IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 7, 2021, accepted February 17, 2021, date of publication February 22, 2021, date of current version March 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3060783

Long and Short Term Maneuver Trajectory Prediction of UCAV Based on Deep Learning

LEI XIE[1], ZHENGLEI WEI[2], DALI DING[1], ZHUORAN ZHANG[3], AND ANDI TANG[1]

[1]Institute of Aeronautics Engineering, Air Force Engineering University, Xi'an 710038, China
[2]China Aerodynamics Research and Development Center, Mianyang 621000, China
[3]95806 Troops, Beijing 100000, China

Corresponding author: Zhenglei Wei (310370487@qq.com)

ABSTRACT Autonomous air combat technology of unmanned combat air vehicles (UCAVs) is a hot issue that is currently being studied by various countries, and maneuvering trajectory prediction is an important part of autonomous air combat research. To address the difficulty of maintaining high prediction accuracy and short prediction time simultaneously in maneuvering trajectory prediction, this paper proposes a maneuvering trajectory prediction method that is based on a layered strategy, which combines long-term maneuvering unit prediction and short-term maneuvering trajectory prediction. In long-term maneuvering unit prediction, the complex trajectory is divided into 21 types of maneuvering units using the four characteristics of maneuvering trajectories, and a maneuvering unit library is established. On the basis of the deep echo state network(DeepESN), to capture multiscale prediction input parameters, autoencoder (AE) technology is incorporated. In addition, to increase the prediction accuracy, adaptive boosting (Ada) learning technology is utilized to build a strong predictor, and seven prediction networks are compared. The results demonstrate that the proposed method realizes the highest prediction accuracy. The single-step prediction time is about 0.002 s, which meets the time requirement. In short-term maneuvering trajectory prediction, the long and short-term memory (LSTM) network is analyzed, and the gaussian random walk strategy particle swarm optimization (GWSPSO) algorithm is used to update the internal weights and biases of the network to overcome the problems of ''gradient disappearance'' and ''gradient explosion'', and a data sharing method is proposed for overcoming the no directionality of optimization algorithms. Compared with four traditional networks, the results demonstrate the method that is proposed in this paper performs better. Compared with the sampling time of 0.3 s, the short-term prediction time of 0.05 s can also meet the requirements. Finally, a long- and short-term layered prediction method is used on a group of complex maneuvering trajectories. The results demonstrate that the prediction accuracy is significantly increased and the real-time requirements are satisfied.

INDEX TERMS Ada-AE-DeepESN, GWSPSO-LSTM, layered strategy, trajectory prediction.

I. INTRODUCTION

With the continuous development of artificial intelligence technology, the intelligence and autonomy of unmanned combat air vehicles (UCAVs), which are represented by the American ''loyal wingman'', have been significantly improved, but the existing degree of intelligence is far from meeting practical requirements [1]; therefore, the autonomous air combat technology of UCAVs is currently a hot issue that is being studied by various countries, and it has also been a persistent research topic for decades [2]. Maneuvering trajectory prediction is an important part of autonomous air combat research. It uses the maneuvering trajectory information of the past moments, excavates the laws of aircraft movement through prediction methods and predicts the trajectory of the future moment [3], [4]. There are high requirements on the prediction accuracy and prediction time consumption. The available trajectory prediction methods are divided into

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

three main categories: dynamic models, hybrid theory, and machine learning. All three methods have unique advantages, but none can satisfy the accuracy and time-consuming requirements simultaneously. All countries in the world have conducted in-depth research on these three methods.

The prediction methods that are based on dynamic models can be regarded as physical models, which predict the trajectory based on the force of the aircraft [5]. Reference [6] proposes a motion model parameter identification method based on the Kalman filter and extended Kalman filter (KF-EKF) joint algorithm for four-dimensional trajectory prediction. By identifying the aircraft speed and using the aircraft's latitude and longitude coordinates as the observational measurements, the aircraft's short-term trajectory is predicted, but the prediction accuracy is low at high speeds. Reference [7] proposes a nonlinear model prediction method that is based on the configuration method. Compared with the traditional linear model, the prediction accuracy is substantially improved, and the real-time performance can satisfy the requirements. However, the test is only performed under low-speed conditions, and the error will be magnified in high-speed flight trajectory prediction. Reference [8] designs an adaptive interactive multiple model (IMM) based on the aircraft motion model controlled by aerodynamic parameters for trajectory prediction, but this method greatly reduces the prediction accuracy when the target attitude changes drastically. Reference [9] uses aerodynamic parameters to design a set of maneuvering modes and realizes trajectory prediction through Monte Carlo sampling and Bayesian theory. Compared with traditional extrapolation theory, the accuracy is higher, but a more complete set of maneuvering modes is required, which is difficult to realize in practical scenarios. Reference [10] proposes a short-term trajectory prediction method that is based on a motion model and a long-term trajectory prediction method that is based on maneuver intentions and constructs a trajectory prediction method using an interactive multiple model algorithm. According to the above studies, the kinetic model prediction method realizes better real-time performance, but it is necessary to establish a more accurate kinetic model for the target, which is difficult to realize under practical conditions, thus, this approach leads to larger prediction errors.

Hybrid theory methods regard the trajectory prediction problem as a hybrid system estimation problem, and combines multiple methods with filtering theory or Markov theory to perform trajectory prediction. Reference [11] clusters according to the characteristics of the historical trajectory and trains a hidden Markov model for each cluster. However, in the flight trajectory, too many categories are divided according to characteristics, which renders the real-time prediction performance unsatisfactory. Reference [12] proposes a gray dynamic filtering method for trajectory prediction. Compared with the traditional Kalman filter and the original gray method, the prediction accuracy is substantially increased; however, it uses the minimum variance estimate to replace the actual value, introduces a differential equation, and cannot accurately estimate the parameters. Inaccurate parameters will reduce the prediction accuracy. Reference [13] proposes a trajectory prediction method that is based on a hidden Markov model (HMM), which adds adaptive parameters to the hidden Markov process, increases the prediction efficiency, and can adaptively adjust the trajectory prediction length, but the prediction accuracy must be enhanced. Reference [14] proposes an improved Gauss-Hermite particle filter (GHPF) method. It integrates the new observation number into the system state transition probability, uses the Gauss-Hermit filter to generate the importance density function, and uses GHPF to predict the aircraft trajectory, which requires many observations. The prediction accuracy is lower when the deviation of observations is large. Reference [15] designs a trajectory prediction framework under the spark platform that is based on the second-order hidden Markov model. Compared with the hidden Markov model and the kernel smooth variable-order Markov model, the robustness is higher, but the algorithm must be improved in terms of prediction accuracy. Reference [16] proposes a prediction method that uses the HMM to model the movement trend of the flight based the historical trajectory, and uses the Gaussian mixture model to predict the aircraft speed vector, but the prediction error is large in the case of high-speed maneuvers. The above method shows that the hybrid theory method has a wide range of applications, but the algorithm complexity is high, and the prediction accuracy and real-time prediction performance are not prominent.

Machine learning methods mainly use a deep learning network to mine historical trajectory information for prediction. Reference [17] proposes an aircraft 4D trajectory prediction model that combines a convolutional neural network (CNN) and a long short-term memory (LSTM) network. The prediction accuracy is higher than that of a single model. However, its main disadvantages are that it can only perform short-term predictions, the aircraft trajectory changes cannot be too drastic, and its application range is limited. Reference [18] proposes a sequence-to-sequence deep long short-term memory network (SS-DLSTM) for trajectory prediction, which increases the accuracy and robustness of the prediction, but it is only applied to the terminal airspace of aircraft navigation, and the trajectory of the terminal airspace is relatively smooth. In the case of complex trajectories, the prediction accuracy will be substantially reduced. Reference [19] improves the flight dynamics model, sets the yaw rate threshold, uses CNN convolution kernel sharing, reduces the storage space and training time, obtains overall information from local information via aggregation, and extracts hierarchical information from the input, but the prediction accuracy at altitude is low. Reference [20] uses the gated recurrent unit (GRU) to predict the flight trajectory, selects the optimal GRU network by comparing the number of network layers and the number of neurons, and compares it with the BP network, and the prediction error is reduced. The offline training process requires excessively high data accuracy and excessively large data volume. Reference [21] proposes an

end-to-end convolutional recurrent neural network to predict the trajectory, which uses an encoder to encode flight information into hidden state variables and uses a decoder to learn the temporal and spatial correlations of historical trajectories. However, when the trajectory changes drastically, the prediction error is large. The above studies show that machine learning methods are easy to implement and realize high accuracy when the data are sufficient, but the training process requires an excessive amount of data and the data dependence is too strong.

To overcome the disadvantages of machine learning methods and solve the problems of low prediction accuracy and long prediction time under complex maneuvering trajectories, we have made the following original contributions in this article:

1. This paper proposes a maneuvering trajectory prediction method that is based on a layered strategy that combines long-term maneuvering unit prediction with short-term maneuvering trajectory prediction.

2. It establishes a library of maneuver units and proposes a long-term mobile unit prediction method, namely, Adaptive boosting-Autoencoder-Deep Echo State Network (Ada-AE-DeepESN). Compared with seven prediction networks, the results demonstrate that the proposed method realizes the highest prediction accuracy and satisfies the real-time performance requirement.

3. It proposes a gaussian walking strategy particle swarm optimization algorithm-long short-term memory (GWSPSO-LSTM) network for online short-term maneuvering trajectory prediction. It overcomes the data dependence of network offline training and avoids the ''gradient explosion'' and ''gradient disappearance'' problems of network update. Compared with the four prediction networks, the results demonstrate that the proposed method realizes the highest prediction accuracy and that the time consumption satisfies the requirement.

4. The long- and short-term layered prediction method is used on a group of complex maneuvering trajectories, and the results demonstrate that the prediction accuracy is significantly increased and the real-time performance satisfies the requirements.

The remainder of this paper is organized as follows: Section II introduces a maneuvering trajectory prediction method that is based on the layered strategy. Section III establishes a long-term maneuver unit library, proposes a long-term mobile unit prediction method that is based on Ada-AE-DeepESN, and conducts simulation verification and analysis. Section IV proposes a short-term maneuvering trajectory prediction method that is based on the GWSPSO-LSTM and conducts simulation verification of the timeliness and accuracy. Section V presents the experimental simulation and result analysis of the combined prediction in the long and short terms. Section VI summarizes the conclusions and discusses future research plans.

II. MANEUVERING TRAJECTORY PREDICTION METHOD BASED ON LAYERED STRATEGY

The maneuvering trajectory prediction problem is divided into two layers: the long-term prediction problem is a maneuvering unit prediction problem with temporal significance. The short-term prediction problem is a multi-ime series prediction problem. The theoretical support of short-term prediction is that due to the limitation of UCAV maneuverability, there will be no sudden changes in the movement trend within a short-term, so it can be predicted based on its historical moment inertia, but when the maneuvering unit changes, it means that the movement trend has changed, at this moment, the short-term prediction is no longer applicable. Hence, this method selects the maneuver trajectory prediction model through long-term maneuver unit prediction. When the long-term maneuvering unit does not change, the trajectory change is relatively stable, and the short-term prediction model is adopted. When the long-term maneuvering unit changes, it shows that the trajectory is at the junction of the maneuvering unit, the change is drastic, and the short-term trajectory prediction model is no longer applicable; hence the long-term maneuvering unit is used for prediction. Information on the maneuvering trajectory prediction method that is based on the layered strategy is presented in Table 1.

TABLE 1. Information on the maneuvering trajectory prediction method that is based on the layered strategy.

Name	Maneuvering trajectory prediction based on the layered strategy
Objective	On the basis of maintaining real-time trajectory prediction, increasing the prediction accuracy
Significance	Enriching the research content of autonomous drone air combat
Method	The long-term maneuvering unit is used to predict whether the enemy's trajectory changes drastically; If so, the long-term maneuvering unit prediction method is used, otherwise, the short-term trajectory prediction method is used.
Trajectory acquisition method	Obtain accurate trajectory information of enemy aircraft based on drone airborne radar or satellite data link

According to the above analysis, long-term maneuvering unit prediction and short-term trajectory prediction are fused through a layered strategy. The process is illustrated in Figure 1, and the steps are as follows:

Step 1: Use trajectory features to classify and construct a long term mobile unit library;

Step 2: Construct the long-term prediction layer and the short-term prediction layer;

Step 3: Obtain track information of the historical moment, extract characteristic parameters of the historical track, and identify maneuvering units of the past moment;

Step 4: Use the current trajectory information and the current maneuvering unit to obtain the maneuvering unit at the future time through the long-term prediction layer;

Step 5: Determine whether the maneuvering unit at the previous moment is the same as the maneuvering unit at

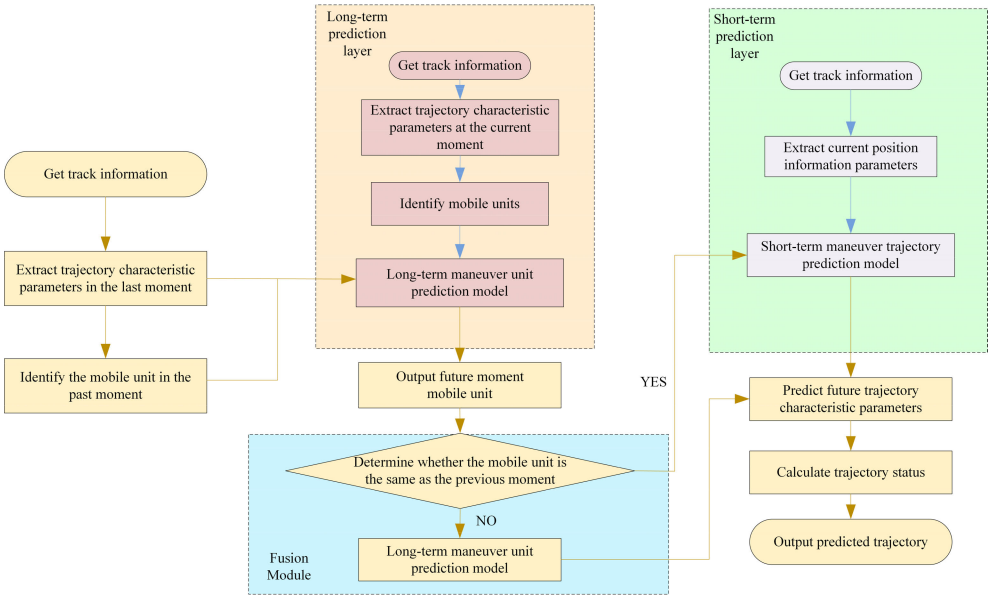


FIGURE 1. Flow chart of maneuver prediction based on layered strategy.

the future moment. If it is the same, the short term domain prediction model is adopted. Otherwise, the long term domain maneuvering unit prediction model is adopted;

Step 6: Calculate the trajectory state information and output the predicted trajectory of the target machine.

III. LONG TERM MANEUVERING UNIT PREDICTION

A. LONG TERM MANEUVER UNIT LIBRARY

Maneuvering trajectories can be divided into horizontal plane maneuvers, vertical plane maneuvers and space combined maneuvers [22]. To define the characteristics of the maneuvering trajectory, a three-degree-of-freedom model of UCAV kinematics is established, as illustrated in Figure 2.

$$\begin{cases} \dot{x}_t = v_t \cos\theta_t \cos\psi_t \\ \dot{y}_t = v_t \cos\theta_t \sin\psi_t \\ \dot{z}_t = v_t \sin\theta_t \end{cases} \tag{1}$$

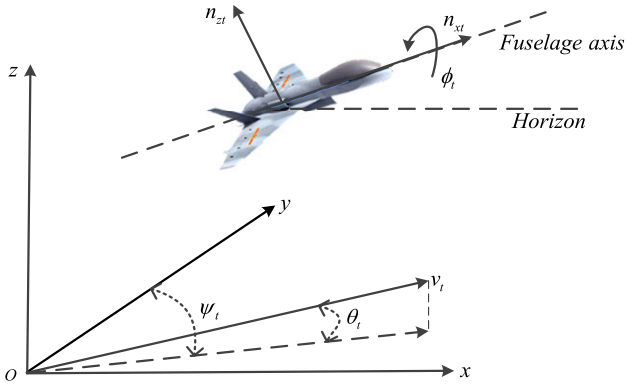


FIGURE 2. Schematic diagram of the UCAV three-degree-of-freedom model.

$$\begin{cases} \Delta v_t = g\,(n_{xt} - \sin\theta_t) \\ \Delta\theta_t = \dfrac{g}{v_t}\,(n_{zt}\cos\phi_t - \cos\theta_t) \\ \Delta\psi_t = \dfrac{g n_{zt}\sin\phi_t}{v_t\cos\theta_t} \end{cases} \tag{2}$$

where, (x_t, y_t, z_t) represents the position information in the UCAV inertial coordinate system; Δv_t, $\Delta\theta_t$ and $\Delta\psi_t$ represent the rates of change of the speed, track inclination angle and track deflection angle; g is the acceleration due to gravity; n_{xt} and n_{zt} are the tangential overload and normal overload, ϕ_t is the velocity of the roll angular; $[x_t, y_t, z_t, v_t, \theta_t, \psi_t]^T$ and $[n_{xt}, n_{zt}, \phi_t]^T$ are the state variables and control variables, respectively.

In the horizontal plane, according to the track deflection angle and the change rate of the track deflection angle, the maneuvering trajectory can be divided into level flight, right turn and left turn. If the change rate of the track deflection angle $\Delta\psi_t$ is positive, the aircraft is turning left; otherwise, the aircraft is turning right. If track deflection angle ψ_t reaches 90° or -90° and the track deflection angle change rate is always the same sign, the maneuvering trajectory is considered to be a complete left- or right-turn maneuver, and the track deflection angle returns to zero. In direct flight, the current track deflection angle remains unchanged, and its change rate is zero. The three maneuvering trajectory units in the horizontal plane are labelled as E01, E08 and E15 in Figure 3.

In the vertical plane, according to the track inclination angle and the change rate of the track inclination angle, the maneuvers can be divided into climbing and diving. Climbing maneuvers with increased height, such as half jack reversal maneuvers, can be composed of two track segments. First, the track inclination angle changes from 0° to

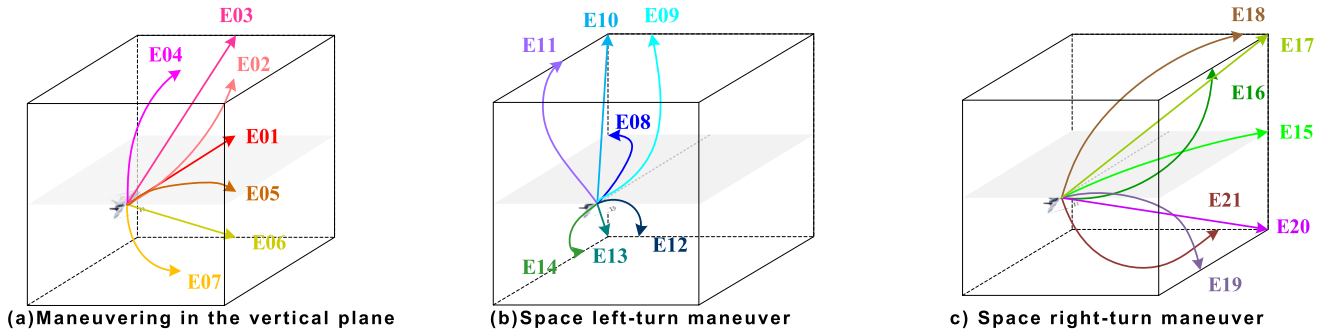


FIGURE 3. Schematic diagram of the maneuvering trajectory unit.

90°, the normal acceleration is upward, and the trajectory shape is a concave upward climb shape. Second, the track inclination angle changes from 90° to 0°, the normal overload is lower, and the flight trajectory is a convex upward climb shape. The difference between these two shapes is mainly caused by the difference in the roll angle of the aircraft by 180°, while the attitude and overload of the target aircraft are not measured by the UCAV sensor. Therefore, the change rate of the flight track inclination angle is used to distinguish the trajectory shape in the vertical plane. At the same time, the inclination angle is used to distinguish between climbing and diving, and it returns to zero when the inclination angle reaches 90° or -90°. The maneuvering trajectory unit in the vertical plane is divided into 6 modes (concave up flight, fly diagonally upward, convex up flight, convex down flight, fly diagonally down, and concave down flight), which are labelled in Figure 3(a) as E02-E07.

In space, a space maneuver can be decomposed into a maneuver in the horizontal plane and a maneuver in the vertical plane. According to the classification in the horizontal plane, the space maneuvering trajectory unit is divided into two categories: the space left turning maneuver and the space right turning maneuver; According to the classification in the vertical plane, the space left turning maneuver is divided into turn left and concave up flight, turn left and fly straight up, turn left and convex up flight, turn left and convex down flight, turn left and fly straight down, turn left and concave down flight. Space right turn is similar to space left turn. The space left turn maneuvering trajectory unit is labelled as E09-E14 in Figure 3(b). Similar to the left turn, the space right turn maneuvering trajectory unit is divided into turn right and concave up flight, turn right and fly straight up, turn right and convex up flight, turn right and convex down flight, turn right and fly straight down, turn right and concave down flight (except for a horizontal right turn), which are labelled as E16-E21 in Figure 3(c).

Based on the above analysis, the track inclination angle θ, track inclination angle change rate $\Delta\theta$, track deflection angle ψ and track deflection angle change rate $\Delta\psi$ are selected as the identification characteristic parameters of the maneuvering unit, and the maneuvering unit table is established, which is presented as Table 2 .

B. ADAPTIVE BOOSTING-AUTOENCODER-DEEP ECHO STATE NETWORK

To solve the problem of long-term maneuvering unit prediction, an Ada-AE-DeepESN is proposed.

The echo state network (ESN) is a recurrent neural network [23], [24] that uses the "reserve pool" method to construct the hidden layer of the network, and it is usually used for time series prediction [25]. The ESN is composed mainly of an input layer, storage layer, and output layer. The structure is shown in Figure 4. The connection weight $\boldsymbol{W}_{in}^{r\times n}$ from the input layer to the storage layer does not need to be trained, and will not change after random initialization. The input of the storage layer comes from the input layer and the output of the previous state of the storage layer. The state feedback weight $\boldsymbol{W}^{r\times r}$ is initialized randomly and does not require training. The weights $\boldsymbol{W}_{out}^{m\times r}$ from the storage layer to the output layer need to be trained. Typically, ridge regression is used for training [26], which is expressed as follows:

$$\boldsymbol{W}_{out} = \boldsymbol{Y}_{long}\boldsymbol{H}^T\left(\boldsymbol{H}\boldsymbol{H}^T + \lambda_r\boldsymbol{I}\right)^{-1} \tag{3}$$

$\boldsymbol{Y}_{long}$ is the output of the training samples, $\boldsymbol{H}$ is the state of the storage layer, and λ_r represents the regularization coefficient. The storage layer status $\boldsymbol{H}$ is expressed as follows:

$$\boldsymbol{H}(t) = \tanh\left(\boldsymbol{W}_{in}\boldsymbol{X}_{long}(t) + \boldsymbol{W}\boldsymbol{H}(t-1)\right) \tag{4}$$

$\boldsymbol{X}_{long}(t)$ represents the input variable at time t, and tanh represents the hyperbolic tangent activation function.

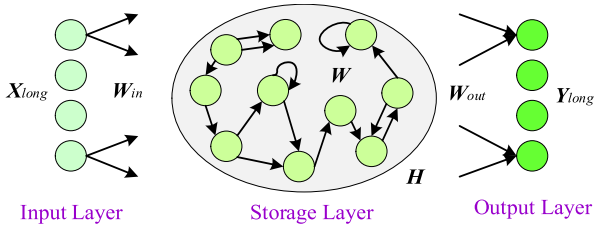


FIGURE 4. Echo state neural network.

The ESN algorithm process includes two stages of weight parameter initialization and training. The ESN network contains a relatively large number of neurons, the connection weights between the neurons in the storage layer are randomly generated, and the connections are sparse. The hyperparameters of ESN strongly influence the prediction

TABLE 2. Maneuvering trajectory unit table.

Number	Maneuvering trajectory unit	Track inclination angle θ	Change rate of the track inclination angle $\Delta\theta$	Track deflection angle ψ	Change rate of the track deflection angle $\Delta\psi$
E_{01}	Level flight	≈ 0	≈ 0	Constant	≈ 0
E_{02}	Concave up flight	(0,90)	>0	Constant	≈ 0
E_{03}	Fly diagonally upward	>0	≈ 0	Constant	≈ 0
E_{04}	Convex up flight	(90,0)	<0	Constant	≈ 0
E_{05}	Concave down flight	(0,–90)	<0	Constant	≈ 0
E_{06}	Fly diagonally down	<0	≈ 0	Constant	≈ 0
E_{07}	Convex down flight	(–90,0)	>0	Constant	≈ 0
E_{08}	Turn left and fly horizontally	≈ 0	≈ 0	/	>0
E_{09}	Turn left and concave up flight	(0,90)	>0	/	>0
E_{10}	Turn left and fly straight up	>0	≈ 0	/	>0
E_{11}	Turn left and convex up flight	(90,0)	<0	/	>0
E_{12}	Turn left and convex down flight	(0,–90)	<0	/	>0
E_{13}	Turn left and fly straight down	<0	≈ 0	/	>0
E_{14}	Turn left and concave down flight	(–90,0)	>0	/	>0
E_{15}	Turn right and fly horizontally	≈ 0	≈ 0	/	<0
E_{16}	Turn right and concave up flight	(0,90)	>0	/	<0
E_{17}	Turn right and fly straight up	>0	≈ 0	/	<0
E_{18}	Turn right and convex up flight	(90,0)	<0	/	<0
E_{19}	Turn right and convex down flight	(0,–90)	<0	/	<0
E_{20}	Turn right and fly straight down	<0	≈ 0	/	<0
E_{21}	Turn right and concave down flight	(–90,0)	>0	/	<0

(a,b) represents the process of change from a to b.
/ denotes no requirement.

performance, hence it is highly important to adjust the hyperparameters. The ESN hyperparameters include the storage layer size N_r, spectral radius SR, input scale factor IS, and storage layer sparsity SD.

To capture the multiscale predictive input parameters, in combination with the autoencoder, AE-DeepESN is proposed. AE-DeepESN is based on the ESN network and increases the number of storage layers through the mapping of the autoencoder [27]. In the AE-DeepESN network structure, the echo state of the previous storage layer is reduced to low dimensionality through autoencoder, and input into the next storage layer, so as to loop to the last layer, sort all the echo states, and output the final result through the output layer. The AE-DeepESN network structure is illustrated in Figure 5. The mathematical model of the deep echo state network is expressed as follows:

$$\boldsymbol{H}_{in}^{(l)}(t) = \boldsymbol{W}_{in}^{(l)}\boldsymbol{X}_{in}^{(l)}(t) + \boldsymbol{W}^{(l)}\boldsymbol{H}^{(l)}(t-1) \tag{5}$$

$$\boldsymbol{X}_{in}^{(l)}(t) = \begin{cases} \boldsymbol{X}_{long}(t), l = 1 \\ f_{enc}\left(\boldsymbol{W}_{enc}^{(l-1)}\boldsymbol{H}^{(l-1)}(t)\right), l > 1 \end{cases} \tag{6}$$

$$\boldsymbol{H}^{(l)}(t) = \left(1 - SD^{(l)}\right)\boldsymbol{H}^{(l)}(t-1) + SD^{(l)}\tanh\left(\boldsymbol{H}_{in}^{(l)}(t)\right) \tag{7}$$

$$\boldsymbol{Y}_{long}(t) = g\left(\boldsymbol{W}_{out}\boldsymbol{H}(t)\right) \tag{8}$$

$\boldsymbol{H}_{in}^{(l)}(t)$ represents the weighted input data of the storage pool of the $l-th$ layer at time t, $\boldsymbol{W}_{in}^{(l)}$ is the connection weight of the input to the storage pool of the $l-th$ layer, $\boldsymbol{X}_{in}^{(l)}(t)$ represents the input of the $l-th$ layer at time t,

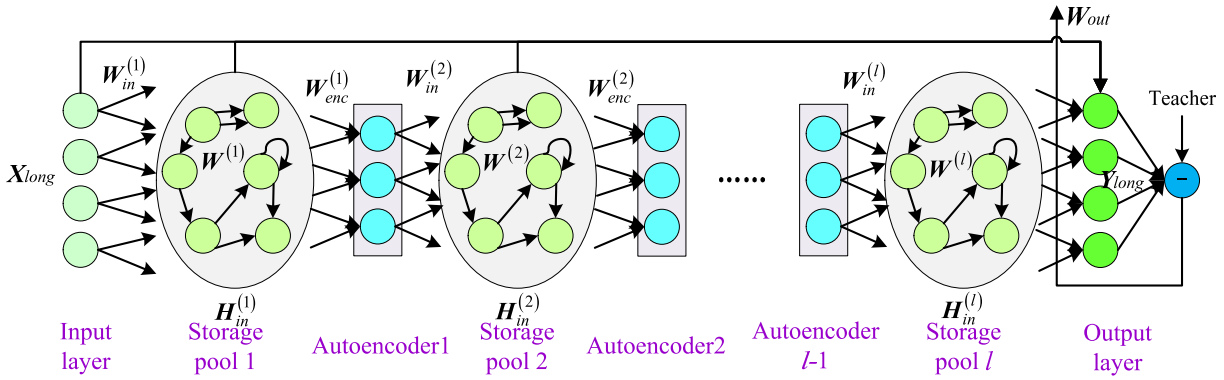


FIGURE 5. Deep echo state network that is based on an autoencoder.

$\boldsymbol{W}^{(l)}$ represents the state feedback of the $l-th$ storage pool weight, $\boldsymbol{H}^{(l)}(t-1)$ represents the storage pool state of the $l-th$ layer at time $t-1$, $\boldsymbol{W}_{enc}^{(l-1)}$ represents the autoencoder projection weight of the $(l-1)-th$ layer at time t, $f_{enc}(\cdot)$ represents the activation function of the autoencoder, $\boldsymbol{X}_{long}(t)$ represents the input variable at time t, $\boldsymbol{H}^{(l)}(t)$ represents the state value of the storage pool of the first layer at time t, and $SD^{(l)}$ is the sparsity degree of the storage pool of the $l-th$ layer. $\boldsymbol{H}(t)$ is the vector that is formed by the states of all storage pools, which is expressed as $\left[\boldsymbol{H}^{(1)}(t), \boldsymbol{H}^{(2)}(t), \cdots, \boldsymbol{H}^{(l)}(t)\right]$; $g(\cdot)$ is the activation function of the output layer; and $\boldsymbol{Y}_{long}(t)$ is the output result at time t.

The training process of the AE-DeepESN neural network includes initializing the network, obtaining storage state values, and training output weights.

To improve the prediction accuracy, use adaptive boosting learning technology (Ada) [28]–[31] to build an external framework, use the AE-DeepESN as a weak predictor, and build an Ada-AE-DeepESN. The structure is shown in Figure 6. The weak predictor is a sub-prediction system in the Ada-AE-DeepESN.

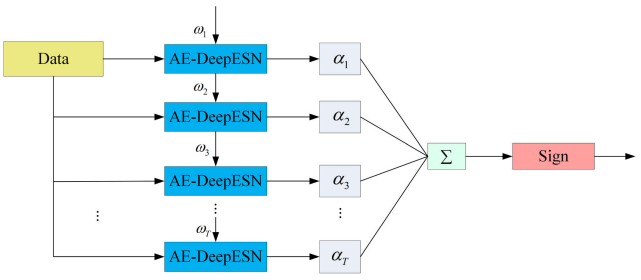


FIGURE 6. structure diagram of Ada-AE-DeepESN.

Consider a training sample set $\boldsymbol{P}_{long} = \{(\boldsymbol{X}_{long,i}, \boldsymbol{Y}_{long,i})\}_{i=1}^{m_s}$, where $\boldsymbol{X}_{long,i}$ is the i-th long-term sequence sample input, $\boldsymbol{Y}_{long,i}$ is the i-th supervision signal, and m_s is the number of samples. Suppose the base prediction algorithm is AE-DeepESN, which is denoted as $f_{AE-DeepESN}(\cdot)$, and the number of weak predictors is T. The algorithm is described as follows:

Step 1: Initialize the weight distribution of time series samples $D_1 = (\omega_{1,1}, \omega_{1,2}, \cdots, \omega_{1,m})$, and calculate the weight of each sample as follows:

$$\omega_{1,i} = \frac{1}{m_s}, i = 1, 2, \cdots, m_s \tag{9}$$

Step 2: For the number of iterations $t = 1, 2, \cdots, T$, use the training samples of the trainer with the current distribution D_t to train the weak predictor $h_t = f_{AE-DeepESN}(\boldsymbol{P}_{long}, D_t)$;

Step 3: Calculate the prediction error rate of the weak predictor h_t on the training sample set:

$$\varepsilon_t = \sum_{i=1}^{m_s} \omega_{t,i} e_{t,i} \tag{10}$$

$$e_{t,i} = \frac{\left(\boldsymbol{Y}_{long,i} - h_t\left(\boldsymbol{X}_{long,i}\right)\right)^2}{E_t^2} \tag{11}$$

$$E_t = \max\left|\boldsymbol{Y}_{long,i} - h_t\left(\boldsymbol{X}_{long,i}\right)\right|, i = 1, 2, \cdots, m_s \tag{12}$$

$e_{t,i}$ represents the relative square error of the $i-th$ sample on the $t-th$ weak predictor, and E_t is the maximum error of the $t-th$ samples.

Step 4: Calculate the weight coefficient a_t of the weak predictor h_t:

$$a_t = \frac{\varepsilon_t}{1-\varepsilon_t} \tag{13}$$

Step 5: Update the sample distribution D_{t+1} of the training set until the number of iterations reaches the maximum value.

$$\omega_{t+1,i} = \frac{\omega_{t,i} a_t^{1-e_{t,i}}}{\sum_{i=1}^{m_s} \omega_{t,i} a_t^{1-e_{t,i}}} \tag{14}$$

Step 6: Linearly combine T weak predictors to obtain the final strong predictor:

$$f_{AE-DeepESN}\left(\boldsymbol{X}_{long}\right) = \sum_{t=1}^{T}\left(\ln\frac{1}{a_t}\right) G\left(\boldsymbol{X}_{long}\right) \tag{15}$$

where $G\left(\boldsymbol{X}_{long}\right)$ is the median of all $a_t h_t\left(\boldsymbol{X}_{long}\right)$.

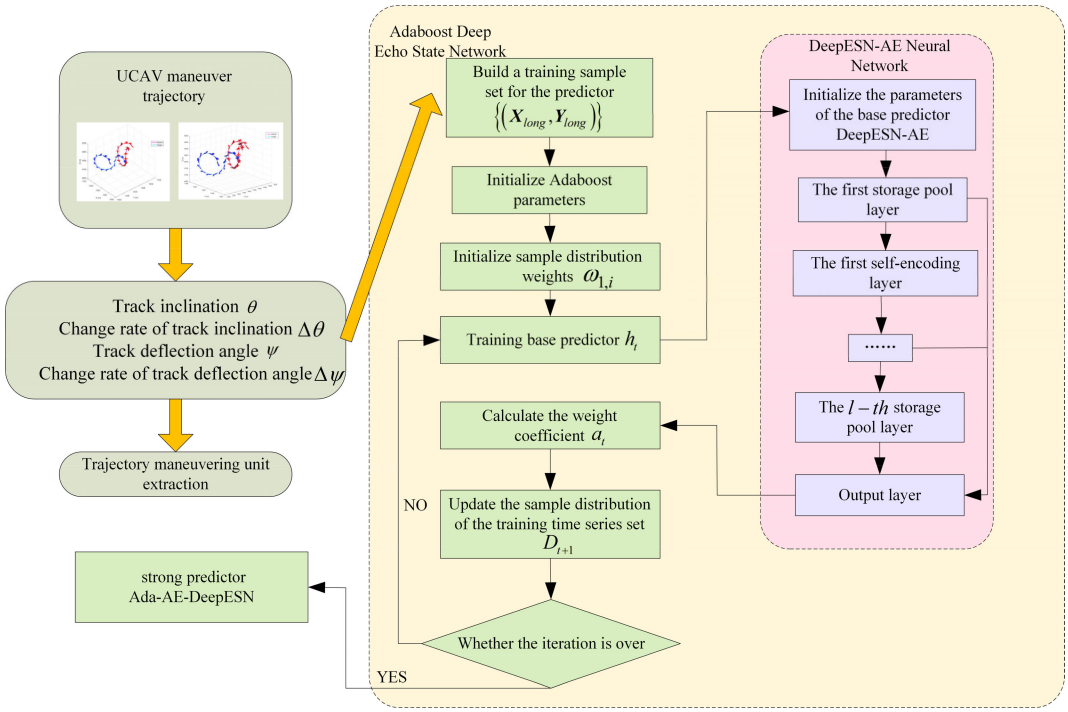


FIGURE 7. Flow chart of long-term maneuvering unit prediction.

C. LONG TERM MANEUVERING UNIT PREDICTION METHOD

Combining the Ada and the AE-DeepESN, a maneuvering unit prediction method based on the Ada-AE-DeepESN is proposed. The current track inclination angle θ, change rate of the track inclination angle $\Delta\theta$, track deflection angle ψ, change rate of the track deflection angle $\Delta\psi$ and historical maneuvering trajectory unit number L_e are the input of the prediction model, the maneuvering trajectory unit number at the future time is the output. The maneuvering unit prediction model that is based on Ada-AE-DeepESN is illustrated in Figure 7. The steps are as follows:

Step 1: Obtain training samples. Use the UCAV kinematics model to obtain trajectory data, and extract the trajectory track inclination angle θ, change rate of the track inclination angle $\Delta\theta$, track deflection angle ψ, change rate of the track deflection angle $\Delta\psi$ and historical maneuvering trajectory unit number L_e as the input $\boldsymbol{X}_{long}$ of the training sample set. The maneuvering trajectory unit number $L_{e-future}$ at the future time is the output $\boldsymbol{Y}_{long}$ of the training sample set. Construct the training sample set $\{(\boldsymbol{X}_{long}, \boldsymbol{Y}_{long})\}$.

Step 2: Initialize the Ada parameters. Initialize the weak predictor AE-DeepESN, and set the number of weak predictors;

Step 3: Initialize the distribution weight D_1 of the training sample;

Step 4: Train the weak predictor AE-DeepESN. First initialize the weak predictor parameters, input them into the storage pool in turn, and input them into the storage pool again after dimensionality reduction by the autoencoder. Use all the storage pool states and supervision signals to train the connection weights of the output layer;

Step 5: Calculate the weight coefficient a_t, and update the sample distribution D_{t+1}. Determine whether the iteration is over. If it is over, output the Ada-AE-DeepESN prediction result in weight, otherwise, continue to train the weak predictor;

D. LONG-TERM MANEUVERING UNIT PREDICTION SIMULATION

To evaluate the performance of the Ada-AE-DeepESN prediction method in solving the long-term maneuvering unit prediction problem, first, the UCAV flight trajectory data are used to obtain the input sequence step size, prediction step size, number of weak predictors, storage layers, storage pool size N_r, spectral radius SR, input scale factor IS, and storage pool sparsity SD of Ada-AE-DeepESN. Second, the training set is used to train the long-term maneuver meta-prediction layer Ada-AE-DeepESN network. Finally, the test set is used to evaluate the performance of Ada-AE-DeepESN. To evaluate the performance of the proposed integrated network, Ada-AE-DeepESN is compared with Support Vector Machines (SVR), Back Propagation (BP), and Group Method of Data Handing (GMDH). In addition, to evaluate the impact of the improved part of Ada-AE-DeepESN, the proposed algorithm is compared with ESN [32], DeepESN [33], AE-DeepESN [34] and Ada-DeepESN. The parameter settings of the comparison algorithms are presented in Table 3. The trajectory data mainly originate from the UCAV kinematics model. The number of training samples is 2075, and the

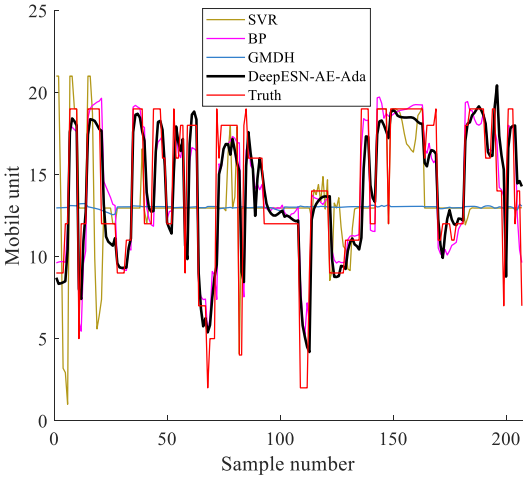
TABLE 3. Parameter settings of the tactical maneuver element prediction method.

Algorithm	Parameter Settings
SVR	C=1000, radial basis kernel function, σ=0.5, tol=1e-3
BP	Number of hidden layer nodes 20, epochs=100, lr=0.1
GMDH	The maximum number of layers is 3, The maximum numbers of neurons in each layer is 5
ESN	The Nr, SR, IS, and SD settings are the same as for Ada-AE-DeepESN
DeepESN	The parameter settings are the same as for the Ada-AE-DeepESN part
DeepESN-AE	The parameter settings are the same as for the Ada-AE-DeepESN part
DeepESN-Ada	The parameter settings are the same as for the Ada-AE-DeepESN part
DeepESN-AE-Ada	The parameters are obtained via experimental analysis

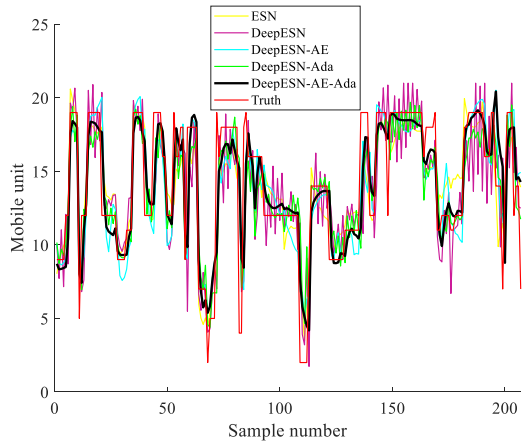
number of test samples is 207. The experimental simulation environment is Windows 10, the CPU is 2.80 GHz, the memory is 8 GB, and the programming language is MATLAB. Each simulation experiment is run 20 times, and the prediction results of the 20 runs are recorded.

According to the experimental analysis of Ada-AE-DeepESN parameters in the appendix B, the maneuvering unit prediction model selects an input step size of $n = 5$ and an output step size of $m = 3$. The Ada-AE-DeepESN network selects 10 weak predictors, the storage pool is 2 layers, the storage pool size N_r is 150, the hidden layer dimension of the autoencoder is 50, the spectral radius SR is 0.9, and the input scale factor IS is 0.8. The sparseness of the storage pool SD is 0.3. The test set of the simulation experiment is extracted by simulating 30 types of close-range air combat using the UCAV dynamic model. The mean absolute error (MAE), root mean square error (RMSE), mean absolute percentage error (MAPE) and prediction time (test time) are used as evaluation indices of the predictive performance.

Figure 8 presents the test sample prediction results of the comparison algorithm. According to Figure 8(a), compared with other algorithms, the overall trend of the prediction results of the Ada-AE-DeepESN is similar to the overall trend of the real maneuvering unit. As shown in Figure 8(b), the prediction results of ESN and its improved algorithm are similar to the real maneuvering unit trajectory, but the prediction results of Ada-AE-DeepESN are the closest. Table 4 presents the prediction performance results of the comparison algorithm. According to Table 4, the MAE, RMSE, and MAPE of Ada-AE-DeepESN are the best, and the average test time is 0.4047 seconds. Based on the variances of MAE, RMSE, MAPE and time, SVR performs the best; the prediction time of ESN is the shortest. As there are more weak predictors, the prediction time of Ada-AE-DeepESN is increased, but it realizes real-time prediction (0.4047÷207≈0.002 s). Moreover, according to the three error indicators, Ada-AE-DeepESN realizes the smallest prediction error.



(a)The prediction results of SVR, BP, GMDH and the proposed algorithm



(b) The prediction results of ESN and its improved algorithm

FIGURE 8. Test sample prediction results of the maneuvering trajectory unit prediction algorithm.

IV. SHORT-TERM MANEUVERING TRAJECTORY PREDICTION

Short-term maneuvering trajectory prediction uses historical maneuvering trajectory position information to mine maneuver flight rules, which are used to predict the maneuvering trajectory position in the future. The GWSPSO-LSTM network is proposed for predicting maneuvering trajectories.

A. LSTM NETWORK

The network is divided mainly into feedforward networks and feedback networks [35]. The feedforward neural network only depends on the current input for each output, and cannot consider the interactions between inputs at various times. Therefore, when dealing with timing issues, feedback networks are typically used. A recurrent neural network (RNN) is a type of feedback network that specializes in processing

TABLE 4. Comparison of algorithm performance indicators.

Prediction algorithm	MAE	RMSE	MAPE	Time/s
	Mean/Variance	Mean/Variance	Mean/Variance	Mean/Variance
SVR	3.4191/**1.78e-15**	5.0010/**0.00e+00**	0.0033/**8.67e-19**	0.0740/**2.19e-03**
BP	1.9667/7.41e-02	2.9584/8.17e-02	0.0016/4.14e-05	0.0335/2.32e-02
GMDH	3.8244/1.50e-01	4.5678/2.11e-01	0.0019/7.84e-05	0.0068/7.66e-03
ESN	2.5752/1.45e-01	3.4483/1.82e-01	0.0018/5.44e-05	**0.0080**/4.07e-03
DeepESN	2.9984/1.19e-01	3.9545/1.14e-01	0.0028/1.92e-04	0.0233/5.91e-03
AE-DeepESN	2.2715/1.02-01	3.0954/1.33e-01	0.0017/4.95e-05	0.0406/1.02e-02
Ada-DeepESN	2.3062/4.35e-02	3.2393/3.50e-02	0.0018/6.01e-05	0.1729/2.50e-02
Ada-AE-DeepESN	**1.8962**/3.21e-02	**2.7646**/4.20e-02	**0.0015**/1.40e-05	0.4047/8.90e-02

time series data samples. The RNN network includes not only the traditional input to the hidden layer to the output process but also the hidden layer to the hidden layer transfer process. The structure is illustrated in Figure 9. The output of the hidden layer not only depends on the input but is also related to the output of the hidden layer at the previous moment. This is why the RNN network can associate the inputs at different times in the same sequence. Theoretically, if the data have long-interval dependence or the dependence range changes over time, the RNN can better solve the problem, but in the actual process, due to the long flow of data information, the amount of calculation will explode, and it is difficult to learn the long-term dependency [36]. To overcome this, the long short-term memory network, which is a variant of RNN, has been proposed, and this network is also adopted in this paper.

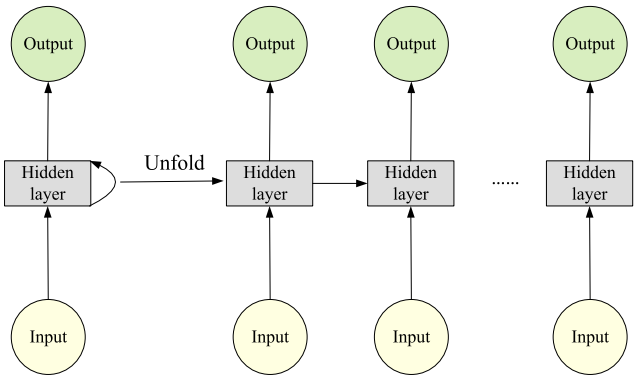


FIGURE 9. RNN structure diagram.

Compared with RNN, the long-and short-term memory network adds three gating systems: an input gate, a forget gate and an output gate [37]–[39]. The specific network unit is illustrated in Figure 10. C_{t-1} is the state of the previous unit, h_{t-1} is the output of the previous unit, X_t is the input of the network unit at the current moment, C_t and h_t are the state and output of the current network unit.

The LSTM network mainly transfers information through the status of the unit. From the calculation process, C_{t-1} is multiplied by the output of the forget gate and summed with the output of the input gate. The essence is to update the information at the previous moment, and then merge it with the information at the current moment, in this way to remember the information for a long time. To simplify the data, a forget gate is added, h_{t-1} and X_t are spliced into a single vector, and the data are normalized to between 0-1 through the sigmoid activation function, where 1 is ''completely reserved'' and 0 is ''completely discarded''. This can effectively filter the data and avoid useless calculations. The calculation formula for the forget gate is:

$$F_t = \sigma(W_f \cdot [h_{t-1}, X_t] + b_f) \tag{16}$$

(σ represents the sigmoid activation function)

The input gate determines the input information of the current unit, and the tanh function represents the current information. At the same time, the sigmoid function is used to determine which information is useful and which is useless, and its output is multiplied by the tanh function output, and input into the current unit state. The formula is:

$$i_t = \sigma(W_i \cdot [h_{t-1}, X_t] + b_i) \cdot \tanh(W_c \cdot [h_{t-1}, X_t] + b_c) \tag{17}$$

(tanh represents the tanh function)

The output gate determines the output of the current unit. The current unit state C_t is represented by the tanh function, h_{t-1} and X_t are activated by the sigmoid function. The two quantities are multiplied, and the result is the current unit output. The calculation formula is:

$$O_t = \sigma(W_o \cdot [h_{t-1}, X_t] + b_o) \tag{18}$$

$$h_t = O_t \cdot \tanh(C_t) \tag{19}$$

The LSTM network solves the problem of long-term dependence through the control of three gates; therefore, this network is used for trajectory prediction. In the [33], it has been posited that independent prediction of three-dimensional coordinates is more accurate than the overall prediction; hence, the coordinates on the X-, Y-, and Z-axes are separately used as the input of the LSTM network.

The disadvantage of offline training LSTM network is that it is very dependent on data, it requires a lot of data under different working conditions for training. Once the amount of data does not meet the demand, or there is a large amount of data in a single case, it will cause very large prediction deviations. This is a big problem that will inevitably occur in offline training. To overcome this problem, an online training method is proposed, which uses small sample real-time

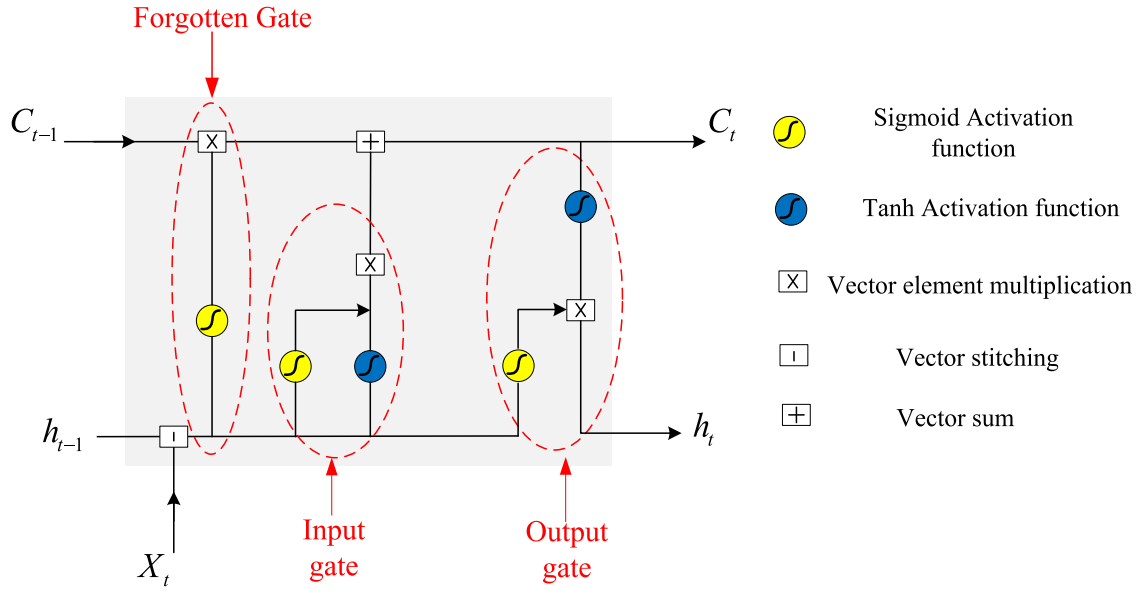
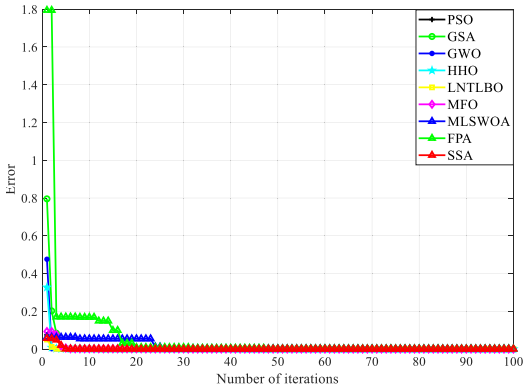
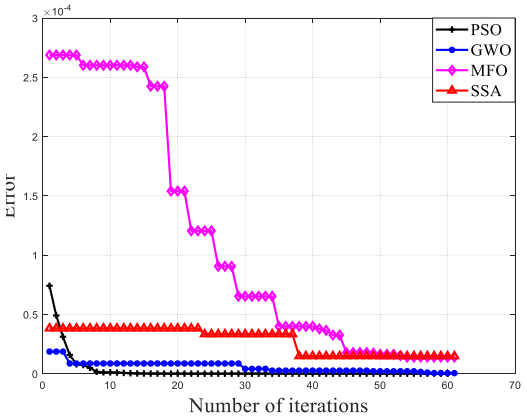


FIGURE 10. LSTM network unit.



(a) Error falling curve



(b) End of error falling

FIGURE 11. Replacement of BPTT with a heuristic algorithm for updating the internal weights and biases of the LSTM network.

data for online prediction. When using the three-degree-of-freedom model to simulate the trajectory, the data are sampled at an interval of 0.3 s and are sampled nine times as a group. A 5 × 5 sliding module matrix is constructed from the first nine samples, which is used to predict the tenth sample data.

Sliding module input matrix:

$$E_{\text{input}} = \begin{bmatrix} e_1 & e_2 & e_3 & e_4 & e_5 \\ e_2 & e_3 & e_4 & e_5 & e_6 \\ e_3 & e_4 & e_5 & e_6 & e_7 \\ e_4 & e_5 & e_6 & e_7 & e_8 \\ e_5 & e_6 & e_7 & e_8 & e_9 \end{bmatrix} \tag{20}$$

Sliding output correction module matrix:

$$E_{\text{output}} = \begin{bmatrix} e_6 & e_7 & e_8 & e_9 \end{bmatrix}^{\tau} \tag{21}$$

In the online test of the LSTM network, the first four rows of the input matrix establish real-time corrections, and the internal weights and biases are adjusted timely. The number of network input nodes is 5, and the number of output nodes is 1, it is a double hidden layer structure, and the number of hidden layer nodes is 5.

B. GWSPSO OPTIMIZATION OF THE INTERNAL WEIGHTS AND BIASES

The LSTM network does not completely solve the problems of "gradient disappearance" and "gradient explosion" of the network through the three gating systems. The main reason is that the internal weight update adopts the traditional Back Propagation Through Time (BPTT) algorithm [40], which continuously chains differentiation and multiplication. To avoid this phenomenon, a heuristic optimization algorithm is used to transform the weight update into an optimization

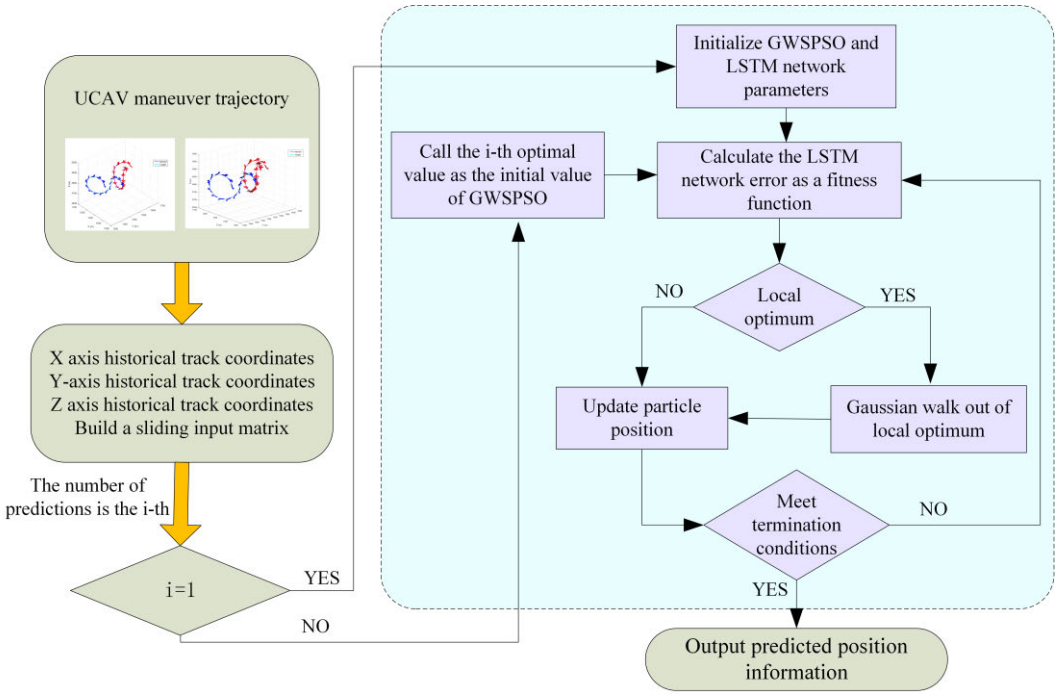


FIGURE 12. Flow chart of short-term maneuvering trajectory prediction.

problem. A variety of heuristic algorithms are used to optimize the LSTM network, and the most suitable algorithm is selected. It can be seen from Figure 11.a that the initial convergence speed of the Particle Swarm Optimization (PSO) algorithm is faster. From Figure 11.b, it can be seen that the PSO algorithm has the lowest error and the highest accuracy. Therefore, the PSO algorithm is most suitable for adjusting the internal weights and biases of the LSTM network.

When a nondirectional optimization algorithm is used, the update time will inevitably increase compared with that of the gradient descent method. Hence, data sharing is adopted. In the prediction process after the second time, the initial population setting of the PSO algorithm becomes the optimal population that was calculated at the last time, which significantly reduces the search time at the beginning of the algorithm. In the iterative optimization process of the algorithm, the LSTM network error is used as the fitness function. If there is no change in three consecutive iterations, the algorithm search is determined to be stagnant, and the Gaussian random walk strategy (GWS) [41] is used to break the local limitation. The GWS model is as follows:

$$X(\mathrm{t}+1) = G\mathrm{aussian}(X(t), \sigma_1) \tag{22}$$

$$\sigma_1 = \cos(\pi \cdot \mathrm{t}/(2 \cdot t_{\max})) \cdot (X_t - X_r^*(t)) \tag{23}$$

$X_r^*(t)$ is the dominant individual in the population, the cosine function $\cos(\pi \cdot \mathrm{t}/(2 \cdot t_{\max}))$ is introduced, and the Gaussian walk step length is adjusted, which is inversely proportional to the number of iterations. The disturbance is large in the early stage and reduced in the later stage, which balances the global search and local search capabilities.

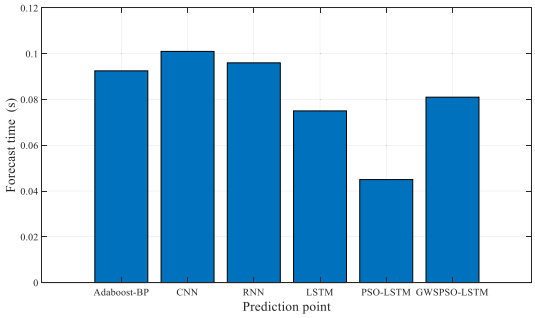


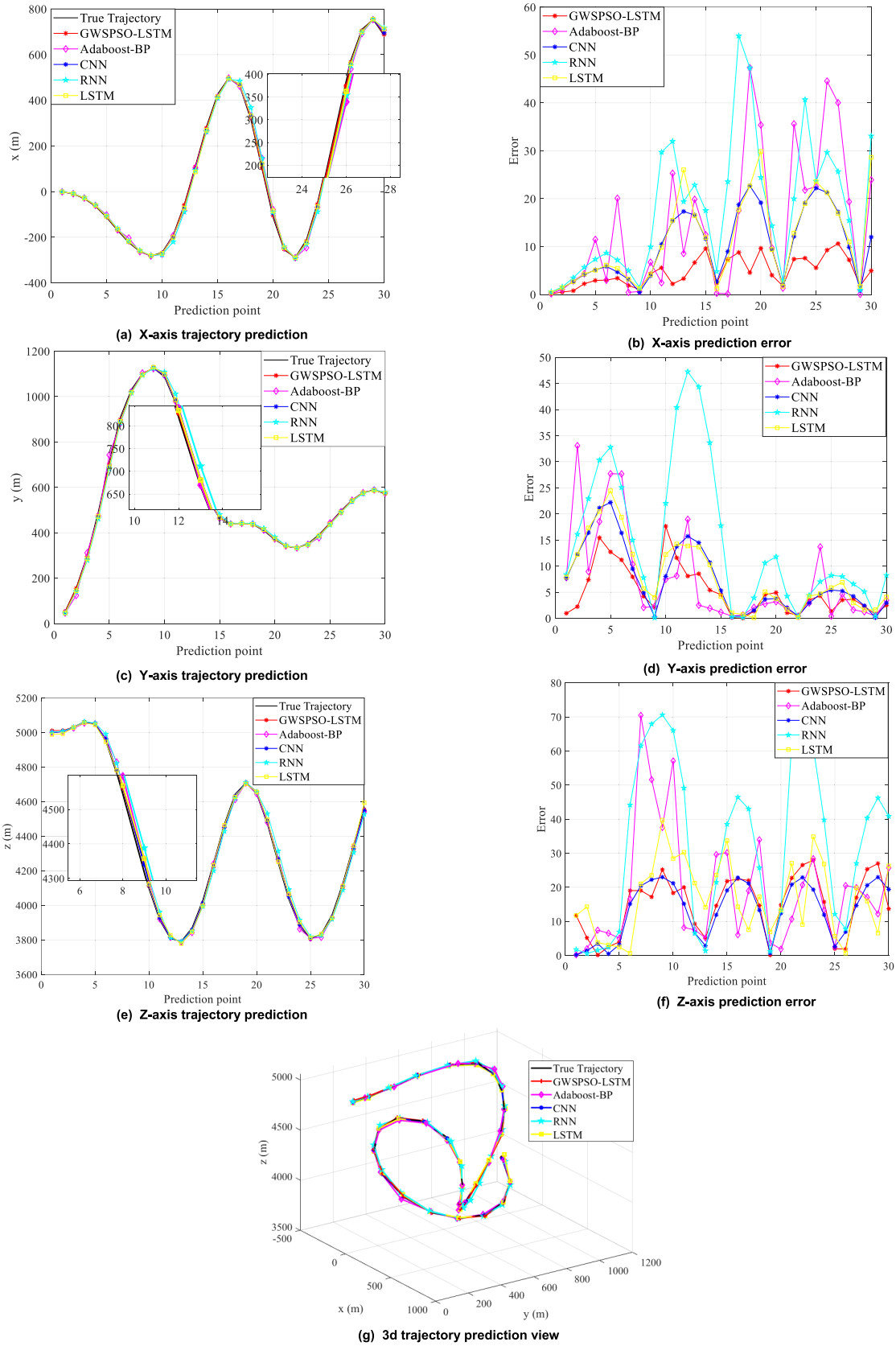
FIGURE 13. Single-step time comparison chart.

C. SHORT-TERM MANEUVERING TRAJECTORY PREDICTION METHOD

Combining the GWSPSO algorithm and LSTM network, a short-term trajectory prediction method that is based on the GWSPSO-LSTM network is proposed. The prediction model takes the historical trajectory position as input and outputs the future trajectory position. The prediction model is illustrated in Figure 12. The steps are as follows:

Step 1: Obtain historical flight trajectories. Decompose historical flight trajectory into the x-axis, y-axis and z-axis to construct a sliding input matrix;

Step 2: Determine the number of predictions. When testing for the first time, initialize the GWSPSO and LSTM network parameters. When the number of predictions exceeds 1, the optimal value of the last optimization is called the initial parameter of GWSPSO;



(a) X-axis trajectory prediction

(b) X-axis prediction error

(c) Y-axis trajectory prediction

(d) Y-axis prediction error

(e) Z-axis trajectory prediction

(f) Z-axis prediction error

(g) 3d trajectory prediction view

FIGURE 14. Trajectory prediction and error.

Step 3: Use the LSTM network error as the fitness function to calculate the fitness value;

Step 4: Determine whether the network is falling into a local optimum. When falling into a local optimum, the Gaussian walking strategy is used to attempt to escape. If not falling into a local optimum, the traditional method is used to update the particle position and velocity;

Step 5: Determine whether the termination conditions are satisfied. Output the predicted position information if they are satisfied, and return to step 3 if they are not satisfied.

D. SHORT-TERM TRAJECTORY PREDICTION SIMULATION

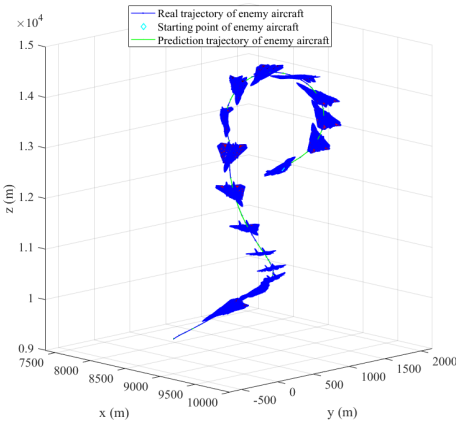
To evaluate the performance of the proposed short-term prediction model, a relatively complex flight trajectory is selected, and the prediction results are compared with those traditional Adaboost-BP, CNN, RNN and LSTM networks in terms of real-time performance and accuracy.

Due to the diversity of maneuvering trajectory changes, offline prediction by all networks requires substantial amount of data for training, and accuracy cannot be guaranteed. Therefore, the online prediction method of the sliding input module is adopted, and a single-step prediction time comparison is performed on this basis.

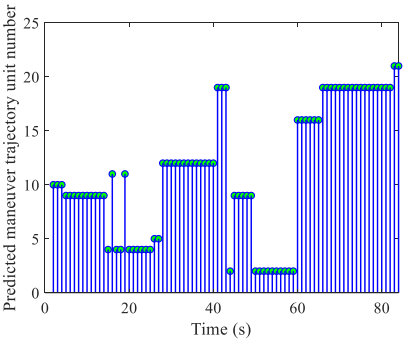
According to Figure 13, the single-step time differences of all network online predictions are not very large, and they are all below 0.12 s. The time consumption of GWSPSO-LSTM is approximately 0.05 s, which can provide timely predictive information relative to the sampling interval of 0.3 s, with higher real-time performance.

A three-degree-of-freedom model is used to randomly generate a group of maneuvering trajectories, sample 300 times, and make a prediction for every 10 groups, a total of 30 cycles are predicted. To increase the prediction accuracy, three-dimensional coordinate independent prediction is used to compare errors with those of other prediction methods. According to Figure 14, the errors of the GWSPSO-LSTM network on the three axes are the smallest; hence, its trajectory is the most suitable prediction of the actual trajectory. The prediction results of the X-axis and Y-axis are more accurate: the average X-axis error is 6.7 m, and the average Y-axis error is 7.2 m. Because the maneuvering on the Z-axis changes drastically, resulting in multiple changes of flight trends, which also produces a large error, the maximum error is 28.5 m, and the average Z-axis error is 14.9 m. Compared with the traditional deep learning network offline training and prediction, the accuracy has been significantly increased.

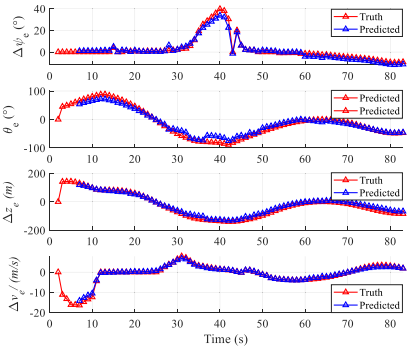
The 3D trajectory view clearly shows that the trajectory predicted by the GWSPSO-LSTM network does not have many sudden changes, but the trajectories that are drawn by the traditional CNN, RNN and LSTM network prediction points have many sudden changes, which are not in line with the true trajectory. The predicted points of the CNN and the LSTM are closer to the true trajectory, but the trajectory that is predicted by the GWSPSO-LSTM network



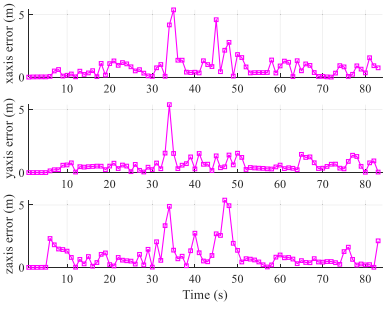
(a) UCAV maneuvering real trajectory and predicted trajectory



(b) Maneuvering trajectory unit prediction results



(C) Prediction results of maneuvering trajectory characteristics



(D) Trajectory prediction position error

FIGURE 15. Prediction results of the Ada-AE-DeepESN and GWSPSO-LSTM fusion model.

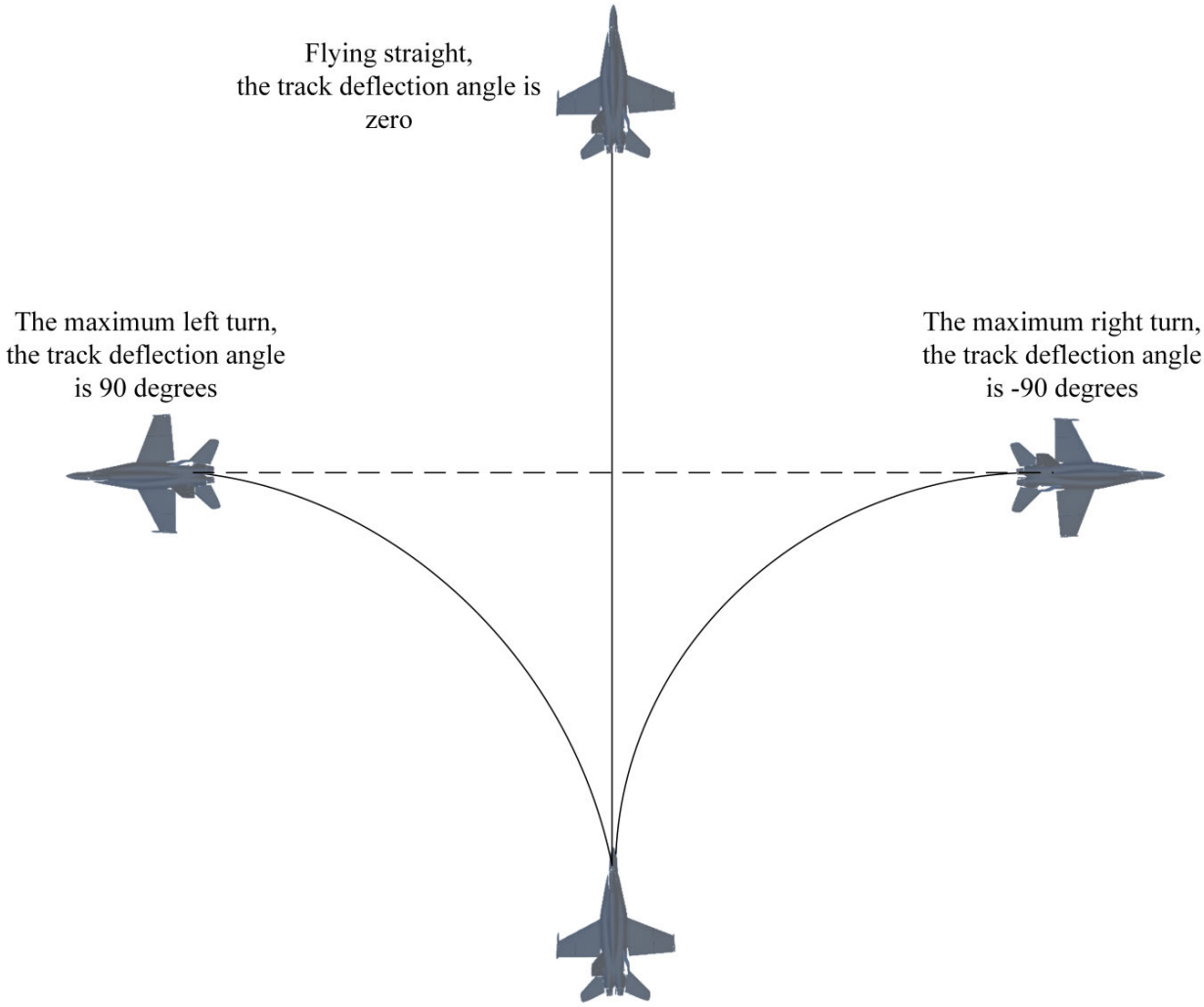


FIGURE 16. UCAV maneuvering view in the horizontal plane.

is closest to the true trajectory. In most cases, the predicted and true trajectories coincide, and only a small subset of the prediction points have deviations, which are small, hence, the GWSPSO-LSTM network outperforms the other networks in terms of accuracy.

V. LONG-AND SHORT-TERM COMBINED PREDICTION SIMULATION

The hybrid prediction method that is based on the combination of long-and short-term domains is a new method for UCAV maneuvering trajectory prediction. To further evaluate the performance and robustness of the long and short-term maneuvering trajectory prediction method, the Ada-AE-DeepESN long term prediction model is combined with the GWSPSO-LSTM short-term prediction model, and the UCAV kinematics model is used to randomly generate a set of maneuvering trajectories for prediction.

When the long-term predicted maneuver unit is different from the previous one, the long-term predicted angle and the short-term predicted speed are used to calculate the position information. The calculation formula is as follows:

$$x_{future} = x_{now} + v_{future} \cos\theta_e \cos\psi_e \tag{24}$$

$$y_{future} = y_{now} + v_{future} \cos\theta_e \sin\psi_e \tag{25}$$

$$z_{future} = z_{now} + v_{future} \sin\theta_e \tag{26}$$

Among them, x_{future}, y_{future}, z_{future} are predicted coordinates; x_{now}, y_{now}, z_{now} are current coordinates; v_{future}, θ_e, ψ_e are predicted speed, track inclination angle and track deflection angle.

Figure 15(b) shows the maneuvering trajectory unit that is predicted by Ada-AE-DeepESN, which is consistent with Figure 15(a). Figure 15(c) shows that the error of the track deflection angle $\Delta\psi_e$ does not exceed 5°, the error of the track inclination angle θ_e does not exceed 10°, the error of the altitude change rate Δz_e does not exceed 1 m, and the error of the speed change rate Δv_e does not exceed 1 m/s. Figure 15(d) shows that the average error of the X-axis does not exceed 4 m, the average error of the Y-axis does not exceed 4 m, and the average error of the Z-axis does not

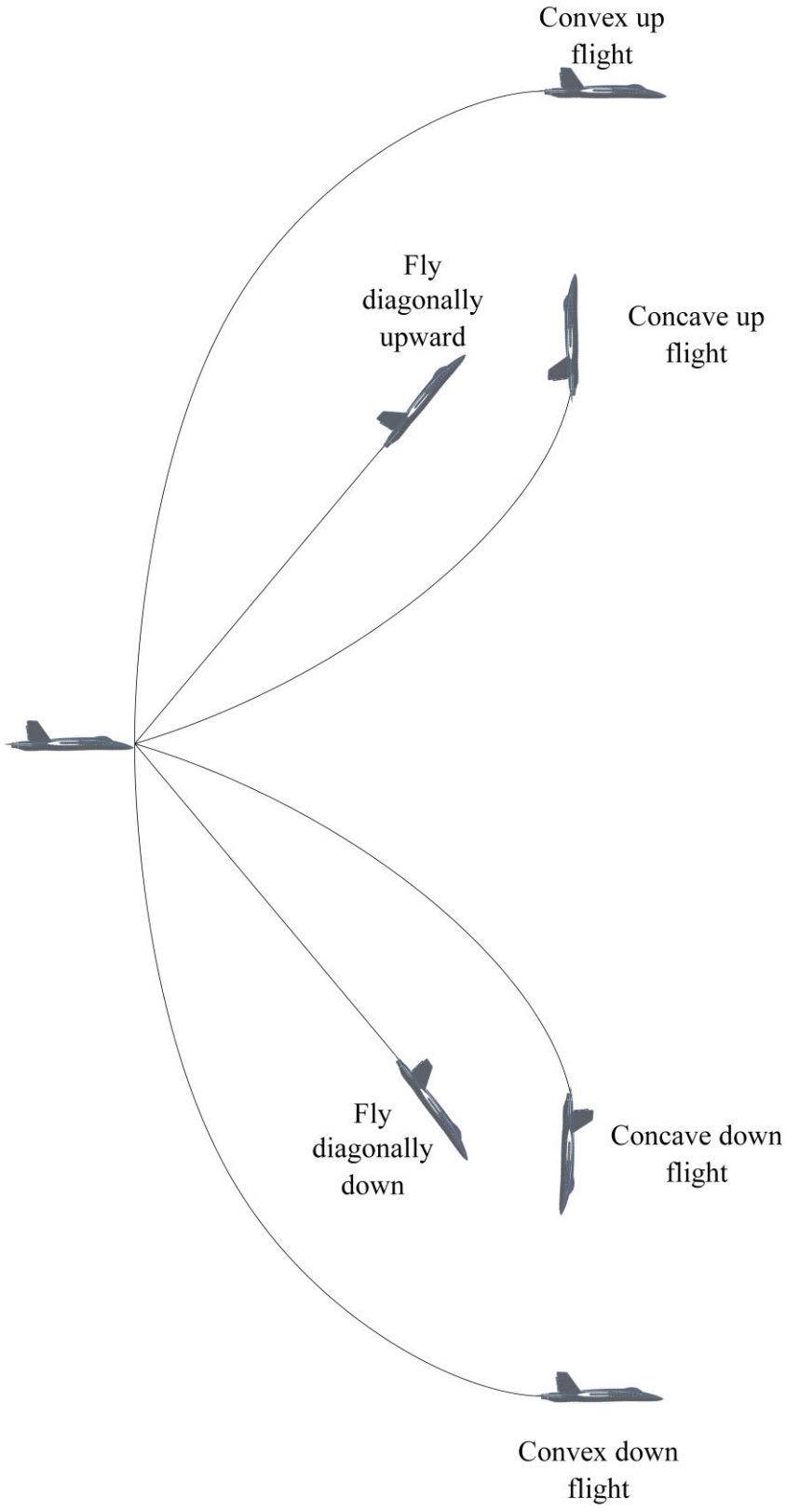
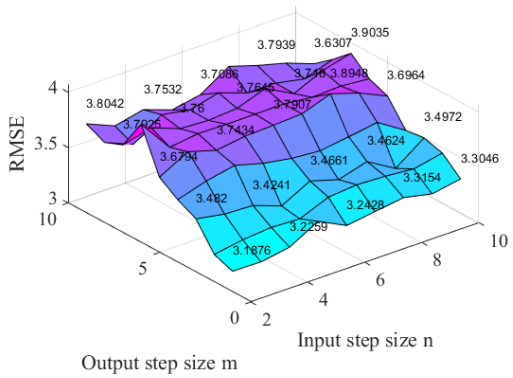
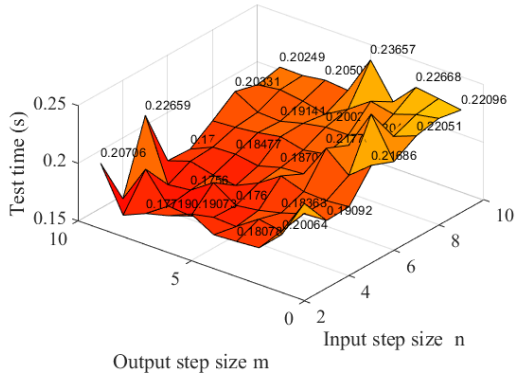


FIGURE 17. UCAV movement view in the vertical plane.

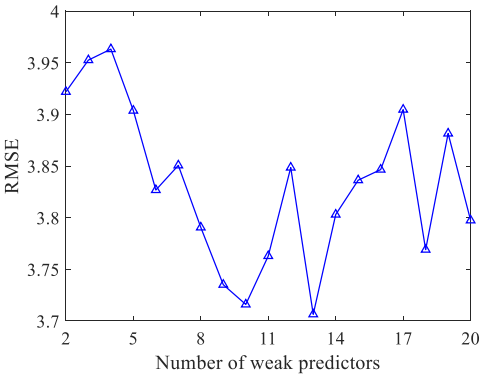


(a) The influences of various step size combinations on the prediction error

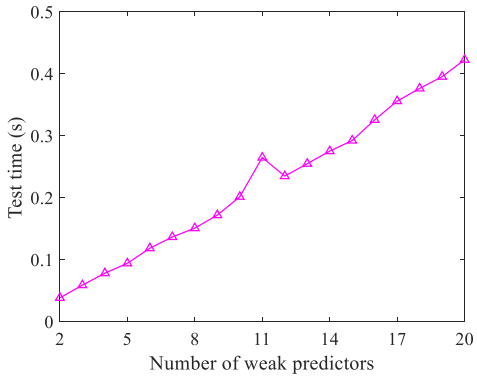


(b) The influences of various step size combinations on the prediction time

FIGURE 18. Influences of various input step sizes and output step sizes on the prediction performance of Ada-AE-DeepESN.



(a) The influence of the number of weak predictors on the prediction error



(b) The influence of the number of weak predictors on the prediction time.

FIGURE 19. Influence of the number of weak predictors on the prediction performance of Ada-AE-DeepESN.

exceed 5 m. Compared with the short-term domain maneuvering trajectory prediction method alone, the position accuracy is significantly increased.

VI. CONCLUSION AND FUTURE DEVELOPMENT

For UCAV maneuvering trajectory prediction, a long-and short-term maneuvering trajectory prediction method that is based on a layered strategy is proposed. This method divides the trajectory prediction problem into two parts: maneuvering unit prediction and maneuvering trajectory prediction. Through multiple sets of simulation experiments, the following conclusions are obtained:

1) The trajectory is divided into 21 basic maneuvering units according to the track inclination angle, the change rate of the track inclination angle, the track deflection angle and the change rate of the track deflection angle, which can fully demonstrate the change characteristics of the maneuvering trajectory.
2) The Ada-AE-DeepESN outperforms SVR, BP, GMDH, ESN, DeepESN, AE-DeepESN and Ada-DeepESN in terms of the prediction accuracy of the long-term maneuvering unit, and the prediction time consumption can also satisfy the requirements.
3) The short-term maneuvering trajectory online prediction method that is based on the GWSPSO-LSTM network can avoid the data dependence of the offline training of the traditional deep learning method, and the prediction accuracy is higher than the Adaboost-BP, CNN, RNN and LSTM networks. Compared with the 0.3 s sampling interval, the 0.05 s prediction time can also satisfy the requirements.
4) Compared with short-term maneuvering trajectory prediction, the hybrid prediction method of long-and short-term combination realizes substantial improvement in the accuracy of prediction.

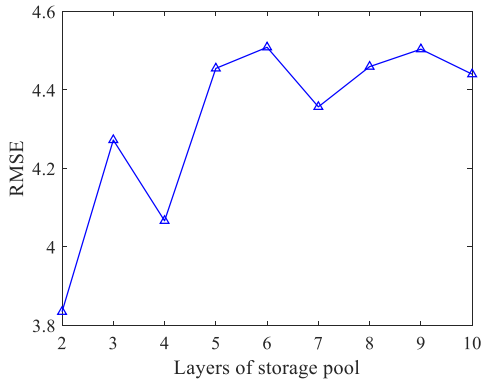
Future studies should establish a more complete maneuvering trajectory unit library, increase the prediction accuracy, and combine trajectory prediction with maneuver decision-making.

APPENDIX A

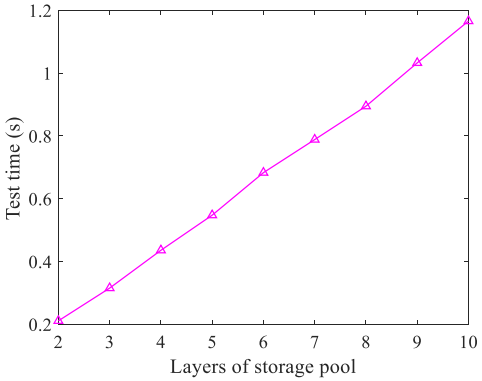
LONG TERM MANEUVER UNIT

Fig.16

Fig.17

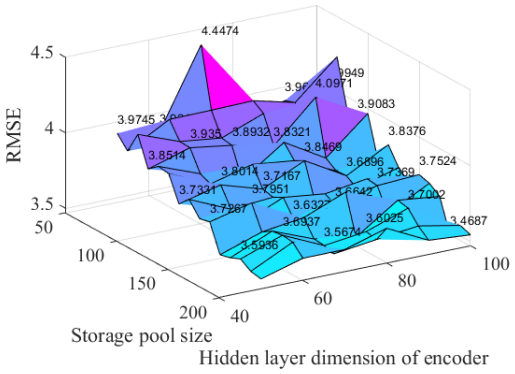


(a) The influence of Nr on the algorithm prediction error

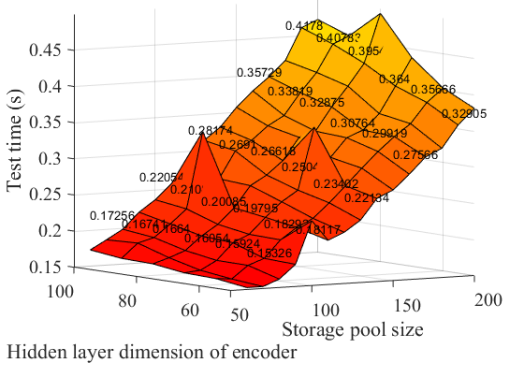


(b) The influence of Nr on the algorithm prediction time

FIGURE 20. Impact of storage pool layers on the prediction performance of Ada-AE-DeepESN.



(a) The influences of various combinations on the algorithm prediction error



(b) The influences of various combinations on the algorithm prediction time

FIGURE 21. Impact of various storage pool sizes and hidden layer dimensions of the autoencoder on the prediction performance of Ada-AE-DeepESN

APPENDIX B

EXPERIMENTAL ANALYSIS OF Ada-AE-DeepESN PARAMETERS

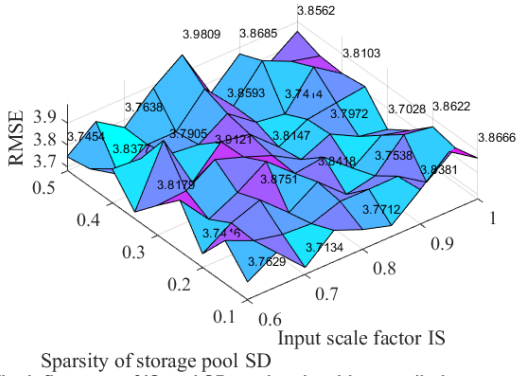
First, analyze the input step size n and output step size m of the long-term prediction model. The effects of various combinations of the input step length and output step length on Ada-AE-DeepESN are presented in Figure 18. According to Figure 18, as the input step size increases, the root mean square error (RMSE) of the prediction of the Ada-AE-DeepESN method does not change substantially, and the prediction time increases. Moreover, as the output step size increases, the RMSE also increases, and the prediction time does not change substantially. In summary, the input step size is 2∼5, and the output step size is 1∼3. To increase the diversity of mobile unit predictions, select an input step size of $n = 5$ and an output step size of $m = 3$.

Figure 19 presents the impact of the number of weak predictors on the prediction performance of Ada-AE-DeepESN. Figure 19(a) shows that as the number of weak predictors increases, the RMSE decreases, and when it reaches 10, the RMSE changes unstably. Figure 19(b) shows that as the number of weak predictors increases, the prediction time increases. In summary, the number of weak predictors of the integrated network should be selected as 10.

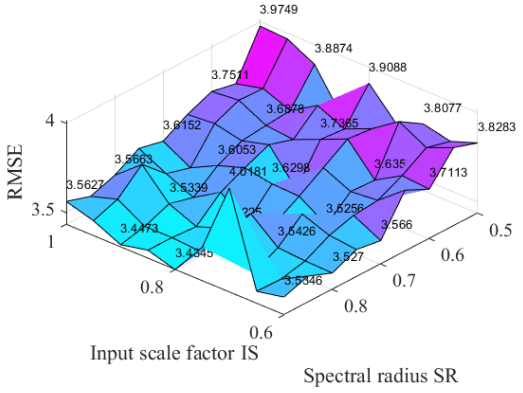
Figure 20 shows the effect of the numbers of storage pool layers on the prediction performance. Figure 20 shows that as the number of storage pool layers increases, the predicted RMSE value of Ada-AE-DeepESN increases, and the prediction time increases. Therefore, the number of ESN storage pool tiers of Ada-AE-DeepESN is selected as 2.

Figure 21 shows the impacts of various storage pool sizes and hidden layer dimensions of the autoencoder on the prediction performance of Ada-AE-DeepESN. According to from Figure 21(a), as the size of the storage pool N_r increases, the prediction error RMSE decreases, and the prediction time increases significantly. As shown in Figure 4(b), as the hidden layer dimension of the autoencoder increases, the prediction error RMSE does not change substantially, but it shows a slight increasing trend, and the prediction time also shows a slight increasing trend. Therefore, after comprehensive considerations, the size of the storage pool N_r is selected as 150, and the hidden layer dimension of the autoencoder is selected as 50.

Figure 22 shows the influences of the spectral radius SR, input scale factor IS, and storage pool sparsity SD on the prediction performance of Ada-AE-DeepESN. These



(a) The influences of IS and SD on the algorithm prediction error



(b) The influences of *IS* and *SR* on the algorithm prediction error

FIGURE 22. Impact of *SR*, *SD* and *IS* on the prediction performance of Ada-AE-DeepESN.

three parameters do not increase the size of the structure of the ESN; hence, the impacts of these three parameters on the prediction time are small. Figure 22(a) shows that as the input scale factor increases, the RMSE does not change substantially but shows a slight increasing trend, and as the sparseness of the storage pool increases, the RMSE does not change substantially. Figure 22(b) shows that as the spectral radius increases, the RMSE shows a slight decreasing trend. In summary, the spectrum radius *SR* is set as 0.9, the input scale factor *IS* as 0.8, and the sparseness of the storage pool *SD* as 0.3.

REFERENCES

[1] D. Dali, X. Lei, and W. Yuan, "The application of manned/unmanned aerial vehicle cooperative combat and its influence on war form," *Unmanned Syst. Technol.*, vol. 3, no. 4, pp. 1–9, 2020.

[2] Y. Dong, J. Ai, and J. Liu, "Guidance and control for own aircraft in the autonomous air combat: A historical review and future prospects," *J. Aerosp. Eng.*, vol. 133, no. 16, pp. 5943–5991, 2019.

[3] Z. M. Chen, P. Wu, Y. Bo, and M. Tian, "Adaptive control bat algorithm intelligent optimization particle filter for maneuvering target tracking," *Tien Tzu Hsueh Pao/Acta Electronica Sinica*, vol. 46, no. 4, pp. 886–894, 2018.

[4] W. Lei and L. Shizhong, "Tactical intention recognition of aerial target based on XGBoost decision tree," *J. Meas. Sci. Instrum.*, vol. 9, no. 2, pp. 148–152, 2018.

[5] W. Schuster, "Trajectory prediction for future air traffic management—Complex manoeuvres and taxiing," *Aeronaut. J.*, vol. 119, no. 1212, pp. 121–143, Feb. 2015.

[6] Z. Tao, G. Yang, and Z. Chengwei, "Short-term 4D trajectory prediction based on KF joint EKF parameter identification," *J. Civil Aviation Univ. China*, vol. 34, no. 5, pp. 1–6, 2016.

[7] K. Zhao, C. Wang, and G. Xiao, "Research for nonlinear model predictive controls to laterally control unmanned vehicle trajectory tracking," *Appl. Sci.*, vol. 10, no. 17, pp. 6034–6049, Aug. 2020.

[8] Z. Dailing, L. Humin, and L. Jiong, "Trajectory prediction of hypersonic vehicle based on adaptive IMM," *Aeronaut. Sci. Found. China*, vol. 37, no. 11, pp. 3457–3466, Mar. 2016.

[9] Z. Kai, X. Jiajun, and L. Fan, "Bayesian trajectory prediction for a hypersonic gliding reentry vehicle based on intent inference," *J. Astronaut.*, vol. 39, pp. 1262–1265, Nov. 2018.

[10] G. Xie, H. Gao, L. Qian, B. Huang, K. Li, and J. Wang, "Vehicle trajectory prediction by integrating physics- and maneuver-based approaches using interactive multiple models," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5999–6008, Jul. 2018.

[11] W. Mathew, R. Raposo, and B. Martins, "Predicting future locations with hidden Markov models," in *Proc. ACM Conf. Ubiquitous Comput. UbiComp*, 2012, pp. 911–918.

[12] Q. Wang, Z. Zhang, and Z. Wang, "The trajectory prediction of spacecraft by grey method," *Meas. Sci. Technol.*, vol. 27, no. 8, Jul. 2017, Art. no. 085011.

[13] S. Qiao, D. Shen, X. Wang, N. Han, and W. Zhu, "A self-adaptive parameter selection trajectory prediction approach via hidden Markov models," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 284–296, Feb. 2015.

[14] L. Ma, Y. Gao, T. Yin, and W. Zhai, "Improved flight conflict detection algorithm based on gauss-Hermite particle filter," *Wuhan Univ. J. Natural Sci.*, vol. 22, no. 3, pp. 269–276, Jun. 2017.

[15] X. Wang, X. Jiang, and Y. Wu, "A second-order HMM trajectory prediction method based on the spark platform," *J. Inf. Hiding Multimedia Signal Process.*, vol. 10, no. 2, pp. 346–358, Mar. 2019.

[16] Y. Lin, J. W. Zhang, and H. Liu, "An algorithm for trajectory prediction of flight plan based on relative motion between positions," *Frontiers Inf. Technol. Electron. Eng.*, vol. 19, no. 7, pp. 95–106, 2018.

[17] L. Ma and S. Tian, "A hybrid CNN-LSTM model for aircraft 4D trajectory prediction," *IEEE Access*, vol. 8, pp. 134668–134680, 2020.

[18] W. Zeng, Z. Quan, Z. Zhao, C. Xie, and X. Lu, "A deep learning approach for aircraft trajectory prediction in terminal airspace," *IEEE Access*, vol. 8, pp. 151250–151266, 2020.

[19] Z. Hongpeng, H. Changqiang, and T. Shangqin, "CNN-based real-time prediction method of flight trajectory," *Acta Armamentarii*, to be published. [Online]. Available: https://kns.cnki.net/kcms/detail/11.2176.TJ.20200622.1419.002.html

[20] Z. H. C. Yongbo, "Real-time air combat flight trajectory prediction using GRU," *Syst. Eng. Electron.*, to be published. [Online]. Available: http://kns.cnki.net/kcms/detail/11.2422.TN.20200226.1154.042.html

[21] Y. Liu and M. Hansen, "Predicting aircraft trajectories: A deep generative convolutional recurrent neural networks approach," 2018, *arXiv:1812.11670*. [Online]. Available: http://arxiv.org/abs/1812.11670

[22] R. Yike, "Traffic flow prediction based on improved LSTM network," M.S. thesis, Dalian Univ. Technol., Dalian, China, 2020.

[23] C. Gallicchio, A. Micheli, and L. Pedrelli, "Design of deep echo state networks," *Neural Netw.*, vol. 108, pp. 33–47, Aug. 2018.

[24] H. Hu, L. Wang, L. Peng, and Y.-R. Zeng, "Effective energy consumption forecasting using enhanced bagged echo state network," *Energy*, vol. 193, Feb. 2020, Art. no. 116778, doi: 10.1016/j.energy.2019.116778.

[25] C. Guojian and W. Junjie, "Deep echo state network overview," *Electron. Sci. Tech.*, vol. 31, no. 8, pp. 92–95, 2018.

[26] Q. Ma, L. Shen, and G. W. Cottrell, "DeePr-ESN: A deep projection-encoding echo-state network," *Inf. Sci.*, vol. 511, pp. 152–171, Feb. 2020.

[27] C. Gallicchio and A. Micheli, "Deep echo state network (DeepESN): A brief survey," Tech. Rep., Sep. 2020.

[28] W. Gaige, L. H. Guo, H. Duan, L. Liu, and H. Q. Wang, "The model and algorithm for the target threat assessment based on Elman–AdaBoost strong predictor," *Acta Electronica Sinica*, vol. 40, no. 5, pp. 901–906, Aug. 2012.

[29] D.-C. Feng, Z.-T. Liu, X.-D. Wang, Y. Chen, J.-Q. Chang, D.-F. Wei, and Z.-M. Jiang, "Machine learning-based compressive strength prediction for concrete: An adaptive boosting approach," *Construct. Building Mater.*, vol. 230, Jan. 2020, Art. no. 117000.

[30] A. Taherkhani, G. Cosma, and T. M. McGinnity, "AdaBoost-CNN: An adaptive boosting algorithm for convolutional neural networks to classify multi-class imbalanced datasets using transfer learning," *Neurocomputing*, vol. 404, pp. 351–366, Sep. 2020.

[31] X.-J. Sui, M.-Y. Li, Y.-L. Ying, B.-Y. Yan, H.-F. Wang, J.-L. Zhou, Z. Gu, and Y.-T. Long, "Aerolysin nanopore identification of single nucleotides using the AdaBoost model," *J. Anal. Test.*, vol. 3, no. 2, pp. 134–139, Apr. 2019.
[32] N. Chouikhi, B. Ammar, N. Rokbani, and A. M. Alimi, "PSO-based analysis of echo state network parameters for time series forecasting," *Appl. Soft Comput.*, vol. 55, pp. 211–225, Jun. 2017.
[33] W. X. R. Jialiang, "Trajectory prediction of target aircraft based on HPSO-TPFENN neural network," *J. Northwestern Polytechnical Univ.*, vol. 37, no. 3, pp. 612–619, Jun. 2019.
[34] J. Long, Z. Sun, C. Li, Y. Hong, Y. Bai, and S. Zhang, "A novel sparse echo autoencoder network for data-driven fault diagnosis of delta 3-D printers," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 3, pp. 683–692, Mar. 2020.
[35] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.
[36] L. Yang, Y. Wu, J. Wang, and Y. Iiu, "Research on recurrent neural network," *J. Comput. Appl.*, vol. 38, no. S2, pp. 1–6, 2018.
[37] K. Junwei, H. Yang, L. Junjiang, and Y. Zhijun, "Dynamic prediction of cardiovascular disease using improved LSTM," *Int. J. Crowd Sci.*, vol. 3, no. 1, pp. 14–25, May 2019.
[38] P. Wang, H. Wang, H. Zhang, F. Lu, and S. Wu, "A hybrid Markov and LSTM model for indoor location prediction," *IEEE Access*, vol. 7, pp. 185928–185940, 2019.
[39] Q. Jiang, C. Tang, C. Chen, X. Wang, and Q. Huang, "Stock price forecast based on LSTM neural network," *Proc. 12th Int. Conf. Manage. Sci. Eng. Manage.*, 2019, pp. 393–408.
[40] H. He, S. Yi, and W. Liu, "Intelligent english learning model based on BPTT algorithm and LSTM network," *J. Intell. Fuzzy Syst.*, vol. 153, pp. 1–12, Jan. 2020.
[41] M. Li, H. Zhao, X. Weng, and T. Han, "Differential evolution based on optimal Gaussian random walk and individual selection strategies," *Control Decis.*, vol. 31, no. 8, pp. 1379–1386, 2016.

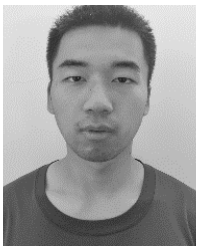
LEI XIE was born in Changzhou, Jiangsu, China, in 1997. He received the B.E. degree in arms engineering from Air Force Engineering University, Xi'an, in 2019, where he is currently pursuing the master's degree.

His research interests include air combat, intelligent optimization algorithm, maneuvering decision, situation assessment, and trace line aiming of gun.

ZHENGLEI WEI was born in Gansu, China, in 1991. He received the Ph.D. degree. He is an Engineer with the China Aerodynamics Research & Development Center. His research interests include maneuvering prediction and maneuvering decision.

DALI DING was born in Hunan, China, in 1980. He received the Ph.D. degree from Air Force Engineering University. He is currently an Associate Professor and Master Tutor with the Air Force Engineering University. His research field is weapon systems and application engineering.

ZHUORAN ZHANG was born in Beijing, China, in 1991. He received the Ph.D. degree from the Air Force Engineering University, in 2019. He is currently with 95806 Unit. His research field is unmanned aerial vehicle combat systems and technology.



ANDI TANG was born in Chongqing, China, in 1996. He is currently pursuing the master's degree in unmanned aircraft combat system and technology with the Air Force Engineering University, Xi'an. His research interests include mission planning.