaccurate information.

Toward the research on the trajectory distribution prediction of drones, this paper collected a total of 34 h of historical trajectory data of three types of quadrotor drones over 30 days. Each drone automatically records relevant information every 0.1 s. Table 2 is an example of part of the trajectory data of the drone, including timestamp, longitude, latitude, altitude, x-speed, y-speed, z-speed, heading angle, pitch angle, roll angle, etc. It can be seen that the trajectory is flexible and changeable. Table 3 is a summary of the collected data for drones, including the number of trajectory records, as well as the statistical information about the ground speed and vertical speed. The ground speed for the drone ranges from 0 to 20.60 m/s and the vertical speed ranges from −9.86 to 11.60 m/s.

2.2. Data Processing

The obtained trajectory data for manned aircraft and the drone are processed and represented as time-series data at equal intervals. The detailed procedure is illustrated as follows.

(1) Data processing for manned aircraft

Step 1-1: Coordinate system transformation
The longitude and latitude in the collected aircraft trajectory data based on the space spherical coordinate system are converted into the Earth-centered, Earth-fixed coordinate system (ECEF), which is shown in Equation (1).

$$\begin{cases} X = (h + N) * \cos \varphi * \cos \lambda \\ Y = (h + N) * \cos \varphi * \sin \lambda \end{cases} \tag{1}$$

In the formula, h is the height; N is the radius of curvature; φ is the radian corresponding to latitude; λ is the radian corresponding to longitude. The unit of latitude and longitude after coordinate transformation is meter (m). X is the longitude after transformation; Y is the latitude after transformation.

Step 1-2: Time series data generation

For the altitude and ground speed in the aircraft trajectory data and the converted longitude X and latitude Y in Step 1-1, time series data at equal intervals are generated through linear interpolation.

Step 1-3: Dimensional processing and sample segmentation

The equally spaced time series data in Step 1-2 are normalized to eliminate the influence of dimension, and the normalized equally spaced time series data are divided into fixed-length time series slices via a sliding time window. The coordinate value of the trajectory point contained in the slice sample is subtracted from the initial position value of the sample, and the slice sample set containing the input and output data of the trajectory prediction model in the take-off stage and the landing stage are generated, respectively. Each slice sample set includes four variables: latitude, longitude, altitude, and ground speed.

Table 2. Trajectory data illustration for drones.

Timestamp (ms)	Longitude (°)	Latitude (°)	Altitude (m)	x-Speed (m/s)	y-Speed (m/s)	z-Speed (m/s)	Heading Angle (°)	Pitch Angle (°)	Roll Angle (°)
6600	107.828147	35.7985822	4.4	−0.447	−0.224	−4.921	144.5	0.6	0.1
6700	107.828147	35.79858204	4.6	−0.447	−0.224	−5.145	144.4	1.7	−0.6
6800	107.8281451	35.79858007	4.8	−0.447	−0.224	−5.369	144.1	2.6	0.4
6900	107.828145	35.79857969	5.1	−0.671	−0.224	−5.592	144.2	2.7	0.2
7000	107.828144	35.79857816	5.3	−0.671	−0.224	−5.816	144.2	4.2	−0.6
7100	107.8281438	35.79857759	5.6	−0.671	−0.224	−5.816	144.0	5.3	−1.4
7200	107.8281427	35.79857622	5.8	−0.671	−0.224	−5.816	143.7	6.0	−1.3
7300	107.8281424	35.79857557	6.1	−0.671	−0.224	−5.816	143.6	5.3	−0.6
7400	107.828142	35.79857462	6.4	−0.671	−0.224	−5.816	143.8	4.7	−1.0
7500	107.8281416	35.79857388	6.6	−0.895	−0.224	−5.816	144.0	4.7	−2.1
7600	107.8281412	35.79857291	6.9	−0.895	−0.224	−5.816	143.7	5.1	−2.7

Table 3. Summary of the collected data for the drone.

Type of Drone	Net Weight/g	Max Speed (m/s)	Max Ascent Speed (m/s)	Max Decent Speed (m/s)	Number of Trajectory Records	Ground Speed (m/s)			Vertical Speed (m/s)		
						Min	Max	Mean	Min	Max	Mean
Mavic Pro	743	18.0	5.0	3.0	11,576	0.00	18.00	2.33	−18.70	19.08	−0.01
Mavic Air	430	19.0	3.0	3.0	8338	0.00	19.55	1.64	−5.92	8.90	−0.01
Mavic 2	905	20.0	5.0	3.0	19,634	0.00	20.60	2.40	−9.86	11.60	0.00

(2)   Data processing for drones

Step 2-1: Noise data processing

The data of unstable trajectory points existing in the process of collecting data during the initial flight of the drone is cleaned up. The cleaned drone trajectory data are divided into equal interval data according to the interval of the above-mentioned time series data for manned aircraft.

Step 2-2: Coordinate system transformation

The drone trajectory data processed in Step 2-1 are presented in the form of longitude and latitude in the World Geodetic System 1984 (WGS-84) Coordinate System. The longitude and latitude in the coordinate system are converted to the vertical and horizontal values in the ECEF Cartesian coordinate system.

Step 2-3: Dimensional processing and sample segmentation

The drone trajectory data in Step 2-2 are normalized to eliminate the influence of dimension. The normalized trajectory data are divided into fixed-length time series slices using a sliding time window, and a sliced sample set containing the input and output data in the trajectory prediction model is generated. Each slice sample set includes nine

variables: latitude, longitude, altitude, speed in the horizontal axis, speed in the vertical axis, vertical speed, heading angle, pitch angle, and roll angle.

(3)   The protection zone of manned aircraft

Conflicts between drones and manned aircraft mostly occur in the take-off and landing stages of manned aircraft. Considering that the flight height of light rotor drones is usually in the range of 0–300 m, the flight height of 0–300 m (meters) is divided into height intervals of 100 m.

The flight speed of the manned aircraft in the take-off phase and the landing phase changes rapidly with the altitude. Based on the divided flight altitude intervals, the separation distance for the safe operation of manned aircraft is set according to the average aircraft speed at the corresponding altitude intervals. The protected area of the manned aircraft is a cylinder with the predicted position of the manned aircraft as the center. For the altitude of 0–100 m, the horizontal distance from the boundary to the center is 609.6 m, and the vertical distance is 76.2 m. In the altitude range of 100–200 m and 200–300 m, the separation distance for the safe operation of the drone and manned aircraft in the horizontal and vertical directions is expanded in proportion to the average speed of manned aircraft. Therefore, the protected areas for manned aircraft in the three altitude intervals can be obtained.

3. Prediction of Manned Aircraft Trajectory in Terminal Areas

In this section, three types of models are used to predict the trajectory of manned aircraft, including the gated recurrent unit (GRU) structure, the long short-term memory (LSTM) network and the support vector regression (SVR) structure. The models are described as follows.

3.1. Model Construction

(a)   The gated recurrent unit structure

The gated recurrent unit (GRU) structure is utilized to train the neural network for manned aircraft trajectory prediction. The loss function is expressed as:

$$Loss = \frac{1}{n} \sum_{i=1}^{n} (y(x)_i - y_i)^2 \tag{2}$$

where $y(x)_i$ is the predicted position of the aircraft at the time i; and y_i is the actual position of the aircraft at the time i.

In the process of neural network training, to minimize the loss function, the stochastic gradient descent method is used to solve the problem. The minimized loss function and corresponding model parameters are obtained. Figure 1 depicts the GRU architecture for aircraft trajectory prediction.

As mentioned above, the small drone will only collide with the manned aircraft during the take-off and landing phases, and for each trajectory, data in the take-off and landing phases are filtered. With the longitude, latitude, altitude, ground speed, and heading angle of the manned aircraft, the input to the network is a three-dimensional dataset with the shape of the batch number of samples * number of features * timestep. The data samples are divided into the training sets, validation sets, and test sets according to the ratio of 80%, 10%, and 10%, respectively.

The structural parameters and internal parameters of the neural network are initialized, including the number of hidden layers, the number of neurons in a single layer, the number of batch training, the number of iterations, and the learning rate. The neural network parameters are iteratively updated by using the manned aircraft trajectory training dataset. In each epoch process, a certain number of samples are randomly selected from the trajectory data training set, and the loss function value is calculated according to the data labels corresponding to the actual value of the aircraft trajectory. The test set is substituted into the trained network structure for performance testing.
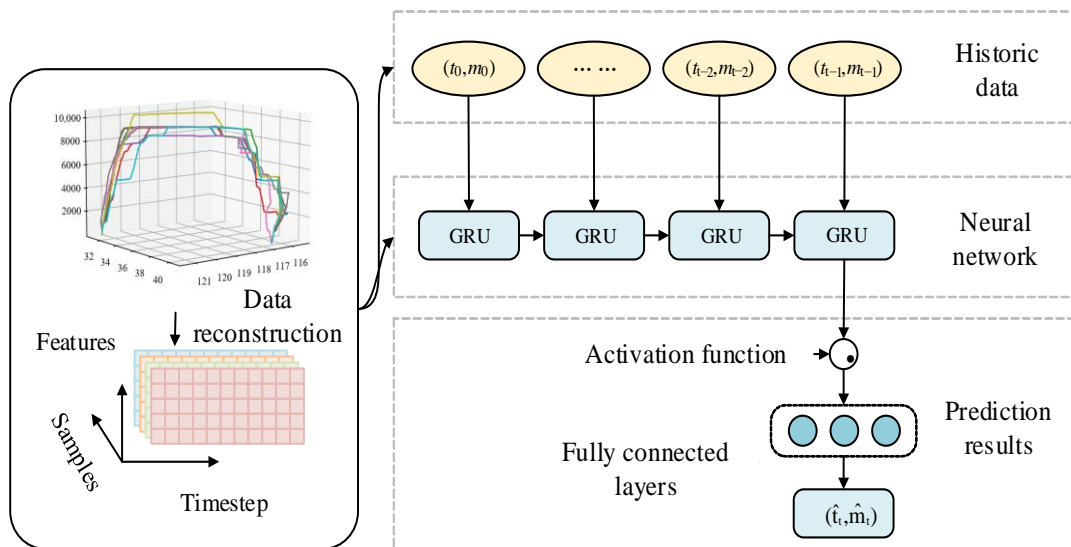
Figure 1. The GRU architecture for aircraft trajectory prediction.

(b)　　The long short-term memory structure

The long short-term memory (LSTM) network was proposed by Hochreiter and Schmidhuber for addressing the vanishing gradients problem in RNNs [30]. It has been proven to perform well on sequence-based tasks with long-term dependencies. Compared with the traditional artificial neural network, the LSTM network realizes the combination of long-term and short-term memory by setting special structures such as the forgetting gate, input gate and output gate. Based on the LSTM network, the historical data of the drone trajectory, including the longitude, latitude, altitude, ground speed, and heading angle of the manned aircraft, are input into the model. The output is the trajectory prediction result in the next moment. Based on the standard LSTM network, the learning rate is automatically adjusted to avoid accuracy degradation and local convergence caused by rapid convergence.

(c)　　The support vector regression structure

Support vector regression (SVR) is a supervised learning algorithm that can be used to predict time series data. SVR trains using a symmetrical loss function, which equally penalizes high and low misestimates. A flexible tube of minimal radius is formed symmetrically around the estimated function. In this way, the absolute values of errors less than a certain threshold ε are ignored both above and below the estimate. The points outside the tube are penalized, but those within the tube, either above or below the function, receive no penalty. One of the main advantages of SVR is that its computational complexity does not depend on the dimensionality of the input space. Additionally, it has excellent generalization capability, with high prediction accuracy [31]. The inputs and outputs of the model are similar to those in the LSTM and GRU models.

3.2. Evaluation Metrics

The average Euclidean distance error (EDE), which represents the distance between the actual and predicted position of the manned aircraft, is used to evaluate the performance of the constructed model. The actual positions are taken from the latitude, longitude, and altitude of the trajectories after coordinate transformation. The equation is shown as follows.

$$EDE = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(\overline{x_i} - x_i)^2 + (\overline{y_i} - y_i)^2 + (\overline{z_i} - z_i)^2} \qquad (3)$$

Among them, n is the number of predicted samples; $\overline{x_i}$ and x_i are the predicted and the actual longitude of the aircraft at the time i; $\overline{y_i}$ and y_i are the predicted and the actual

latitude of the aircraft at the time i; $\overline{z_i}$ and z_i are the predicted and the actual altitude of the aircraft at the time i, respectively.

3.3. Model Training and Testing

Before training, the parameters for the neural network need to be determined. Different parameters for the input layer and output layer of the GRU trajectory prediction model are compared first. After comparison, it is finally determined that the number of training samples in a single batch is five; the timestep of the input network is twenty; the timestep of the input network in the landing phase is twenty-five. In the take-off phase, the input data dimension of each sample is set as [20, 4]. In the landing stage, the input data dimension of each sample is [25, 4]. The output data dimension is [1, 3]. The initial parameters are randomly selected in the [0, 1] interval.

The other parameters for the GRU trajectory prediction model are also compared to determine an appropriate value. The maximum number of iterations of the network in the take-off stage and the landing stage are both set to be 200. The learning rate is 0.0001. The number of hidden layers in the take-off and landing stage is set to three. The number of neurons in a single hidden layer is 90 for both the take-off and landing stages. The initial parameters are randomly selected in the [0, 1] interval, and Sigmoid is selected as the activation function.

For the LSTM model, the network adopts three hidden layers with 100 neurons in each layer. The timestep of the input network is 35. The learning rate is 0.0001, and the number of training samples in a single batch is five. The activation function is ReLU. The model takes five main features as input, including latitude, longitude, altitude, horizontal velocity, and vertical velocity, to generate time series data for each variable.

The computer environment for this study is configured as Win 10 (64-bit) operating system, Intel(R) Core(TM) i5-8265U 1.60 GHz processor, 16 GB memory, and 4 GB video memory. The Python 3.8 programming language, Pycharm-based development environment, Tensorflow, and Keras model framework are utilized for model training and testing.

3.4. Trajectory Prediction Results of the Manned Aircraft

To testify the performance of the GRU model, two commonly used trajectory prediction methods are selected as benchmark methods, including the LSTM and SVR models. Since the trajectory prediction results of manned aircraft are mainly used to evaluate the short-term collision risk between manned aircraft and drones, the prediction time length is initially set to be 20 s. Through the model prediction errors, the prediction performances of the three different models are compared for the take-off phase and the landing phase, respectively. A comparison of prediction errors in longitude, latitude, and height are shown in Table 4 and Figure 2.

The prediction timestep is gradually increased in one second time intervals to obtain the prediction error. It can be seen from Figure 2 that, in the take-off stage, the prediction error of GRU is about 400 m for the 20 s time interval. The prediction error of LSTM is slightly larger than that of GRU. The prediction error of SVR has reached 500 m for the 10 s time interval. This prediction error is generally comparable with the results in previous studies, which ranges from several hundred meters to over 1000 m during the take-off and landing stages [6,32].

Table 4. The prediction error for manned aircraft in the take-off and landing stages.

Prediction Timestep	Method	*EDE* for the Take-Off Stage (m)	*EDE* for the Landing Stage (m)
5 s	GRU	151.95	193.10
	LSTM	270.44	311.81
	SVR	350.05	484.55
10 s	GRU	292.69	411.35
	LSTM	389.81	540.66
	SVR	524.62	740.65
15 s	GRU	363.65	541.06
	LSTM	425.01	728.62
	SVR	762.20	899.54
20 s	GRU	416.33	620.61
	LSTM	454.67	846.69
	SVR	926.82	1085.70
25 s	GRU	443.99	641.25
	LSTM	501.54	870.42
	SVR	1458.09	1831.64
30 s	GRU	515.81	760.49
	LSTM	602.21	896.20
	SVR	1630.39	2060.62
35 s	GRU	536.05	872.92
	LSTM	660.77	935.91
	SVR	1847.63	2486.47
40 s	GRU	583.63	960.82
	LSTM	725.68	955.61
	SVR	1989.57	2603.30
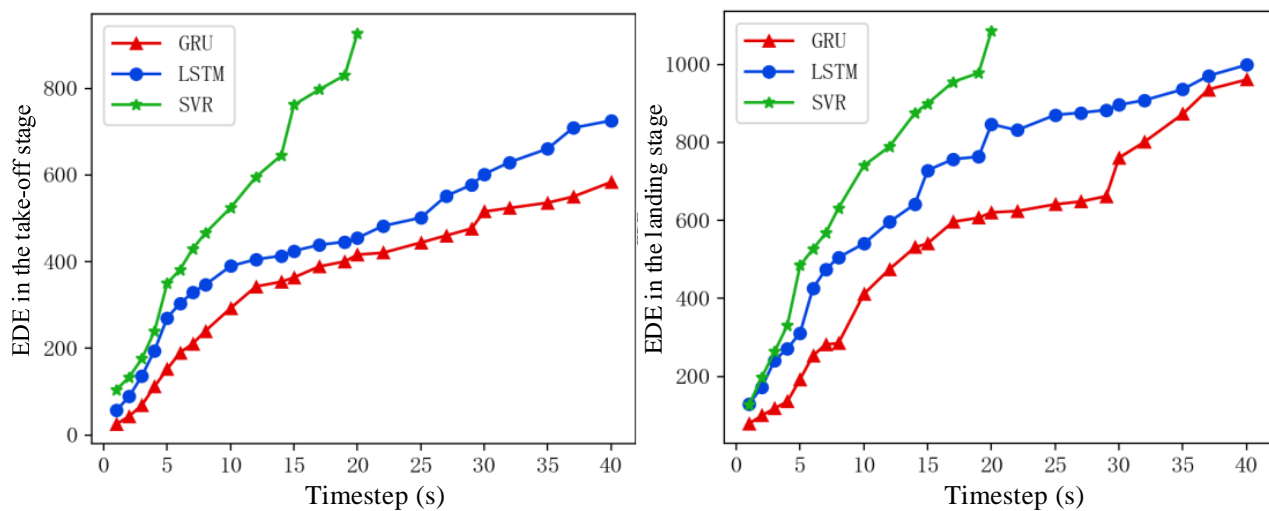
Figure 2. Performance comparison for three models.

In the landing stage, for the prediction timestep of 20 s, the prediction error of GRU is about 600 m, the prediction error of LSTM is about 800 m, and the prediction error of SVR is as high as 1000 m. It can be seen that the prediction error of the SVR model is significantly larger than that of the GRU and LSTM models. As the prediction timestep increases, the error shows an overall upward trend. By comparing Figure 2a,b, it can be seen that the error in the landing stage is generally higher than that in the take-off stage, indicating that the manned aircraft trajectory prediction model has better performance in the take-off stage.

4. Prediction of Distribution of Drones

In this section, a quartile regression bidirectional gate recurrent unit neural network (QRDGRU) is established to predict the trajectory of drones. The proposed model is compared with traditional GRU, SVR, quartile regression long short-term memory (QRLSTM), and quartile regression gate recurrent unit (QRGRU) models to testify its performance. The proposed model structure is described as follows.

4.1. Model Construction

The traditional GRU model can only predict the point position of the aircraft. Considering the flexible and changeable characteristics of the drone, this paper constructs a quartile regression bidirectional gate recurrent unit neural network (QRDGRU). This QRDGRU model is an improved model from the bidirectional gated recurrent unit (D-GRU) structure from the authors' previous research. The D-GRU model integrates the prediction based on the change of location for two adjacent points, as well as the prediction based on the change of location at intervals of two points [24]. In this research, the quartile regression structure is further incorporated. The QRDGRU model can provide the position distribution of the drone to compensate for possible irregularities in the mapping learned by the neural network layer when sampling at a single interval, from which the air transport managers can have a better knowledge of the position of the drone.

Similar to GRU, the QRDGRU structure includes two gates, update gate z_t and reset gate r_t, to control input and output values. The essence of QRDGRU is the improvement of data processing and loss function. Equation (4) demonstrates the loss function of QRDGRU:

$$
\begin{aligned}
f_{loss} &= \min \sum_{i=1}^{N} \rho_\tau [Y_i - f(X_i, W_{\tau_i}, b_{\tau_i})] \\
&= \min \left[\sum_{Y_i \geq f(X_i, W_{\tau_i}, b_{\tau_i})} \tau |Y_i - f(X_i, W_{\tau_i}, b_{\tau_i})| + \sum_{Y_i < f(X_i, W_{\tau_i}, b_{\tau_i})} (1-\tau)|Y_i - f(X_i, W_{\tau_i}, b_{\tau_i})| \right]
\end{aligned}
\tag{4}
$$

The estimation of the QRDGRU parameters can be regarded as an optimization problem of parameters in Equation (4). The parameters are iteratively optimized by backpropagation through the network. Among them, X_i is the time series trajectory input of the drone; Y_i is the true position of the drone; $f(X_i, W_{\tau i}, b_{\tau i})$ is the predicted value of the neural network; and τ is the quartile, usually in the range of (0, 1); $W_{\tau i}$ is the weight matrix of the neural network under quartile τ; $b_{\tau i}$ is the bias parameter of the neural network under quartile τ. Figure 3 shows the structure of the QRDGRU model. The QRDGRU takes the sample sequences at two time intervals as the input of two parallel GRU stacked layers, and combines the extracted features with fully connected layers, and the latitude, longitude, altitude, horizontal speed, vertical speed, heading angle, pitch angle, and roll angle are learned by the QRGRU under each quartile. After that, the network output value under each quartile is obtained, and the loss is calculated through the regression layer.

The QRDGRU obtains the optimal estimation of its parameters by continuously adjusting $W_{\tau i}$ and $b_{\tau i}$, thereby training the network under different quartiles, and obtaining the predicted value of the drone position under different quartiles according to the learned network.

$$
Q_Y(\tau|X) = f(X, \hat{W}_\tau, \hat{b}_\tau)^2
\tag{5}
$$

Among them, $Q_Y(\tau \mid X)$ is the predicted position of the drone under quartile τ when the trajectory data of the drone is input as X; $\hat{W}_\tau$ and $\hat{b}_\tau$ are the optimal weight matrix and optimal bias parameter of the network under quartile τ, respectively; τ is a continuous value between (0, 1). The predicted position of the drone under different quartiles can form the predicted position distribution of the drone. The output of the QRDGRU model $Q_Y(\tau \mid X)$ under different quartiles is the conditional distribution of the predicted position of the drone.

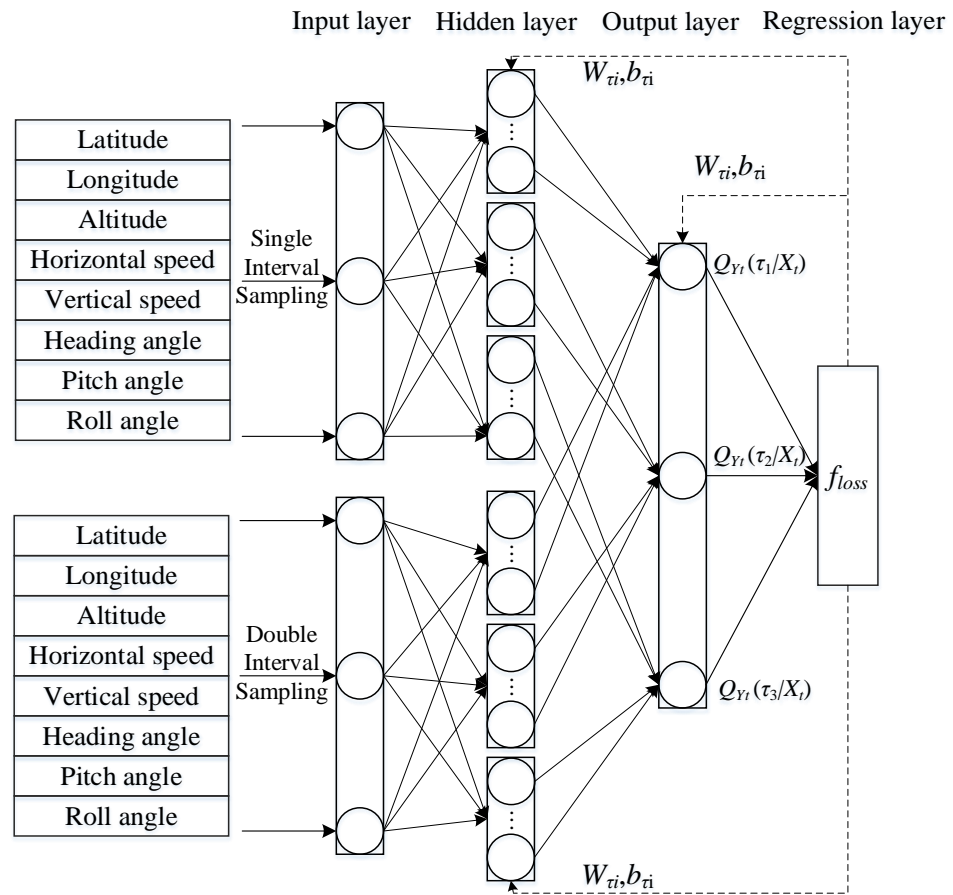Input layer   Hidden layer   Output layer   Regression layer

Figure 3. Structure of the QRDGRU model.

4.2. Evaluation Metrics

To evaluate the performance of the proposed model, mean absolute error (MAE), root mean squared error (RMSE), and mean absolute percentage error (MAPE) are calculated for each method, respectively. The equations are shown as follows.

$$MAE_\tau = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{x}_{i|\tau} - x_i\right| + \left|\hat{y}_{i|\tau} - y_i\right| + \left|\hat{z}_{i|\tau} - z_i\right| \tag{6}$$

$$RMSE_\tau = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{x}_{i|\tau} - x_i\right)^2 + \left(\hat{y}_{i|\tau} - y_i\right)^2 + \left(\hat{z}_{i|\tau} - z_i\right)^2} \tag{7}$$

$$MAPE_\tau = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\hat{x}_{i|\tau} - x_i}{x_i}\right| + \left|\frac{\hat{y}_{i|\tau} - y_i}{y_i}\right| + \left|\frac{\hat{z}_{i|\tau} - z_i}{z_i}\right| \tag{8}$$

Among them, n is the number of predicted samples; $\hat{x}_{i|\tau}$ and x_i are the predicted and true latitude of the drone under quartile τ at time i, respectively; $\hat{y}_{i|\tau}$ and y_i are the predicted and true longitude of the drone under quartile τ at the time i, respectively; $\hat{z}_{i|\tau}$ and z_i are the predicted and true altitude of the drone under quartile τ at the time i, respectively.

4.3. Model Training and Testing

The latitude, longitude, altitude, speed in the x, y, and z-axis, heading angle, pitch angle, and roll angle of the drone are input into the neural network. The outputs are the predicted longitude, latitude, and altitude of the drone. The specific steps are as follows:

Step 1: Data processing. The collected drone trajectory data are processed using a sliding window, and spatial reconstruction and normalization are performed to construct samples that can be input into the QRDGRU. Assuming that the number of drone trajectory data is k, the input of a sample is a matrix of $k \times 9$, for which the i-th row of X is $(x_i, y_i, z_i, v_{xi}, v_{yi}, v_{zi}, \varphi_i, \theta_i, \gamma_i)$,

and the target matrix corresponding to X is $[x_{i+k+s}, y_{i+k+s}, z_{i+k+s}]$, where s is the prediction timestep.

Step 2: Network training. The processed drone trajectory data set is divided into the training set and testing set. The training set is input into the $QRDGRU_i$ under different quartiles, and the optimal weight and bias parameters are obtained. The validation set is input into the trained $QRDGRU_i$. The hyperparameters of the network, such as the number of iterations and the number of neurons, are determined in the network.

Step 3: Predict the drone location distribution. Based on the constructed $QRDGRU_i$ with determined hyperparameters, the outputs including latitude, longitude, and altitude are obtained under each quartile. The outputs are denormalized to obtain the predicted value of the $QRDGRU_i$. Then, the predicted position distribution of the drone is formed.

Step 4: Model evaluation. Compare the predicted location distribution of the drone with the actual location for model evaluation.

The experimental environment configuration is the same as that of the manned aircraft. It is finally determined that three hidden layers are used, the number of nodes in the input layer is nine, and the number of nodes in the output layer is three. The activation function is "ReLU". The learning rate is set to 0.0001. The number of training iterations is set as epoch = 100. For each batch, ten samples and nine feature variables are trained, generating time series data T for each variable. Therefore, the actual input data of a batch of samples are three-dimensional vectors with the shape $B \times N \times T$, where B is the number of batch samples, N is the number of features, and T is the timestep.

The hyperparameters of the network are determined according to MAE, MAPE and RMSE, including the number of neurons in a single layer c and the timestep T. It is finally determined that the number of neurons in a single layer is 80, and the timestep of the QRDGRU model is T = 25.

4.4. Trajectory Prediction Results of the Drones

To testify the performance of the QRDGRU model, several benchmark trajectory prediction models are constructed and compared, including the GRU, SVR, QRLSTM and QRGRU models. As a traditional classific model, the GRU model was first applied to justify whether to predict the three-dimensional coordinates at one time or to predict the three-dimensional coordinates separately. The prediction accuracies in the two ways are compared. The hyperparameters for the models are provided in Table 4 and the results are summarized as Model 1 and Model 2 in Table 5. The prediction performance indicators include RMSE, MAE, MAPE, and consumed time. The computer in this research is equipped with the R5-3600 CPU, with the frequency of 4.00 GHz and batch size of 100.

It can be seen that all the models converge rapidly within several seconds. For the GRU model, the accuracy acquired by predicting the three-dimensional coordinates together tends to be higher than by predicting three-dimensional coordinates separately, indicating the fact that the drone's movements in the three axes are related. Therefore, in the following models, the three-dimensional coordinates were input into the model together.

Comparing the prediction results from Model 1 to Model 6 in Table 6 and Figure 4, it can be found that the prediction performance indicators RMSE, MAE, and MAPE increase with the increase of timestep, due to the uncertainty of the drones' movement. In general, the QRDGRU model has the highest and most stable prediction accuracy, as compared with the SVR, GRU, QRLSTM, and QRGRU models for different timesteps. The proposed QRDGRU method improves the model structure by integrating two different prediction frameworks, including the D-GRU model proposed in our previous research as well as the quartile regression. The performance indicators confirm the superiority of the established model.

Table 5. The hyperparameters for the models.

Model	Parameters		
	Timestep (s)	Number of Neurons in a Layer	Number of Hidden Layers
GRU (predicting three-dimensional coordinates together)	10	80	3
GRU(predicting three-dimensional coordinates separately)	20	80	3
QRGRU	25	80	3
QRLSTM	35	80	3
QRDRGRU	25	80	3
SVR	**Timestep (s)**	**c**	**kernel**
	10	0.1	linear

Table 6. Prediction performance under different timesteps for the six models (m).

Timestep (s)	RMSE (m)	MAE (m)	MAPE (%)	Time (s)	RMSE (m)	MAE (m)	MAPE (%)	Time (s)
	Model 1: GRU (Predicting Three-Dimensional Coordinates Together)				Model 2: GRU (Predicting Three-Dimensional Coordinates Separately)			
5	20.59	13.18	2.97	0.516	34.12	21.56	8.90	1.696
10	45.46	32.48	5.08	0.589	53.12	42.89	20.93	1.699
15	76.92	55.50	7.68	0.521	110.43	76.99	30.01	1.671
20	104.00	74.79	16.57	0.519	162.92	156.11	60.67	1.735
25	125.70	90.92	16.49	0.575	186.34	168.13	58.88	1.766
30	147.06	110.01	23.27	0.538	215.77	202.43	72.06	2.004
35	158.33	123.32	22.13	0.523	250.35	249.94	57.89	1.750
40	175.25	127.84	39.14	0.502	265.55	264.99	57.55	1.680
	Model 3: SVR				Model 4: QRLSTM			
5	19.63	17.42	10.77	0.193	18.33	12.10	2.90	0.622
10	55.71	50.69	15.91	0.274	49.41	32.81	5.88	0.626
15	93.78	90.73	28.63	0.260	73.31	55.78	13.49	0.600
20	135.17	131.20	44.10	0.261	104.13	75.10	13.41	0.602
25	168.73	166.85	49.95	0.254	123.46	90.93	12.38	0.606
30	192.88	196.12	61.15	0.265	139.35	100.76	17.13	0.619
35	219.62	220.05	53.70	0.302	165.95	114.36	27.17	0.651
40	248.97	234.77	62.40	0.265	161.91	120.59	19.24	0.615
	Model 5: QRGRU				Model 6: QRDGRU			
5	16.35	10.54	2.38	0.356	15.65	10.25	2.10	0.780
10	43.75	29.50	6.48	0.349	40.95	28.21	5.64	0.898
15	68.70	50.20	9.37	0.403	69.22	50.62	8.84	0.974
20	94.58	70.89	14.60	0.358	94.96	71.30	12.66	0.838
25	119.47	90.54	19.26	0.350	115.16	88.88	14.22	0.764
30	138.35	96.09	16.06	0.390	145.74	105.49	13.29	0.867
35	154.14	115.04	22.89	0.365	155.83	115.58	16.59	0.838
40	172.77	118.27	15.43	0.389	161.73	120.39	14.90	0.885

According to the QRGRU method, the distribution of the probability for the drone with a specifically predicted timestep can be obtained. Figure 5 depicts the position distribution in the horizontal plane of the drone at a specific moment in the future for the 10 s, 20 s, 30 s, and 40 s timesteps, respectively. The darker the colour, the higher the position distribution probability is. It can be seen that the area of predicted location spreads with the increase of timestep, which further indicates the uncertainty of the drones' movement. The blue circle indicates the maximum uncertainty calculated using the maximum acceleration and speed for the given time-step, which is a circle with the initial position of the drone as its center and the furthest distance that can be arrived as its radius. The furthest distance is calculated based on the assumption that the drone accelerates with its initial speed to the maximum speed and then flies with the maximum constant speed to reach the largest distance within a given timestep. For the Mavic 2 drone, the maximum acceleration rate is 6 m/s^2 and the maximum speed is 20 m/s.
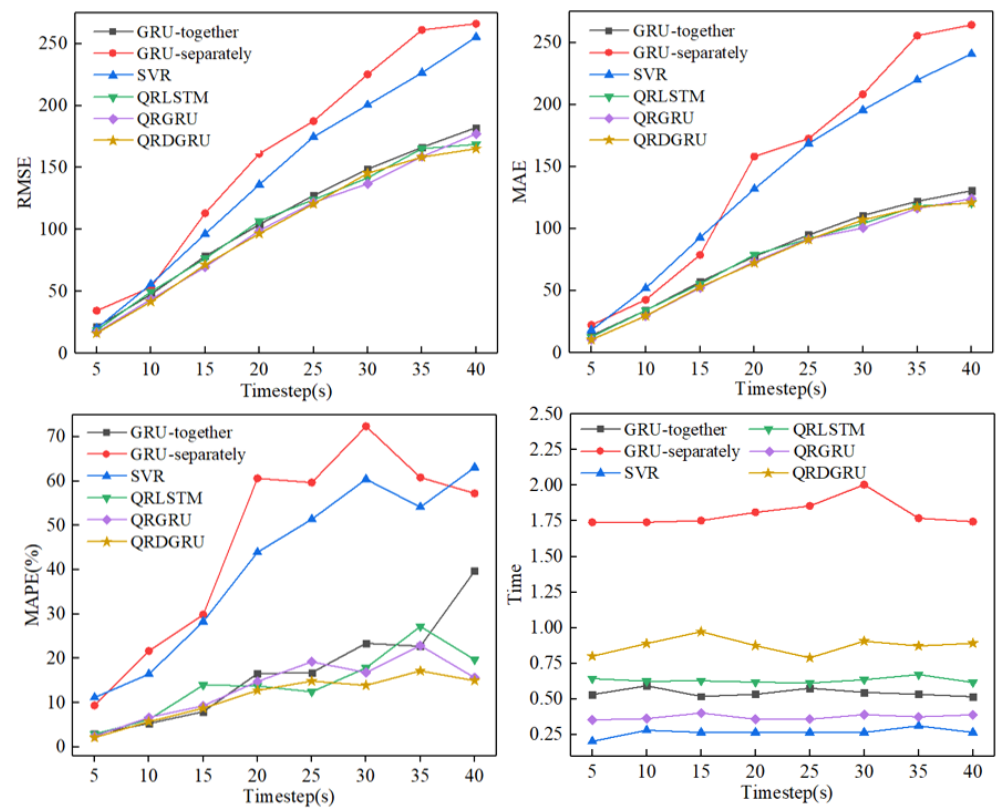
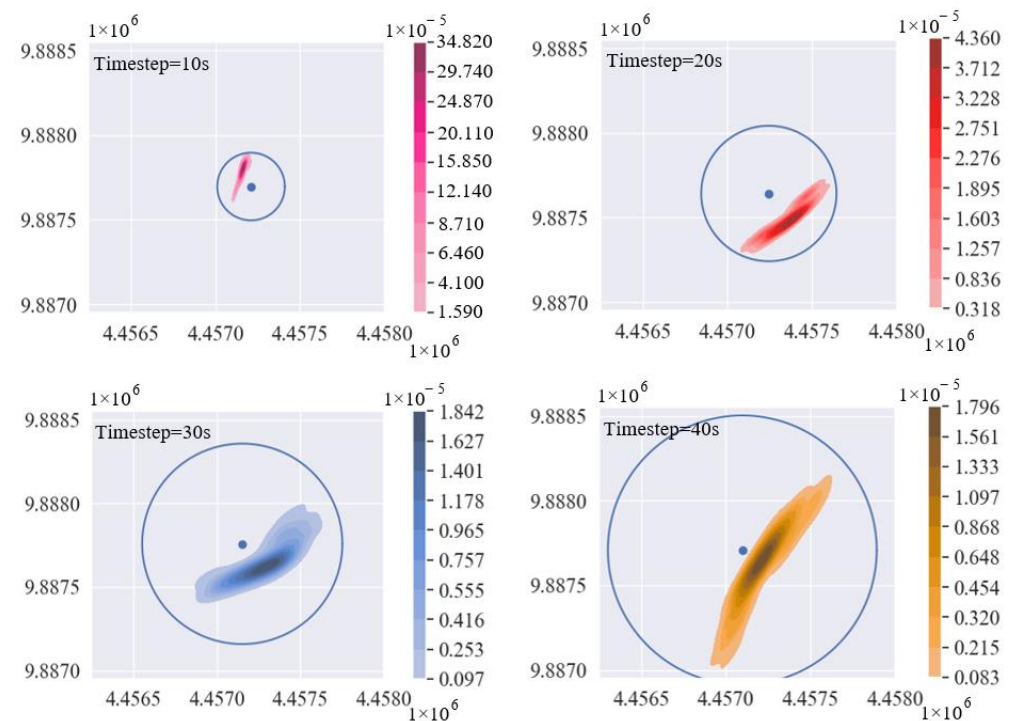**Figure 4.** Performance comparison of the trajectory prediction models for drones.

Figure 5. Illustration of the position distribution for the drone.

5. Conflict Risk Assessment between Drones and Manned Aircraft

Based on the predicted position of the manned aircraft and the predicted position distribution of the drone obtained in Sections 3 and 4, it is determined whether a conflict risk exists between the drone and the manned aircraft. Additionally, the conflict probability

between the drone and the manned aircraft with the conflict risk is calculated. The specific process is as follows:

First, it is determined whether there is an intersection between the predicted position of the drone under each quartile at the prediction time and the protected area of the predicted location for the manned aircraft. If the predicted value of the drone at the τ quartile at the i-th time is in the protected area of the manned aircraft in the x, y, and z directions, that is, when Formula (9) is satisfied, it means that the drone is in the risk of conflict with manned aircraft. The horizontal and vertical distance between the predicted position of the drone and the protected area of the manned aircraft is judged as follows:

$$\left\{ \begin{array}{c} \sqrt{(O(x)_{\tau i} - P(x)_i)^2 + (O(y)_{\tau i} - P(y)_i)^2} < dis_e \\ |O(z)_{\tau i} - P(z)_i| < dis_f \end{array} \right. \tag{9}$$

In the formula, dis_e is the horizontal distance between the predicted position of the drone and the protected area of the manned aircraft; dis_f is the vertical distance between the predicted position of the drone and the protected area of the manned aircraft; $O(x)_{\tau i}$, $O(y)_{\tau i}$, $O(z)_{\tau i}$ are the predicted values of the drone at the τ quartile at the i-th time in the x, y, and z axes; $P(x)_i$, $P(y)_i$, $P(z)_i$ are the predicted values of the manned aircraft in the x, y, and z axes.

Second, the probability of collision between the drone and the manned aircraft at the predicted time can be estimated. According to the conflict risk assessment, the number of predicted position points of the drone under all the quartiles at the prediction time in the protected area of the manned aircraft is obtained. The ratio of the number in the protected area to the number of total predicted position points of the drone under all quartiles is the collision probability between the drone and the manned aircraft at the predicted time. The collision probability $Prob_i$ between the drone and the manned aircraft at the i-th moment is expressed as:

$$Prob_i = \frac{\sum\limits_{\tau=1}^{m} b_{\tau|i}}{m} \tag{10}$$

In the formula, m is the number of predicted position points of the drone under all quartiles, and $b_{\tau|i}$ represents the conflict between the predicted position of the drone and the manned aircraft under the τ quartile at the i-th time. When there is a conflict risk, $b_{\tau|i}$ takes the value of 1, and otherwise, $b_{\tau|i}$ is 0.

The conflict probability threshold is set to 50%. The threshold is the same as that set in previous research [27,29]. If the probability of conflict at the predicted time is greater than or equal to the conflict probability threshold, it is considered that there is a conflict between the drone and the manned aircraft. If it is less than the conflict probability threshold, it is considered that the drone and the manned aircraft are not in conflict.

Then, the correct detection rate and false alarm rate of collision probability prediction are calculated. In reality, it is difficult to carry out a collision test between a manned aircraft and a quadrotor. Therefore, a collision simulation test is carried out based on the real trajectory data, which not only ensures that the trajectory data in the simulation test conforms to the actual flight situation but also reduces the test risk and cost. In this research, Monte Carlo simulation experiments are conducted. In the simulation experiment, the ratio of the number of correctly predicted collisions to the number of real collisions between the drone and manned aircraft is the correct detection rate. The ratio of the number of incorrectly predicted collisions to the number of non-collisions is the false alarm rate.

$$Per_{cd} = \frac{\sum\limits_{r=1}^{Num} q_r = 1}{Rel_{col}} \tag{11}$$

$$Per_{fa} = \frac{Pre_{col} - \left(\sum\limits_{r=1}^{Num} q_r = 1 \right)}{Num - Rel_{col}} \tag{12}$$

In the above formulas, *Num* is the total number of simulation tests; Per_{cd} is the correct detection rate; Per_{fa} is the false alarm rate. $q_r = 1$ indicates the correct prediction of real collisions in the *r*-th simulation test, which means that both the predicted results and the real results between the drone and manned aircraft are collisions; Rel_{col} is the number of real collisions in the simulation test; and Pre_{col} is the number of collisions predicted in the simulation test.

Finally, the collision probability between the drone and the manned aircraft at the closest encountering point and the estimated time to reach the closest encountering point are calculated. The following assumptions are made:

- The drone and the manned aircraft are regarded as a particle with a direction of movement;
- During the movement, the drone and the manned aircraft are independent of each other, and the influence of the wake factor is excluded;
- The effects of bad weather such as wind, rain, and thunderstorms are ignored;
- The horizontal and vertical movements of the drone are independent;
- The speed of the drone and the manned aircraft remain consistent at the closest moment.

Based on the above assumptions, it can be judged whether there is a conflict risk according to the predicted position points of the manned aircraft and the predicted position distribution of the drone. The drones that do not have a conflict risk are screened out. For the remaining drones with conflict risks, assuming that the drone and the manned aircraft are predicted in the next *k* seconds from the current time *n*, the minimum distance between the predicted position of the manned aircraft and the drone at the same time is defined as the closest encountering point.

For any time $n + s$ within the time $n + 1 \sim n + k$, calculate the distance (L^{μ}_{n+s}) between the predicted position of the manned aircraft at the time of $n + s$ for each point in the predicted position distribution of the drone trajectory.

$$L^{\mu}_{n+s} = \sqrt{\left(x_{n+s} - u^{\mu}_{n+s}\right)^2 + \left(y_{n+s} - v^{\mu}_{n+s}\right)^2 + \left(z_{n+s} - w^{\mu}_{n+s}\right)^2} \\ \mu = 1, 2, \ldots, j \\ s = 1, 2, \ldots, k \tag{13}$$

where x_{n+s}, y_{n+s}, z_{n+s} are the predicted longitude, latitude and altitude, respectively, for manned aircraft at time $n + s$; u^{μ}_{n+s}, v^{μ}_{n+s}, w^{μ}_{n+s} are the predicted longitude, latitude and altitude, respectively, for the drone at time $n + s$; *j* is the number of quartiles. The minimum distance between the drone and the manned aircraft at time $n + s$ (L_{n+s}) is calculated as:

$$L_{n+s} = \min(L^{\mu}_{n+s}) \\ \mu = 1, 2, \ldots, j \tag{14}$$

According to the obtained minimum distance between the drone and the manned aircraft at the time $n + s$, the minimum distance between the drone and the manned aircraft from time $n + 1 \sim n + k$ (L_{n+l}) are obtained.

$$L_{n+l} = \min(L_{n+s}) \\ s = 1, 2, \ldots, k \tag{15}$$

Thus, $n + l$ is the timestamp when the drone and the manned aircraft meet at the nearest point, and *l* is the time when the drone and the manned aircraft are expected to reach the nearest encountering point from the current time.

The procedure is illustrated using the following example. Based on the trajectory data of the take-off stage and the landing stage of the manned aircraft and the collected drone trajectory data, the drone and the manned aircraft are placed in the same space by means of translation, and simulation experiments are carried out within four conflict risk levels.

Figure 6 illustrates the conflict scenario based on the predicted drone and aircraft trajectory. The cylinder is the protected area of the manned aircraft when they meet at the nearest point; the solid line is the historical trajectory of the drone; the dotted line is the historical trajectory of the manned aircraft; the asterisk * is the predicted position of

the manned aircraft when it meets at the nearest point. The continuous curve predicts the location distribution for the drone.
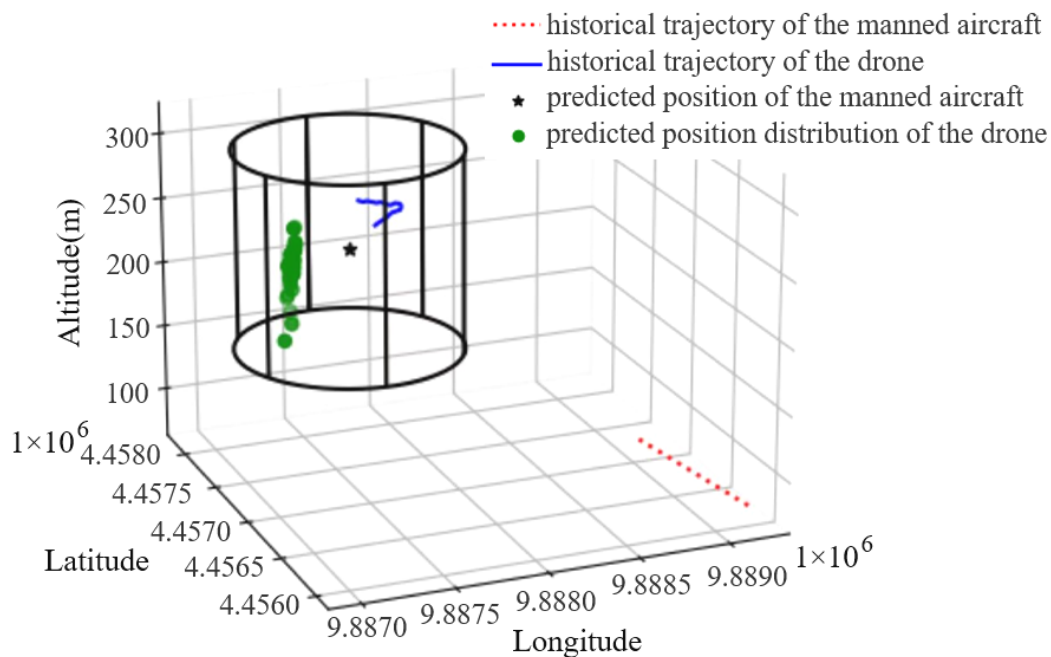

Figure 6. A typical conflict scenario based on the predicted drone and aircraft trajectory.

It can be seen that most of the predicted drone trajectory points fall within the manned aircraft protection zone when arriving at the closest encountering point. The closest distance at the encountering point is 457.27 m. According to the current historical trajectory of the drone and the manned aircraft, it is predicted that the closest encountering point between the drone and the manned aircraft will be 18 s later. With Formula (8), it is estimated that the collision probability between the drone and the manned aircraft at the closest point is 81.63%. The air transport management department can make decisions based on the above simulation results.

The Monte Carlo simulation tests were conducted at each flight level during the departure and arrival stages. The prediction accuracy of collision probability at different flight stages and levels can be obtained, as shown in Table 7. With the increase of altitude, the prediction accuracy of collision probability decreases. This is because the higher the altitude, the higher the flight speed. The mean absolute error for the trajectory prediction of the manned aircraft will be reduced as the altitude increases, which affects the collision probability prediction accuracy.

Table 7. Conflict prediction accuracy under different flight levels.

Altitude (m)	The Take-Off Stage		The Landing Stage	
	Per_{cd}	Per_{fa}	Per_{cd}	Per_{fa}
0–100	97.37%	8.20%	96.97%	5.90%
100–200	91.07%	11.63%	94.44%	12.70%
200–300	88.89%	6.70%	90.00%	16.32%

Naturally, there is a tradeoff between correct detections and false alarms. For example, with a larger protection zone of the manned aircraft, the correct detection rate tends to be higher while the false alarm rate also tends to be higher. The performance indicators are also affected by the type of intruder. For example, it has been demonstrated in previous research that when the correct detection rate is around 80%, the false alarm rate is around 8% for

Zagi size as the intruder, 35% for ScanEagle size, and 90% for Cessna size as intruder [33]. In another study, alert results from 10,000 simulated unmanned aerial system tracks indicate that, to obtain a relatively high failure rate, the false alarm rate ranges from 18.89% to 71.38% based on different methods [27]. Overall, the correct detection rate ranges from 88.89% to 97.37% and the false alarm rate ranges from 5.9% to 16.32% in the current research. The model estimation results are generally comparable with those in previous research.

6. Conclusions and Discussion

Collision risk assessment between manned aircraft and drones is a key technology for the safe operation of air transport in low-altitude airspace. This paper begins with the problem of the illegal flight of drones, leading to the risk of collision with manned aircraft in the take-off and landing stages. A short-term risk assessment method between manned aircraft and drones has been implemented to identify the collision risk and enhance the safe operation of air traffic. The main findings of this paper are as follows:

First, based on the historical ADS-B trajectory data of manned aircraft, a short-term trajectory prediction model of manned aircraft in different flight stages was conducted. The original manned aircraft trajectory data were collected. Data processing was performed, including the removal of noise and outliers, and the reconstruction of the trajectory into equally spaced data to improve data quality. The model comprehensively takes the longitude, latitude, altitude, speed, and other information of the manned aircraft as inputs in the trajectory prediction model to learn the flight intention of the manned aircraft. The performance of the model was compared with the commonly used trajectory perdition models.

Second, considering the flexible and changeable characteristics of drones, and based on the historical trajectory data, a QRDGRU structure is proposed to predict the trajectory of the drone. The model integrates the bidirectional gated recurrent unit structure and the quartile regression structure, which can obtain the position distribution of the drone. The proposed model is compared with traditional GRU, SVR, QRLSTM and QRGRU models to testify its performance. The performance indicators confirm the superiority of the established model.

Third, given the difficulty in estimating the uncertain collision risk between manned aircraft and drones, a Monte-Carlo-based collision probability estimation model is proposed. Based on the predicted manned aircraft trajectory and drone position distribution, the estimated probability is calculated by the Monte Carlo simulation method. Time to the closest distance at the encountering point, and the collision probability between the drone and the manned aircraft at the closest point are calculated.

This paper predicts the conflict risk between non-cooperative drones and manned aircraft in the take-off and landing phases. The model structure can be directly applied by airport operators or air transport managers to identify the potential risk between manned aircraft and drones ahead of time. The intention of drones can be judged to detect dangerous behaviors. Furthermore, according to the estimated time to the closest distance at the encountering point, and the collision probability between manned aircraft and drones, different warning strategies can be implemented based on the predicted conflict levels. However, due to the limitation of acquired data, the following aspects can be further improved.

First, for trajectory prediction of manned aircraft, only ADS-B data are currently considered, and there is no support for flight plans, high-altitude weather, and other data. The performance of the model can be further improved as long as these parameters can be obtained and input into the model.

Second, the predicted trajectory of drones is only based on the historic ADS-B data, while the aircraft performance is not incorporated. As a matter of fact, the weather, especially wind, and the maneuverability of drones may influence the predicted position distribution. Future studies should also be conducted with regard to the intention of drones.

Third, in the estimation of the collision probability between the drone and manned aircraft, since the collision between the drone and manned aircraft cannot be tested in actual situations, this paper translates the trajectory to generate an intersection. The behaviors of

drones and manned aircraft may change for risk abatement in real conditions. At the same time, this paper focuses on estimating the probability of collision between a single manned aircraft and a single drone. In actual scenarios, the manned aircraft may also conflict with the drone fleet during the take-off and landing stages. Research can also be conducted on the collision risk between aircraft and multiple drones.

Author Contributions: R.Z., Z.Y. and J.C. contributed to the manuscript equally. All authors have read and agreed to the published version of the manuscript.

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

References

1. Ruiz, S.; Leones, J.L.; Ranieri, A. A novel performance framework and methodology to analyze the impact of 4D trajectory based operations in the future air traffic management system. *J. Adv. Transport.* **2018**, *2018*, 1601295. [CrossRef]
2. Zeng, W.; Chu, X.; Xu, Z.; Liu, Y.; Quan, Z. Aircraft 4D Trajectory Prediction in Civil Aviation: A Review. *Aerospace* **2022**, *9*, 91. [CrossRef]
3. Dalmau, R.; Pérez-Batlle, M.; Prats, X. Real-Time Identification of Guidance Modes in Aircraft Descents Using Surveillace Data. In Proceedings of the 2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC), London, UK, 23–27 September 2018.
4. Lymperopoulos, I.; Lygeros, J. Sequential Monte Carlo methods for multi-aircraft trajectory prediction in air traffic management. *Int. J. Adapt. Control Signal Process.* **2010**, *24*, 830–849. [CrossRef]
5. Ayhan, S.; Samet, H. Aircraft Trajectory Prediction Made Easy with Predictive Analytics. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; p. 27.
6. Lin, Y.; Zhang, J.-W.; Liu, H. An algorithm for trajectory prediction of flight plan based on relative motion between positions. *Front. Inf. Technol. Electron. Eng.* **2018**, *19*, 905–916. [CrossRef]
7. Lin, Y.; Yang, B.; Zhang, J.; Liu, H. Approach for 4-d trajectory management based on HMM and trajectory similarity. *J. Mar. Sci. Technol.* **2019**, *27*, 246–256.
8. Benavides, J.V.; Kaneshige, J.; Sharma, S.; Panda, R.; Steglinski, M. Implementation of a Trajectory Prediction Function for Trajectory Based Operations. In Proceedings of the AIAA Atmospheric Flight Mechanics Conference, National Harbor, MD, USA, 13–17 January 2014.
9. Sun, J.; Ellerbroek, J.; Hoekstra, J.M. WRAP: An open-source kinematic aircraft performance model. *Transp. Res. Part C Emerg. Technol.* **2019**, *98*, 118–138. [CrossRef]
10. Schuster, W.; Porretta, M.; Ochieng, W. High-accuracy four-dimensional trajectory prediction for civil aircraft. *Aeronaut. J.* **2012**, *116*, 45–66. [CrossRef]
11. Lemon, K.; Steck, J.; Hinson, B.; Rokhsaz, K.; Ngyen, N. Application of a Six Degree of Freedom Adaptive Controller to a General Aviation Aircraft. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Portland, OR, USA, 8–11 August 2011.
12. Kwon, H.; Kim, Y.; Park, K.W.; Yoon, H.; Choi, D. Advanced ensemble adversarial example on unknown deep neural network classifiers. *IEICE Trans. Inf. Syst.* **2018**, *101*, 2485–2500. [CrossRef]
13. Kwon, H. Detecting backdoor attacks via class difference in deep neural networks. *IEEE Access* **2020**, *8*, 191049–191056. [CrossRef]
14. Gallego, C.E.V.; Comendador, V.F.G.; Nieto, F.J.S.; Imaz, G.O.; Valdés, R.M.A. Analysis of air traffic control operational impact on aircraft vertical profiles supported by machine learning. *Transp. Res. Part C Emerg. Technol.* **2018**, *95*, 883–903. [CrossRef]
15. Shi, Z.; Xu, M.; Pan, Q. 4-D flight trajectory prediction with constrained LSTM network. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 7242–7255. [CrossRef]
16. Wu, Z.-J.; Tian, S.; Ma, L. A 4D trajectory prediction model based on the BP neural network. *J. Intell. Syst.* **2019**, *29*, 1545–1557. [CrossRef]
17. Gallego, C.E.V.; Comendador, V.F.G.; Carmona, M.A.A.; Valdés, R.M.A.; Nieto, F.J.S.; Martínez, M.G. A machine learning approach to air traffic interdependency modelling and its application to trajectory prediction. *Transp. Res. Part C Emerg. Technol.* **2019**, *107*, 356–386. [CrossRef]
18. Ghazbi, S.N.; Aghli, Y.; Alimohammadi, M.; Akbari, A.A. Quadrotors Unmanned Aerial Vehicles: A review. *Int. J. Smart Sens. Intell. Syst.* **2016**, *9*, 309–333. [CrossRef]
19. Netanel, R.B.; Nassi, B.; Shamir, A.; Elovici, Y. Detecting spying drones. *IEEE Secur. Priv.* **2020**, *19*, 65–73. [CrossRef]
20. Nassi, B.; Bitton, R.; Masuoka, R.; Shabtai, A.; Elovici, Y. SoK: Security and privacy in the age of commercial drones. In Proceedings of the 2021 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 24–27 May 2021; pp. 1434–1451.
21. Liang, J.; Ahmad, B.I.; Jahangir, M.; Godsill, S. Detection of malicious intent in non-cooperative drone surveillance. In Proceedings of the 2021 Sensor Signal Processing for Defence Conference (SSPD), Edinburgh, UK, 14–15 September 2021; pp. 1–5.
22. Zhou, T.; Chen, M.; Wang, Y.; He, J.; Yang, C. Information entropy-based intention prediction of aerial targets under uncertain and incomplete information. *Entropy* **2020**, *22*, 279. [CrossRef]

23. Alexis, K.; Nikolakopoulos, G.; Tzes, A. On trajectory tracking model predictive control of an unmanned quadrotor helicopter subject to aerodynamic disturbances. *Asian J. Control.* **2014**, *16*, 209–224. [CrossRef]
24. Maeder, U.; Morari, M.; Baumgartner, T.I. Trajectory prediction for light aircraft. *J. Guid. Control. Dyn.* **2011**, *34*, 1112–1119. [CrossRef]
25. Renault, A. A model for assessing UAV system architectures. *Procedia Comput. Sci.* **2015**, *61*, 160–167. [CrossRef]
26. Yang, Z.; Tang, R.; Bao, J.; Lu, J.; Zhang, Z. A real-time trajectory prediction method of small-scale quadrotors based on GPS data and neural network. *Sensors* **2020**, *20*, 7061. [CrossRef]
27. Wang, C.H.; Tan, S.K.; Low, K.H. Collision risk management for non-cooperative UAS traffic in airport-restricted airspace with alert zones based on probabilistic conflict map. *Transp. Res. Part C Emerg. Technol.* **2019**, *109*, 19–39. [CrossRef]
28. Zhang, N.; Liu, H.; Ng, B.; Low, K. Collision probability between intruding drone and commercial aircraft in airport restricted area based on collision-course trajectory planning. *Transp. Res. Part C Emerg. Technol.* **2020**, *120*, 102736. [CrossRef]
29. Wang, C.H.; Tan, S.K.; Low, K.H. Three-dimensional (3D) Monte-Carlo modeling for UAS collision risk management in restricted airport airspace. *Aerosp. Sci. Technol.* **2020**, *105*, 105964. [CrossRef]
30. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
31. Awad, M.; Khanna, R. Support Vector Regression. In *Efficient Learning Machines*; Chapter 4; Apress: Berkeley, CA, USA, 2015; pp. 67–80.
32. Han, P.; Wang, W.; Shi, Q.; Yang, J. Real-time short-term trajectory prediction based on GRU neural network. In Proceedings of the 2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC), San Diego, CA, USA, 8–12 September 2019.
33. Sahawneh, L.R.; Mackie, J.; Spencer, J.; Beard, R.W.; Warnick, K.F. Airborne radar-based collision detection and risk estimation for small unmanned aircraft systems. *J. Aerosp. Inf. Syst.* **2015**, *12*, 756–766. [CrossRef]