

Received 14 November 2023, accepted 25 November 2023, date of publication 1 December 2023,  
date of current version 12 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3338727

## RESEARCH ARTICLE

# SVM-BTS Based Trajectory Identification and Prediction Method for Civil Rotorcraft UAVs

QINGCHUN JIAO<sup>1,2</sup>, LIN BAO<sup>1</sup>, HUIHUI BAI<sup>1</sup>, HUIJIE NIU<sup>1</sup>, AND CHAO HAN<sup>1</sup>

<sup>1</sup>School of Automation and Electrical Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China

<sup>2</sup>Zhejiang Province Security Technology and Prevention Industry Association, Hangzhou 310023, China

Corresponding author: Lin Bao (blin6599@163.com)

This work was supported in part by the Special Fund for Basic Scientific Research of the Zhejiang University of Science and Technology under Grant 2023QN043.

**ABSTRACT** To address the issue of low predictive accuracy in complex trajectory forecasting for civilian rotorcraft unmanned aerial vehicles (UAVs), this paper presents a method that utilizes the SVM-BTS technique for recognizing and predicting these intricate trajectories. Initially, the Support Vector Machine-Binary Tree Support Vector Machine model (SVM-BTS) is employed to segmentally recognize the complex trajectories of civilian UAVs. Based on this identification, five distinct flight states are identified: vertical, pitch, transverse, roll, and transitional. To assess the predictive performance of these states, a combination of Sliding Window Polynomial Least Squares, Unscented Kalman Filtering, and Long Short-Term Memory neural network methods is utilized. Consequently, the most suitable prediction algorithm is determined for each flight state. Experimental results demonstrate that the SVM-BTS recognition method, compared to SVM, achieves a 10.4% increase in recognition accuracy. Across different flight datasets, this prediction method exhibits the lowest mean squared error values compared to SWPLS, UKF, and LSTM. Therefore, this study accurately predicts the complex flight trajectories of civilian rotorcraft UAVs, enhancing the precision and efficiency of UAV flight prediction.


**INDEX TERMS** Trajectory classification, binary tree SVM, motion state identification, joint prediction.

## I. INTRODUCTION

In the 1920s, the inaugural development of the first unmanned aerial vehicle (UAV) marked the advent of the unmanned era [1]. Subsequently, from the 1940s onward, drones were primarily employed as military assets within the realm of aerial warfare [2]. It wasn't until the 1980s and 1990s that UAVs ventured into the civilian domain, gaining prevalence due to their compact design, convenience, and user-friendly operation as time progressed. Currently, most civilian UAVs are consumer-grade or industrial models. They are used for a wide range of tasks, including aerial photography, farm management, weather sensing, and inspections of power grids [3], [4], [5], [6], [7]. The increasing prevalence of UAVs has brought about the simultaneous appearance of associated risks and dangers. In the period spanning 2015 to 2020, the global UAV landscape witnessed a total

of 1,296 incidents, predominantly characterized by drone-related mishaps. A noteworthy revelation, as reported by the Federal Aviation Administration (FAA) in the United States, denotes a total of 583 UAV accidents transpiring during the timeframe extending from August 2015 to January 2016 [8]. These unfortunate occurrences wrought numerous casualties and incurred substantial property losses. Consequently, NASA has proffered the conceptual framework of Unmanned Aircraft System Traffic Management (UTM) [9]. However, there are still a lot of unanswered technological questions about low-altitude UAVs that operate under the control of the traffic management system. This paper aims to investigate intricate trajectory prediction methods tailored for civilian UAVs, with applications intended to address critical issues related to civilian UAV flight safety, motion state identification, and resource optimization.

The trajectory prediction of UAVs constitutes an integral facet of UAV flight oversight. UAV trajectory prediction involves using mathematical computing methods based on

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

past UAV motion data to anticipate the future flight route of UAVs [10]. The prediction of UAV trajectories allows for advanced knowledge of UAV flight paths, facilitating the adoption of appropriate measures in exigent circumstances to ensure flight safety. However, when dealing with civilian UAVs, characterized by their diminutive scale, agility, and human-operated nature, their flight trajectories exhibit considerable variability and diversity, posing formidable challenges to trajectory prediction. Due to the complex nature of civilian UAV flight paths, researchers have lately started to concentrate on identifying patterns in UAV flight motion. The ability to accurately identify the motion patterns of UAVs allows for the quick change of model parameters, resulting in more accurate forecasts of their trajectories [11]. Clearly, the accurate prediction of complex UAV trajectories now heavily depends on the determination of UAV motion states, which is more crucial than traditional trajectory prediction approaches. However, there is a lack of study in the field of categorizing and identifying the flying conditions of UAVs, which hinders progress in creating accurate prediction algorithms that can be used for many types of movements.

This paper focuses on the study of complex trajectories formed by various motion states. It introduces a Support Vector Machine-Binary Tree Support Vector Machine (SVM-BTS) model for recognizing different motion states within complex trajectories. Subsequently, the most suitable prediction algorithms are determined for each flight state, resulting in the development of a comprehensive method for predicting the complex trajectories of civilian rotorcraft UAVs. The organization of this article is as follows: Section II reviews prior research on UAV trajectory prediction. Section III presents trajectory prediction algorithms based on UAV state identification. Section IV includes a classification identification study based on historical UAV flight data and comparative experiments with trajectory prediction algorithms. Section V provides a summary of the research conducted in this paper and outlines future research directions.

## II. RELATED WORK

Trajectory prediction for UAVs serves as the foundation for air traffic management. It has always been a critical aspect of traffic management, with precise trajectory prediction playing a vital role in preventing collisions and reducing unnecessary losses. In the past, numerous scholars have proposed various trajectory prediction methods, as shown in Table 1, primarily including state estimation models and machine learning models [12].

(1) State estimation models: State estimation models are established based on attributes such as latitude, longitude, altitude, and velocity of UAVs to formulate motion equations for trajectory estimation. These primarily encompass the Kalman Filtering algorithm (KF), Particle Filtering algorithm, Hidden Markov Models (HMM), and their improved iterations [13], [14], [15]. Zhang and Yu [16] utilized domestic civil aviation trajectory data to construct a

flight motion model and applied the KF algorithm for trajectory prediction, further underscoring the significance of the Kalman filter in trajectory prediction. Lin et al. [15] predicted the 4D trajectory of aircraft before takeoff based on the HMM model and optimized HMM model parameters using the Expectation-Maximization (EM) algorithm, reducing computational complexity. However, since trajectory prediction is contingent on flight plan routes, deviation from the planned route significantly reduces prediction accuracy. In fact, trajectory prediction constitutes a complex stochastic estimation problem, and the aforementioned single-state models fail to capture state variations. In cases where aircraft operate in multiple flight states, multi-state estimation models should be employed. The Interacting Multiple Model (IMM) algorithm serves as a multi-state estimation model. IMM concurrently employs multiple motion models to track and predict the trajectories of moving targets, effectively mitigating model mismatch issues that occur when tracking and predicting targets with a single model [17]. Dalmau et al. [18] estimated aircraft guidance modes after observing flight data from Automatic Dependent Surveillance-Broadcast (ADS-B) and Mode S transponders, establishing corresponding models for aircraft guidance modes, and introducing the IMM algorithm to enhance short-term trajectory prediction accuracy. When dealing with real-time aircraft prediction, Bin et al. [19] combined the State-Dependent Transition Hybrid Estimation (SDTHE) with the Improved Intent Inference Algorithm (IIIA). This approach not only effectively accommodates real-time changes in aircraft motion modes but also overcomes the limitation of the likelihood function being zero in the Interacting Multiple Model (IMM) algorithm. Although state estimation models enhance prediction accuracy through physical modeling, they often find application in short-term prediction due to their inability to accurately capture long-term target maneuver uncertainties.

(2) Machine learning models: Analyzing patterns in trajectories over time using large amounts of data and using these patterns to make predictions about positions. This methodology basically entails analyzing extensive historical trajectory information to identify probable patterns, and utilizing these patterns to predict future flight positions. As a result, machine learning methods are effective for both long-term and short-term prediction. This primarily includes regression models [20], neural networks [21], [22], and hybrid models [23], [24], [25]. Kanneganti et al. [26] utilized aircraft direction and velocity data to construct a simple regression prediction model that accurately forecasted aircraft positions. However, when it comes to trajectory prediction, the paths of moving targets often exhibit nonlinearity, especially when dealing with small civilian UAVs. To improve the accuracy of nonlinear trajectory prediction, Shi et al. [27] introduced Long Short-Term Memory (LSTM) neural networks as a trajectory prediction tool, enhancing prediction accuracy by computing correlations between states within trajectory sequences. Due to the assumed structures of machine learning models, there may be some degree of error in the training models, and

machine learning-based methods cannot explicitly explain the current motion of aircraft. Therefore, machine learning models may produce future states that violate aircraft dynamics. To address this, Barratt et al. [28] integrated clustering algorithms with machine learning techniques. They initially clustered trajectories using K-means and then constructed Gaussian mixture models based on these clusters, ultimately formulating a terminal domain probability trajectory generation model. Furthermore, Choi et al. [25] combined machine learning with estimation-based methods, designing a Residual-Averaging Interactive Multiple Model to significantly enhance trajectory prediction accuracy. Although machine learning surpasses state estimation approaches in dealing with nonlinear systems, it requires high-quality data for optimal performance. The quality of flight data gathered by civilian UAVs may diminish in certain settings when climatic elements such as temperature, wind speed, and precipitation are present [29]. Therefore, machine learning and state estimation methods should be appropriately combined in trajectory prediction to ensure the best prediction results under different flight conditions.

Presently, there is an increasing amount of research being conducted in the domain of UAV regulation and trajectory prediction. This study mostly concentrates on commercial aircraft, with comparatively few studies pertaining to civilian UAVs. Commercial UAVs adhere to planned flight routes; however, civilian UAVs exhibit high flight maneuverability. Consequently, trajectory prediction algorithms developed for commercial UAVs may not be suitable for civilian UAVs. It is worth noting that the trajectories of civilian UAVs are marked by complexity and uncertainty. Sole reliance on a single prediction model can lead to an inability to adapt promptly to new flight patterns, resulting in diminished accuracy and efficiency in trajectory prediction [29]. Therefore, to improve the precision of UAV trajectory estimation, it is not enough to only train trajectory prediction models using data. Instead, it is necessary to take into consideration the present flight state of the UAV and develop predictive algorithms based on the motion state. Recently, researchers have been studying the mobility behavior of UAVs. For instance, Wang et al. [30] used the Fuzzy Min-Max (FMR) method to classify UAV flight data into several motion modes. The adoption of this approach successfully overcomes the constraints associated with single-data-driven methods in fully capturing flight trajectories, leading to an enhanced precision in trajectory prediction. Notably, the method presented in this paper surpasses conventional approaches by conducting a comprehensive experimental evaluation, considering both the mean square error and average delay of the prediction results. This dual assessment not only elevates the accuracy of predicting intricate trajectories but also ensures the real-time efficacy of trajectory prediction.

This study presents a collaborative model that combines Support Vector Machines (SVM) and Binary Tree Support Vector Machines (SVM-BTS) to accurately classify and recognize the flight states of UAVs. The aim is to address the

challenge of predicting complex trajectories for these UAVs. Firstly, initial feature extraction for UAVs is conducted, and SVM is employed for the preliminary classification of UAV trajectories. Subsequently, secondary feature extraction is performed, utilizing the Binary Tree SVM classification method to discern the motion postures of UAVs. Finally, a comprehensive predictive model is established, integrating Sliding Window Polynomial Least Squares, Unscented Kalman Filters, and Long Short-Term Filters, enabling trajectory prediction for UAVs across five distinct motion states: vertical, pitch, lateral, roll, and transition. This approach begins with the classification and identification of complex UAV trajectories, followed by the application of specific prediction algorithms tailored to different motion states, significantly enhancing the accuracy and reliability of complex trajectory predictions.

### III. THE PROPOSED UAV MOTION STATE IDENTIFICATION AND PREDICTION METHOD BASED ON SVM-BTS

The method proposed in this paper comprises four main components. The first component involves data collection and preprocessing. The second component focuses on a two-stage feature extraction process for trajectory characteristics. The third component is dedicated to UAV trajectory classification and identification based on SVM-BTS. The fourth component involves the trajectory prediction of UAVs based on state identification. An overview of the entire model workflow is illustrated in Fig. 1 below.

#### A. TRAJECTORY DATA COLLECTION AND PROCESSING

The actual UAV flight data used in this paper is acquired by installing an onboard terminal on the UAV, as illustrated in Fig. 2. The UAV's onboard terminal consistently stores UAV data programs and UAV flight status data, employing wireless communication technology to transmit real-time UAV flight data to the backend database.

The positioning module, along with its integrated sensors within the terminal, is proficient in acquiring data related to the UAV's longitude, latitude, altitude, speed, and heading angle. The characteristics of the individual trajectory data points collected and transmitted by the UAV's onboard terminal encompass ID (UAV model),  $lon$  (longitude),  $lat$  (latitude),  $H$  (altitude),  $\alpha$  (longitude yaw angle),  $\beta$  (latitude yaw angle),  $\gamma$  (altitude yaw angle),  $v$  (velocity),  $a$  (acceleration), and  $Date$  (timestamp), as illustrated in Table 2.

Data collection is often just the initial step in building a model. Data collected in the real world can often present various issues during data transmission due to signal interference, leading to outliers, missing values, duplicate values, and other problems.

In the case of UAV trajectory data, when certain fields or a set of fields in the database are found to be missing, it is necessary to address these missing values. For fields with a low rate of missing data, imputation is performed based on the distribution of neighboring data points. For fields with a high rate of missing data but low importance in terms of the

TABLE 1. Comparison of mainstream methods for trajectory prediction.

Categories		Advantages	Limitations	Representative literature
State estimation models	Kalman Filtering algorithm	<ul style="list-style-type: none"> <li>Modeling based on physical principles, enabling the provision of relatively accurate trajectory predictions.</li> <li>Model interpretability</li> </ul>	<ul style="list-style-type: none"> <li>Model construction is complex, posing challenges in capturing all factors.</li> <li>Unsuitable for complex scenarios</li> </ul>	Zhang <i>et al.</i> [16]
	Particle Filtering algorithm			Sakthivel <i>et al.</i> [14]
	Hidden Markov Models			Lin <i>et al.</i> [15]
	Improved iterations			Dalmau <i>et al.</i> [18] Bin <i>et al.</i> [19]
Machine learning models	Regression models	<ul style="list-style-type: none"> <li>Strong adaptability</li> <li>Capable of addressing complex nonlinear dynamical systems</li> </ul>	<ul style="list-style-type: none"> <li>Lacks interpretability</li> <li>Requires high data quality</li> <li>Involves significant computational costs</li> </ul>	Kanneganti <i>et al.</i> [26]
	Neural networks			WU <i>et al.</i> [21] Shi <i>et al.</i> [27]
	Hybrid models			Barratt <i>et al.</i> [28] Choi <i>et al.</i> [29]

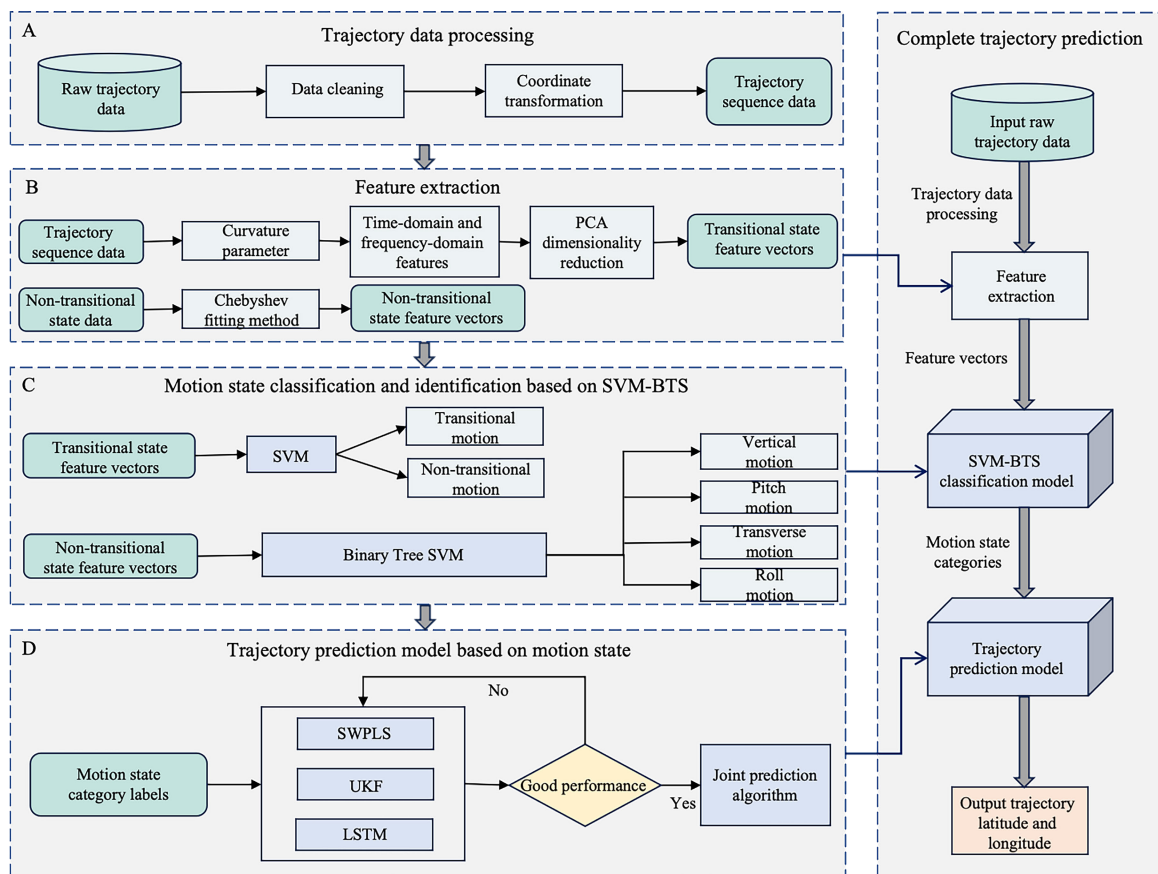


FIGURE 1. Process for UAV motion state identification and prediction based on SVM-BTS.

missing samples, direct deletion is applied. For fields with a high rate of missing data and high importance, imputation techniques are employed. In cases of duplicate values, this study removes duplicated data from the UAV flight data based on time and location features.

The UAV flight position data obtained using onboard terminals in this paper are given in the geodetic coordinate system as  $(lon, lat, H)$ , where the longitude and latitude data are not in the same dimension as the altitude data. Therefore, it is necessary to transform the geodetic coordinate system



FIGURE 2. UAV onboard terminal.

TABLE 2. Characteristics of single trajectory point.

Data characteristic	Example of data
ID	13
$lon(^{\circ})$	120.1417465209961
$lat(^{\circ})$	30.254547119140625
$H$ (m)	68.27
$\alpha(^{\circ})$	43.071
$\beta(^{\circ})$	29.135
$\gamma(^{\circ})$	32.143
$v$ (m/s)	0.936
$a$ (m/s <sup>2</sup> )	1.23
Date	2022-04-14 14:33:17

coordinates into the spatial Cartesian coordinate system as  $(X, Y, Z)$  [31].

The transformation from the geodetic coordinate system to the spatial Cartesian coordinate system is given by Equation (1):

$$\begin{cases} X=(R+H)\cos(lat)\cos(lon) \\ Y=(R+H)\cos(lat)\sin(lon) \\ Z=H \end{cases} \quad (1)$$

where  $lon, lat, H$  represent longitude, latitude, and altitude, respectively.  $R$  denotes the radius of curvature of the Earth in the geodetic coordinate system. The formula for calculating  $R$  is as follows:

$$R = \frac{m}{\sqrt{1 - e^2 \sin^2(lat)}} \quad (2)$$

$$e = \frac{m^2 - n^2}{m^2} \quad (3)$$

where  $m$  represents the Earth's major axis, and  $n$  represents the Earth's minor axis.

The transformation from geodetic coordinates  $(lon, lat, H)$  to spatial Cartesian coordinates  $(X, Y, Z)$  can be accomplished using Equation (1). When predicted trajectory data is obtained, the reverse transformation from spatial Cartesian coordinates  $(X, Y, Z)$  to geodetic coordinates  $(lon, lat, H)$  is performed to facilitate the visualization of precise trajectory

information. The specific formula for the conversion from spatial Cartesian coordinates  $(X, Y, Z)$  to geodetic coordinates  $(lon, lat, H)$  is as follows:

$$\begin{cases} \tan(lat) = \frac{Z}{\sqrt{X^2 + Y^2}} \left(1 + \frac{Re^2}{Z} \sin(lat)\right) \\ lon = \arctan \frac{Y}{X} \\ H = Z \end{cases} \quad (4)$$

### B. UAV TRAJECTORY FEATURE EXTRACTION

Trajectory feature extraction is the process of extracting key features or key attributes from the trajectory data to describe the flight behavior and patterns of the UAV. The extracted features can be used for subsequent classification and identification tasks. In the field of machine learning algorithms, having good data features will improve the accuracy of the model to some extent. In this paper, two feature extractions are carried out for two flight states of UAVs, namely transitional state feature extraction and non-transitional state feature extraction.

#### 1) TRANSITIONAL STATE FEATURE EXTRACTION

In this paper, a section of a complex trajectory is simulated according to the real flight law of small civil UAVs, as shown in Fig. 3.

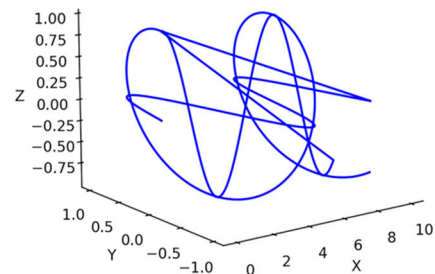


FIGURE 3. Complex trajectory map of UAV flight.

The acquired trajectory sequence is shown below:

$$S = \{s_1, s_2, \dots, s_n\} \quad (5)$$

$$s_n = [X_n Y_n Z_n v_n a_n t_n]^T \quad (6)$$

where  $X_n, Y_n, Z_n$  are the coordinate positions of the track point in the spatial Cartesian coordinate system, respectively, and  $v_n, a_n, t_n$  are the velocity, acceleration and timestamp of the track point. A sliding window with a window length of  $b$  and a sliding window with a step size of 1 is set to segment the trajectory sequence data. The data in any sliding window will be obtained as  $S'$ :

$$S' = \{s_i, s_{i+1}, \dots, s_{i+b-1}\} \quad (7)$$

where  $i, (i + 1), \dots, (i + b - 1) \in n$ . Based on the trajectory sequence data within the sliding window  $S'$ , it is possible to derive the speed and acceleration parameter change characteristics of the UAV. Additionally, the curvature parameter

change characteristics of the UAV can be computed based on the beginning data:

$$q_i = \sum_{f=1}^2 \frac{\sum_{j=i-f}^{i+f-1} d_{j,j+1}}{2d_{i-f,i+f}} \quad (8)$$

The symbol  $q_i$  represents the curvature, whereas  $d_{i-f,i+f}$  represents the distance between two coordinate points. Specifically, it refers to the distance from the  $i-f$  coordinate point to the  $i+f$  coordinate point. The variable  $f$  represents the lag. To enhance the accuracy of calculation outcomes, consider utilizing the lagged value  $f=1,2$ .

Within each sliding window, a total of 30 features are extracted from the three parameters of velocity, acceleration, and curvature. With these features are divided into six groups based on their time-domain characteristics and four groups based on their frequency-domain characteristics, it is possible to obtain a feature vector with 30 dimensions for every sliding window. The resultant dataset can be represented as:

$$S'' = \{R_1, R_2, \dots, R_m\} \quad (9)$$

Each of these data  $R_m$  is a 30 dimensional feature vector, and  $m$  is the number of segments for sliding window segmentation. However, the excessive dimensionality of the data is a challenge as it necessitates extensive computational resources. Hence, this study uses principal component analysis (PCA) as a means to reduce the dimensionality of the feature vectors. The specific method can be primarily categorized into the subsequent steps:

Step 1: The segmentation acquired by the previously mentioned sliding window segmentation method. The 30-dimensional data can be represented as a matrix  $M$ , which possesses a distinct structure:

$$M = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1,30} \\ r_{21} & r_{22} & \dots & r_{2,30} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \dots & r_{m,30} \end{bmatrix} \quad (10)$$

Step 2: Zero-mean each column of the sample matrix  $M$  to obtain a new matrix by zero-averaging each column of the sample matrix  $\bar{M}$ :

$$r_i \leftarrow r_i - \frac{1}{m} \sum_{i=1}^m r_i \quad (11)$$

Step 3: The correlation between the dimensions of the sample data can be calculated by utilizing the covariance matrix, denoted as  $Cov$ :

$$Cov = \frac{1}{m} \bar{M} \bar{M}^T \quad (12)$$

Step 4: Calculate the covariance matrix  $Cov$ . The eigenvalues and eigenvectors of the covariance matrix are listed in descending order of the eigenvalues:

$$(\tau_1, \tau_2, \dots, \tau_t) \rightarrow (p_1, p_2, \dots, p_t) \quad (13)$$

Step 5: According to the requirement of dimensionality reduction, such as down to  $k$  dimension, take the first  $k$  vectors to construct a dimension reduction matrix  $P$ :

$$P = (p_1, p_2, \dots, p_k)^T \quad (14)$$

Step 6: By transforming matrix transformations on matrix  $P$  The original sample  $M$  coordinate transformation should be modified in order to achieve the objective of reducing data dimensionality.

$$Y = M \cdot P \quad (15)$$

After dimensionality reduction of all data, it is necessary to compute the extent of information loss incurred as a consequence of this reduction:

$$E_1 = \frac{1}{k} \sum_{i=1}^k \|r^{(i)} - r_{approx}^{(i)}\|^2 \quad (16)$$

$$E_2 = \frac{1}{40} \sum_{i=1}^{40} \|r^{(i)}\|^2 \quad (17)$$

where  $r^{(i)}$  denotes  $k$  the data of the samples, and  $r_{approx}^{(i)}$  denotes the data after dimensionality reduction.

This is accomplished by  $\rho$  to measure the information loss after data dimensionality reduction, according to the  $\rho$  the size of the data to choose the appropriate  $k$  value.

$$\rho = \frac{E_1}{E_2} \quad (18)$$

## 2) NON-TRANSITIONAL STATE FEATURE EXTRACTION

Above is the UAV trajectory classified as transitional state feature extraction, and next will be the non-transitional state trajectory identification feature extraction. The same sliding window segmentation method is used to segment the UAV trajectory data. Set a window length of  $n$ . The sliding window with a step size of 1 is used to segment the trajectory data. Segment the data into  $m$  segments and get the sample set as:

$$D = \{x_1, x_2, \dots, x_m\} \quad (19)$$

In each sliding window, any one-dimensional time series data is represented as a matrix with a feature dimension of 5, as follows:

$$\begin{matrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{matrix} \begin{bmatrix} a_{11} & a_{12} & \dots & a_{15} \\ a_{21} & a_{22} & \dots & a_{25} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{n5} \end{bmatrix} \quad (20)$$

$$[a_{11}a_{12}a_{13}a_{14}a_{15}] = [X_{11}Y_{12}Z_{13}v_{14}a_{15}] \quad (21)$$

The variables  $X_{11}$ ,  $Y_{12}$ ,  $Z_{13}$ ,  $v_{14}$  and  $a_{15}$  are used to indicate the location coordinates, velocity, and acceleration of the UAV within the spatial coordinate system.

However, the trajectory data of UAV flights exhibit strong nonlinear properties, so the Chebyshev fitting method [32] is employed to individually segment each one-dimensional time series of the trajectory, denoted as  $\{a_1, a_2, \dots, a_n\}$ . The data within the sliding window is seen as a collection of two-dimensional data points, denoted as  $\{t_i, a_i\}$ , where  $t_i$

represents the time of the trajectory sequence data for  $i = (1, 2, \dots, n)$ . The fitting of the model is done based on the relationship between the parameters and their variation over time.

$$a_i = \sum_{j=1}^k c_j T_j(t_i) \tag{22}$$

Subsequently, the least squares method was employed to minimize the sum of squares of the residuals in order to achieve a better fit.

$$\min J = \sum_{i=1}^n V(t_i)^2 = [TC - A, TC - A] \tag{23}$$

where  $C = [c_0, c_1, \dots, c_k]^T$  is the matrix of coefficients of Chebyshev. The actual data of a specific dimension parameter is represented by the matrix  $A = [a_1, a_2, \dots, a_n]$ , which corresponds to the Chebyshev basis functions.

$$T = \begin{bmatrix} T_0(d_1) & T_1(d_1) & \dots & T_k(d_1) \\ T_0(d_2) & T_1(d_2) & \dots & T_k(d_2) \\ \vdots & \vdots & \ddots & \vdots \\ T_0(d_n) & T_1(d_n) & \dots & T_k(d_n) \end{bmatrix} \tag{24}$$

When the coefficient matrix satisfies Equation (25), it represents the optimal solution at this point.

$$T^T TC = T^T A \tag{25}$$

The identity of the parameter is determined by selecting the quotient of the first two Chebyshev coefficients from the solved matrix.

$$\mu = \frac{c_1}{c_0} \tag{26}$$

Therefore, for the above time series, the eigenvalue of the  $i$ th dimension is denoted as  $\mu_i$ . Arranging the eigenvalues of the five parameters sequentially allows us to obtain the feature vectors for each segment of the segmented trajectory.

$$p_i = [\mu_1, \mu_2, \dots, \mu_5] \tag{27}$$

**C. MOTION STATE CLASSIFICATION AND IDENTIFICATION BASED ON SVM-BTS**

This research presents a comprehensive model for the classification and identification of the complex trajectories of UAVs. The proposed approach is based on the Support Vector Machine-Binary Tree Support Vector Machine (SVM-BTS) technology. As shown in Fig. 4, the model comprises two main stages. The first stage involves the utilization of the support vector machine binary classification algorithm to differentiate between transitional flight states and non-transitional flight states. Subsequently, the second stage employs a Binary Tree SVM model to identify the four fundamental flight states (vertical motion, pitch motion, transverse motion, and roll motion) within the non-transitional flight states. This paper employs the SVM-BTS method to address this issue, as it is challenging to distinguish between the trajectory data generated by the transitional motion state and the roll motion in the non-transitional flight state.

The utilization of SVM-BTS to achieve the classification of trajectories formed by UAV transitional and non-transitional motion states is specified as follows:

Step 1: Training the SVM binary classification model. Train a binary classification SVM model using transitional state feature vectors as the input.

Step 2: Training Binary Tree SVM classification model. There are four distinct types of motion states observed in UAV non-transitional motion states, including vertical motion, pitch motion, transverse motion, and roll motion.

In this study, the samples are labeled as  $Y = \{0, 1, 2, 3, 4\}$ , representing the vertical motion, pitch motion, transverse motion, roll motion, and transitional motion states.

For a given data,  $M = \{x_1, x_2, \dots, x_m, \dots, y_1, y_2, \dots, y_n\}$ , where there are  $n_1$  training samples, denoted as  $x_1, x_2, \dots, x_{n_1}$ , for a given class  $i$ . Additionally, there are  $n_2$  training samples, denoted as  $y_1, y_2, \dots, y_{n_2}$ , for a certain category  $j$ .

The sample centers of concentration are respectively:

$$\bar{x}_1 = \frac{1}{n_1} \sum_{i=1}^{n_1} x_i, \bar{y}_1 = \frac{1}{n_2} \sum_{j=1}^{n_2} y_j \tag{28}$$

The variable  $d(x, y)$  denotes the Euclidean distance between class  $i$  and  $j$ .

$$d(x, y) = \sqrt{\sum_{i,j=1}^{n_1, n_2} (x_i, y_j)} \tag{29}$$

The distance between sample centers is:

$$d'(x, y) = \bar{x}_1 - \bar{y}_1 \tag{30}$$

By calculating the sample set  $M$ , classify the classes  $i$  and  $j$  with the maximum sample-to-sample distance  $d(x, y)$ , then place classes  $i$  and  $j$  into sets  $X1$  and  $X2$ , respectively. Update the sample set  $M$  as  $M = X - (X1 \cup X2)$ . Next, compute the minimum distance  $d'(x, y)$  from the remaining classes in the sample set to the sample centers of classes  $i$  and  $j$ , and compare the minimum distances  $d'(x, y)$  from other classes to classes  $i$  and  $j$ , respectively. If the distance from other classes to class  $i$  is smaller than the distance from other classes to class  $j$ , then place the other classes into set  $X1$ ; otherwise, place them into set  $X2$ . Continue this process until set  $M$  becomes empty. Then, proceed to construct a binary classifier for training on these two sets. Subsequently, repeat the above steps for classification training on each subset until all five UAV motion states are completely classified and recognized. In each SVM, the kernel function is set to RBF.

**D. UAV TRAJECTORY PREDICTION MODEL**

There are three main types of single prediction algorithms, namely, Sliding Window Polynomial Least Squares (SWPLS), Untraced Kalman Filter (UKF), and Long Short-Term Memory Neural Network (LSTM). SWPLS is more suitable for time series data with obvious trends and periodic changes. UKF is more suitable for systems dealing with linear or Gaussian noise. LSTM is suitable for the prediction of nonlinearly varying time series data. It can be seen

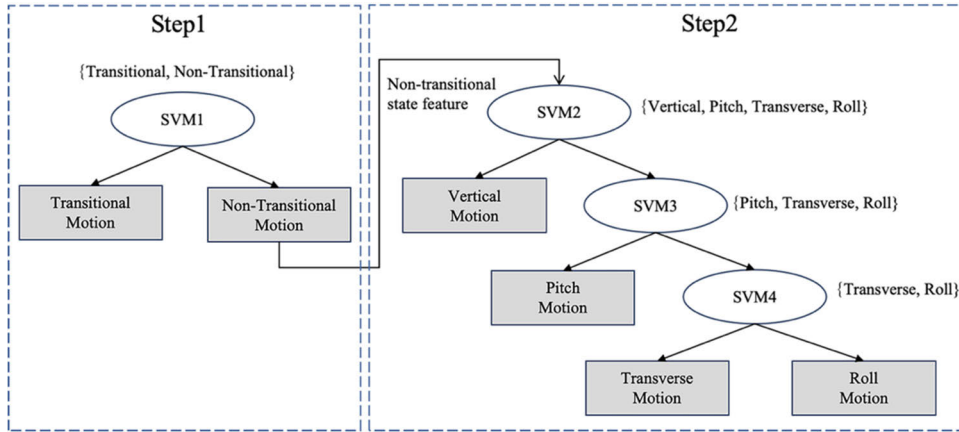


FIGURE 4. Classification and identification model.

that different prediction algorithms are suitable for different situations, and different prediction algorithms should be adopted for civil UAVs with complex trajectories according to different motion states so as to improve the accuracy and efficiency of prediction.

1) SLIDING WINDOW POLYNOMIAL LEAST SQUARES PREDICTION OF UAV TRAJECTORIES

Define a Sliding Window Polynomial Least Squares (SWPLS) is a polynomial least squares method for fitting long sequence data. The sliding window polynomial least squares method is used to fit long time series data into multiple short time series by setting a sliding window. This can avoid the problems of traditional polynomial least squares, such as excessive fitting errors caused by too much data in the antecedent. Specifically, the process of predicting UAV trajectories by SWPLS involves advancing the UAV’s longitude, latitude, and altitude data outward over time.

Suppose that the UAV flight trajectory over a short period of time is a set of polynomial time functions  $s(t_i)$  :

$$s(t_i) = \sum_{j=0}^m a_j t_i^j = i \text{ that } m < n - 1 \quad (31)$$

By minimizing the  $\sum_{i=1}^n (s(t_i) - x_i)^2$  with respect to the coefficient  $a_j$  through least squares estimation,  $\hat{a}_j$  is obtained:

$$A = (P^T \cdot P)^{-1} \cdot P^T \cdot X \quad (32)$$

$$A = (\hat{a}_0 \hat{a}_1 \cdots \hat{a}_{m-1} \hat{a}_m)^T \quad (33)$$

$$P = \begin{bmatrix} 1 & t_1 & \cdots & t_1^{m-1} & t_1^m \\ 1 & t_2 & \cdots & t_2^{m-1} & t_2^m \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & t_{n-1} & \cdots & t_{n-1}^{m-1} & t_{n-1}^m \\ 1 & t_n & \cdots & t_n^{m-1} & t_n^m \end{bmatrix} \quad (34)$$

$$X = (x_1 x_2 \cdots x_{n-1} x_n)^T \quad (35)$$

The optimal estimate of  $s(t)$  is:

$$s(t) = \hat{a}_j t^j \quad (36)$$

The trajectory prediction at time  $(t_n + dt)$  is:

$$\hat{s}(t_n + dt) = \sum_{j=0}^m \hat{a}_j (n + dt)^j \quad (37)$$

The highest power of the polynomial set in the prediction of the UAV trajectory is 2. The polynomial function of the trajectory of the UAV moving in the three directions of X-axis, Y-axis, and Z-axis is:

$$\begin{cases} s_x(t_i) = \sum_{j=0}^2 a_{xj} \cdot i^j \\ s_y(t_i) = \sum_{j=0}^2 a_{yj} \cdot i^j \\ s_z(t_i) = \sum_{j=0}^2 a_{zj} \cdot i^j \end{cases} \quad (38)$$

Finally, we can refer to Equation (33) to obtain the least-squares estimated coefficients for  $a_{xj}, a_{yj}, a_{zj}$  as  $\hat{a}_{xj}, \hat{a}_{yj}, \hat{a}_{zj}$ . Subsequently, we can utilize Equation (38) to predict the UAV’s position at time  $t_n$ .

2) UNTRACED KALMAN FILTER PREDICTION OF UAV TRAJECTORIES

Untraced Kalman Filter (UKF) is formed based on the untraced transform (UT) technique, which propagates the mean and covariance by means of nonlinear transformations. UKF utilizes the UT technique to apply the Kalman filter under linear conditions to a nonlinear system and then completes the nonlinear computation of the system by means of a set of sampling points for the mean and variance distributions.

The design of the UKF algorithm consists of the following six steps:

(1) Modeling the state of the system:

$$X_t = f(X_{t-1}, U_{t-1}, W_{t-1}) \quad (39)$$

$X_{t-1}$  is  $L$ -dimensional state matrix at time  $t - 1$ ,  $U_{t-1}$  is the  $L$ -dimensional input matrix at time  $t - 1$ , and  $W_{t-1}$  is the  $L$ -dimensional process noise matrix at time  $t - 1$ . For a nonlinear system, the system equation is:

$$\begin{cases} x_{k+1} = f(x_k, w_k) \\ y_k = h(x_k, v_k) \end{cases} \quad (40)$$



where  $w_k \in N(0, Q_k)$ ,  $v_k \in N(0, R_k)$ ,  $x_k$  is  $L$ -dimensional state matrix at time  $k$ ,  $w_k$  is the  $L$ -dimensional process noise matrix at time  $k$ , and  $v_k$  is the  $L$ -dimensional measurement noise matrix at time  $k$ .

Next, the state variables are expanded to obtain the augmented state matrix as well as to compute the mean of the augmented states:

$$\hat{x}_{a,k} = [x_k 00]^T \quad (41)$$

The variance of the augmented state is:

$$P_k = \begin{bmatrix} P_{a,k} & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & R \end{bmatrix} \quad (42)$$

(2) Initialization parameters.

Initialize process noise variance  $Q$ , measurement noise  $R$ , expected value  $\hat{x}_{a,k}$ , and covariance matrix  $P_k$ .

(3) Acquire sigma test points.

Construct augmented sigma points based on the expected value and covariance.

$$\begin{cases} x_{a,k-1}^0 = \hat{x}_{a,k-1} \\ x_{a,k-1}^j = \hat{x}_{a,k-1} + \sqrt{(N+k)P_{a,k-1}}_j (j = 1, 2, \dots, N) \\ x_{a,k-1}^j = \hat{x}_{a,k-1} - \sqrt{(N+k)P_{a,k-1}}_j (j = N+1, \dots, 2N) \end{cases} \quad (43)$$

where  $N$  is the dimension of the augmented state.

(4) Solve the time update equation to find the sampling point state estimate:

$$X_{x,k|k-1}^j = f(X_{x,k-1}^j, X_{w,k-1}^j) \quad (44)$$

(5) Solve the observation update equation for the observed estimate:

$$r_{k|k-1}^j = h(X_{x,k-1}^j, X_{w,k-1}^j) \quad (45)$$

(6) Kalman filter update to get  $K$ ,  $\hat{x}_k$ ,  $P_{x,k}$ :

$$K = P_{xy,k} P_{y,k}^{-1} \quad (46)$$

$$\hat{x}_k = \hat{\hat{x}}_k + K(\hat{y}_k - \hat{\hat{y}}_k) \quad (47)$$

$$P_{x,k} = P_{\hat{x},k} - K P_{y,k} K^T \quad (48)$$

### 3) LONG SHORT-TERM MEMORY NEURAL NETWORK PREDICTION OF UAV TRAJECTORIES

Long Short-Term Memory Neural Network (LSTM) is a special variant of Recurrent Neural Network (RNN), which is a neural network specialized in processing time series. The only defect of the RNN model is that it cannot deal with dependency on long time series, which led to the creation of LSTM. The structure of LSTM is shown in Fig. 5.

An LSTM consists of three gates: the forgetting gate, the input gate, and the output gate. The forgetting gate is used to forget redundant information, the input gate is used to transmit the input information, and the output gate is used to receive the information from the forgetting gate and the output gate of the previous moment, filter it, and then forward

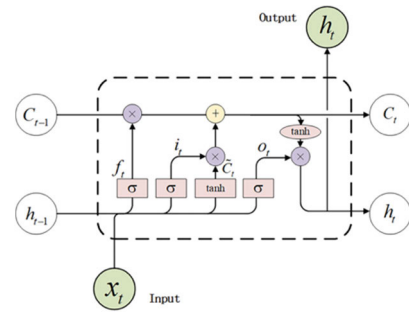


FIGURE 5. LSTM structure.

it to the next LSTM unit. The specific calculations are as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (49)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (50)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (51)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (52)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (53)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (54)$$

where  $f_t$  represents the forgetting gate, and  $i_t$  represents the input gate, and  $\tilde{C}_t$  and  $C_t$  represent the current input and cell states, respectively.  $\sigma$  uses the sigmoid function, where  $W_f$ ,  $W_i$ ,  $W_o$ , and  $W_c$  represent the weight matrices for the forgetting gate, input gate, output gate, and the current input unit state, respectively.  $[h_{t-1}, x_t]$  denotes the interconnections between two vectors, and  $b_f$ ,  $b_i$ ,  $b_o$  and  $b_c$  represent the bias terms for the forgetting gate, input gate, output gate, and the current input unit state, respectively.

The specific steps are as follows:

Step 1: Assume that the track point data parameters are  $s_r = [X, Y, Z, v, a, t]$ , normalize the data:

$$M^* = \frac{M - M_{min}}{M_{max} - M_{min}} \quad (55)$$

where  $M_{max}$  is the maximum value of the track point parameter, and  $M_{min}$  is the minimum value of the track point parameter, and  $M$  is the original training parameter, and  $M^*$  is the data after parameter normalization.

Step 2: Establish the LSTM deep neural network training model, take the normalized UAV trajectory dataset as an input into the hidden layer of the neural network, and finally compute the data through the fully connected layer to output the UAV trajectory position at a future moment.

### 4) HYBRID ATTITUDE JOINT PREDICTION ALGORITHM

The simulated complex UAV trajectories, encompassing five distinct motion states, including vertical, pitch, transverse, roll, and transitional motions, are subjected to classification and identification using the SVM-BTS algorithm. Subsequently, three prediction algorithms, namely SWPLS, UKF, and LSTM, are deployed to forecast the trajectory data originating from a particular UAV motion state. By contrasting

prediction data errors and average time delays among trajectories associated with various motion states, the optimal trajectory prediction algorithm for a given UAV motion state is chosen.

The training parameters for these trajectory points comprise longitude, latitude, altitude, velocity, acceleration, and time. The algorithm parameters are determined by comparing the Root Mean Square Error (RMSE) of the trajectory prediction results, defined as:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{t=1}^N (\text{observed}_t - \text{predicted}_t)^2} \quad (56)$$

The parameters exhibiting fewer RMSE values are chosen, and the algorithm parameters are presented in Table 3.

**TABLE 3. Parameters of trajectory of UAV predicted by LSTM.**

Parameters	Worth
Sliding window length	8
Segmentation of training track segments	9612
Number of hidden layers	1
Learning rate	0.001
Number of samples per training	10
Number of neurons in the hidden layer	144

Using the above three algorithms, trajectory prediction is conducted separately for the vertical, pitch, transverse, roll, and transitional motions of simulated UAV flights. Each motion state consists of 40 sets of trajectory data. The prediction errors for SWPLS, UKF, and LSTM for each motion state are calculated. The error comparison plots are shown in Fig. 6 (a, b, c, d, e). The average delay predicted by the three algorithms for different motion states is shown in Table 4.

When the UAV is in vertical motion, pitch motion, and transverse motion, the errors of the three trajectory prediction algorithms are within the range of 0 to 0.5m, but the average delay of SWPLS prediction is not more than 0.3s lower than that of UKF and LSTM. So when the UAV is in the process of recognizing vertical motion, pitch motion, and transverse motion by using the SVM-BTS method, it is chosen to predict the trajectory of the UAV using SWPLS.

When the UAV is in roll motion, the error of SWPLS among the three trajectory prediction algorithms is more than 1.5m, and the errors of UKF and LSTM still remain in the range of 0 to 0.5m, but the average prediction delay of UKF is lower than that of LSTM. So, when the UAV is recognized to be in the process of roll motion using the SVM-BTS method, UKF is chosen to predict the UAV trajectory.

The prediction error of LSTM is minimized when the UAV is in a transitional motion state. In this case, LSTM is chosen to predict the trajectory of the UAV in a transitional motion state.

Thus, the expression for the joint prediction algorithm under mixed postures is derived as follows:

$$\text{Joint Prediction Algorithm} = \begin{cases} \text{SWPLS} & m_v, m_p, m_t \\ \text{UKF} & m_r \\ \text{LSTM} & m_g \end{cases} \quad (57)$$

where  $m_v, m_p, m_t, m_r, m_g$  represent the vertical motion, pitch motion, transverse motion, roll motion, and transitional motion states recognized by SVM-BTS, respectively.

## IV. EXPERIMENTAL COMPARISON AND ANALYSIS

### A. EXPERIMENTAL ENVIRONMENT

In the experiments, the TensorFlow framework is used to build predictive models, and Pycharm is used as an integrated development environment. The TensorFlow framework simplifies the construction of neural networks and supports both CPU and GPU environments. The experimental environment in this study is: CPU: AMD R7 4800H@2.90 GHz, RAM: 16GB.

The parameters involved in the experiments are as follows: C and gamma are 0.1 and 10, respectively. The number of single training samples used in batch-size is 10, the number of training round epochs is 50, the learning rate is 0.001, the Adam optimizer is used, and the mean square error is computed to indicate the cost of the training process.

A total of 864 flight trajectories were collected, each labeled with vertical motion, pitch motion, transverse motion, roll motion, and transitional motion tags, denoted as 0, 1, 2, 3, and 4, respectively. Out of these 864 trajectory segments, 720 data sets were employed for training purposes, while the remaining 144 data sets were designated for testing. The composition of these data sets is provided in Table 5.

### B. EVALUATION INDICATORS

The assessment of classification and identification employs a confusion matrix, which is employed to depict the performance of the proposed detection and classification techniques. Accuracy (Acc), Precision (PRC), Recall (REC), and F1 scores serve as performance metrics for model evaluation:

$$\text{PRC} = \frac{M[i, i]}{\sum_j M[j, i]} \quad (58)$$

$$\text{REC} = \frac{M[i, i]}{\sum_j M[i, j]} \quad (59)$$

$$\text{F1} = \frac{2 \times \text{PRC} \times \text{REC}}{\text{PRC} + \text{REC}} \quad (60)$$

$$\text{Acc} = \frac{\sum_i M[i, i]}{\sum_{i,j} M[i, j]} \quad (61)$$

The  $i \in \{0, 1, 2, 3, 4\}$  are represented as the UAV vertical motion, pitch motion, transverse motion, roll motion, and transitional motion states, respectively.  $M[i, j]$  is a  $5 \times 5$  confusion matrix where the element in the  $i$ th row and  $j$ th column of  $M$  represents the number of samples with label  $i$  predicted as label  $j$ .

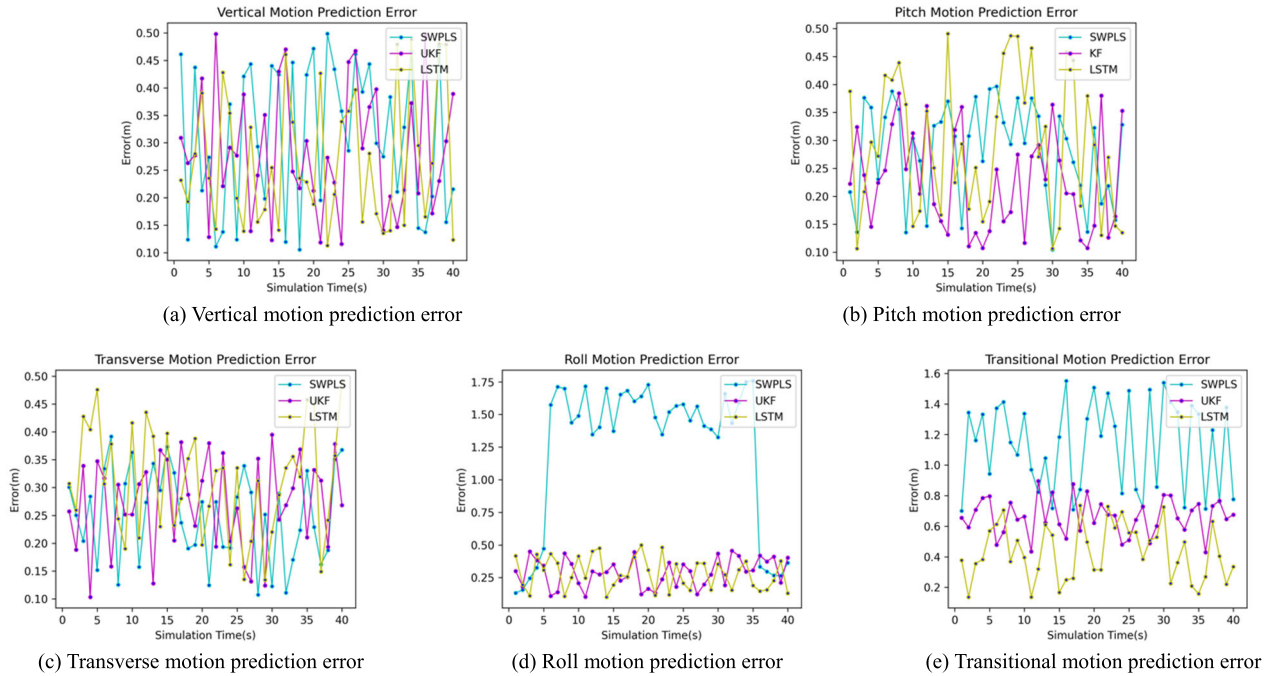


FIGURE 6. Comparison of prediction errors.

TABLE 4. Average delay for trajectory prediction.

Prediction model	SWPLS	UKF	LSTM
	Average delay(s)	Average delay(s)	Average delay(s)
Vertical motion	<b>0.263</b>	0.615	0.864
Pitch motion	<b>0.265</b>	0.624	0.728
Transverse motion	<b>0.265</b>	0.635	0.952
Roll motion	<b>0.262</b>	0.627	0.756
Transitional motion	<b>0.257</b>	0.624	0.834

TABLE 5. Data components.

Category	Train	Test
Vertical motion	165	33
Pitch motion	90	18
Transverse motion	135	27
Roll motion	205	41
Transitional motion	125	25
Total	720	144

In order to facilitate the comparison between predicted values and actual values, and to analyze the predictive performance of the model, mean squared error (MSE) is chosen as the evaluation metric.

$$MSE = \frac{1}{N} \sum_{t=1}^N (y_t - \hat{y}_t)^2 \quad (62)$$

where  $y_t$  represents the actual observed trajectory coordinates,  $\hat{y}$  represents the predicted trajectory coordinates, and  $N$  denotes the number of samples.

### C. ANALYSIS OF EXPERIMENTAL RESULTS

#### 1) ANALYSIS OF CLASSIFICATION AND IDENTIFICATION EXPERIMENT RESULTS

The Binary Tree SVM and SVM-BTS are utilized to classify and recognize the complex trajectories, respectively, and the PRC, REC, and F1 values of the Binary Tree SVM and SVM-BTS are shown in Table 6. The recognized confusion matrix is shown in Fig. 7 (a, b).

Table 6 provides clear evidence that the SVM-BTS method proposed in this study outperforms Binary Tree SVM in the task of recognizing the motion states of civilian UAVs with complex trajectories. All performance metrics exceed 97%, with PRC surpassing Binary Tree SVM by 3%, REC being 8% higher than Binary Tree SVM, and the F1 score

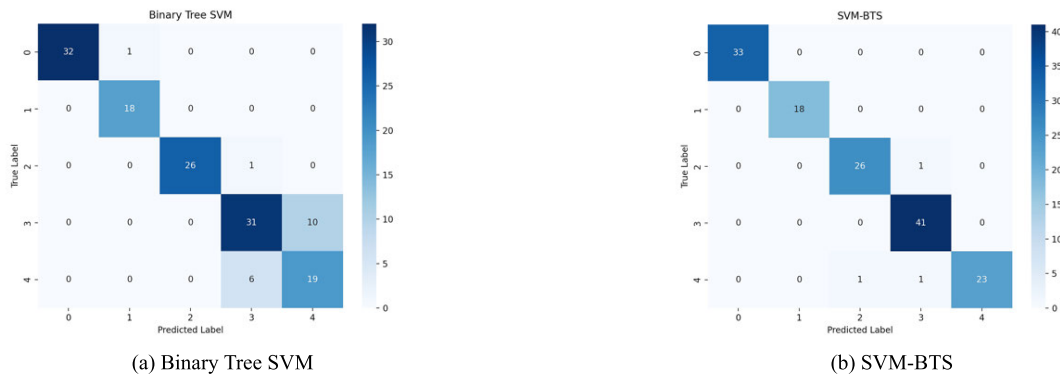


FIGURE 7. Comparison of confusion matrices.

outperforming SVM by 6.2%. Particularly noteworthy is the accuracy, which is 10.4% higher than Binary Tree SVM. Fig. 7 illustrates the confusion matrices for both Binary Tree SVM and SVM-BTS, where the horizontal and vertical axes represent the predicted labels and true labels of UAV motion states, respectively. The diagonal elements represent the number of samples correctly predicted in the respective category. Fig. 7 clearly demonstrates that SVM-BTS exhibits significant superiority over Binary Tree SVM in the overall classification and identification of various motion states for UAVs with complex trajectories.

## 2) ANALYSIS OF EXPERIMENTAL RESULTS FOR COMPLEX TRAJECTORY PREDICTION

In order to investigate the predictive performance of our joint prediction model at various time scales, experiments were conducted with prediction time intervals set at 0.5s, 1s, 1.5s, 2s, 2.5s, as well as 5s, 10s, 15s, 20s, and 25s. Real UAV flight data was used for prediction, and the predictive performance is illustrated in Fig. 8 (a, b).

In Fig. 8 (a, b), the gray shaded region represents the confidence band, with a 95% confidence level set in this study. The lower boundary of the confidence interval represents the lowest possible value of Mean Square Error (MSE), while the upper boundary represents the highest possible value of MSE. The blue curve represents the average value of MSE, approximately fitted as a curve along the confidence interval. The black segments indicate error bars, representing the standard deviation of MSE values. When the prediction time interval is 0.5s, the mean square error of the trajectory prediction method of the joint prediction algorithm in this paper is 0.322m, the mean square error is 0.523m when 1s, and the mean square error is 4.23m when 5s. Beyond 10s, the MSE rapidly increased, exceeding 10m, resulting in a significant decrease in prediction accuracy. To better illustrate the predictive performance of our method for UAV trajectories at different prediction time intervals, we selected and displayed the predictive results for intervals of 0.5s, 1s, 5s, and 10s, as shown in Fig. 9 (a, b, c, d).

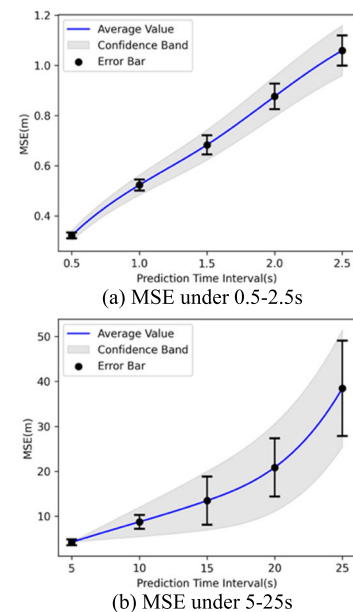


FIGURE 8. Mean square error under different prediction time intervals.

Based on Fig. 9, it is visually evident that the joint prediction model proposed in this paper demonstrates a commendable level of accuracy when the prediction time interval does not exceed 5 seconds. However, as the prediction time interval continues to extend, significant discrepancies arise between the actual trajectories and the predicted trajectories. These disparities pose a significant challenge to trajectory prediction accuracy, making the method proposed in this paper more superior in short-term predictions.

To emphasize the superiority of the joint prediction model proposed in this paper compared to other prediction models, a series of comparative analyses were conducted. Specifically, the study employed SWPLS, UKF, LSTM, and the joint prediction algorithm proposed in this paper to predict complex UAV flight trajectories, followed by a detailed comparison and evaluation of their performance. In this research, a 0.5-second time interval was used for predicting UAV

TABLE 6. Comparison of identification results.

Identification model	Binary Tree SVM			SVM-BTS		
	PRC	REC	F1	PRC	REC	F1
Vertical motion	1.000	0.970	0.985	1.000	1.000	1.000
Pitch motion	0.947	1.000	0.973	1.000	1.000	1.000
Transverse motion	1.000	0.963	0.981	0.963	0.963	0.963
Roll motion	0.816	0.756	0.785	0.953	1.000	0.976
Transitional motion	1.000	0.760	0.864	1.000	0.920	0.958
AVG	0.953	0.890	0.917	<b>0.983</b>	<b>0.977</b>	<b>0.979</b>
Accuracy	87.5%			<b>97.9%</b>		

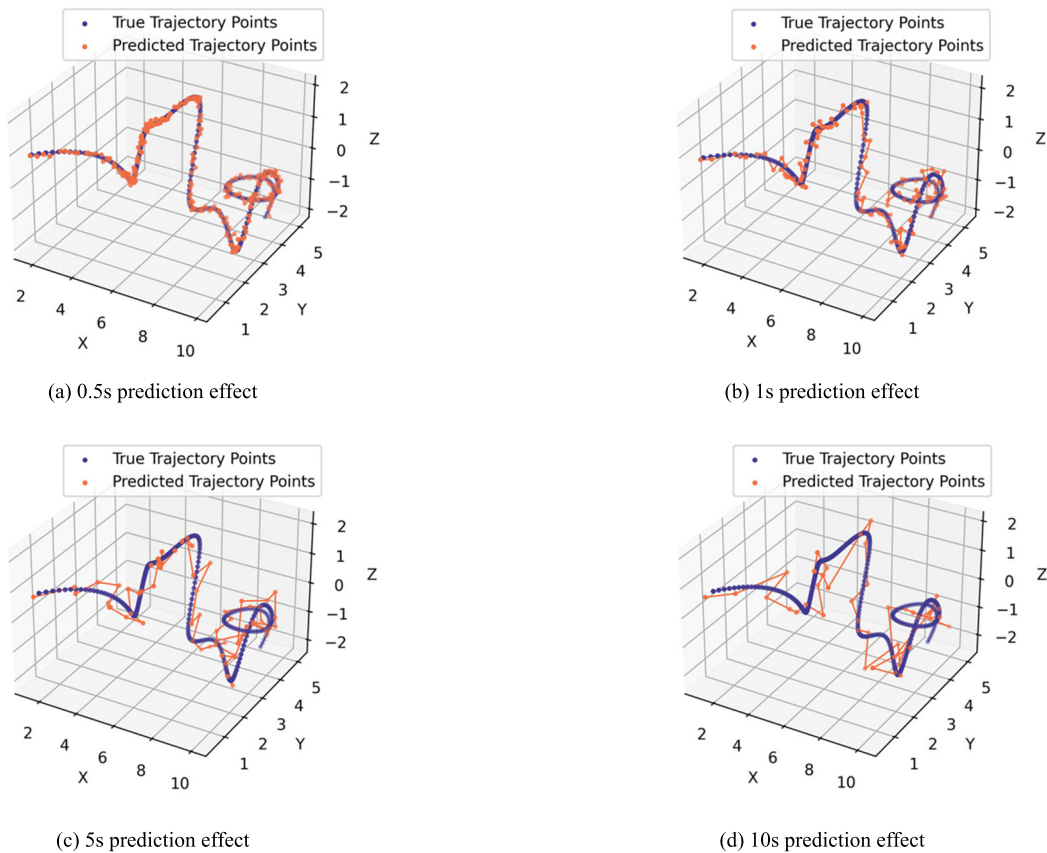


FIGURE 9. Prediction effect at 0.5s, 1s, 5s, 10s prediction time intervals.

trajectories, and four different UAV flight trajectories, along with their respective prediction results, were showcased. The specific predictive results are illustrated in Fig. 10 (a, b, c, d).

From Fig. 10 (a, b, c, d), it is visually apparent that the predicted trajectories generated by the joint prediction algorithm in this paper closely resemble the actual trajectories. Subsequently, for the four different datasets, the Mean Square Error (MSE) and average time delay for various models were calculated, as shown in Fig. 11 (a, b, c, d) and Table 7. The average time delay in this paper refers to the mean

time required to complete a prediction task within the next prediction time interval (0.5 seconds).

Table 7 indicates that the MSE for the complex trajectories predicted by the classification-based joint prediction algorithm proposed in this paper is 0.309, 0.322, 0.431, and 0.421 for the four distinct datasets, respectively. Compared to the prediction results of SWPLS, UKF, and LSTM, the errors are significantly reduced, demonstrating a notable improvement in the predictive performance of the model proposed in this paper. Additionally, the average time delays on the

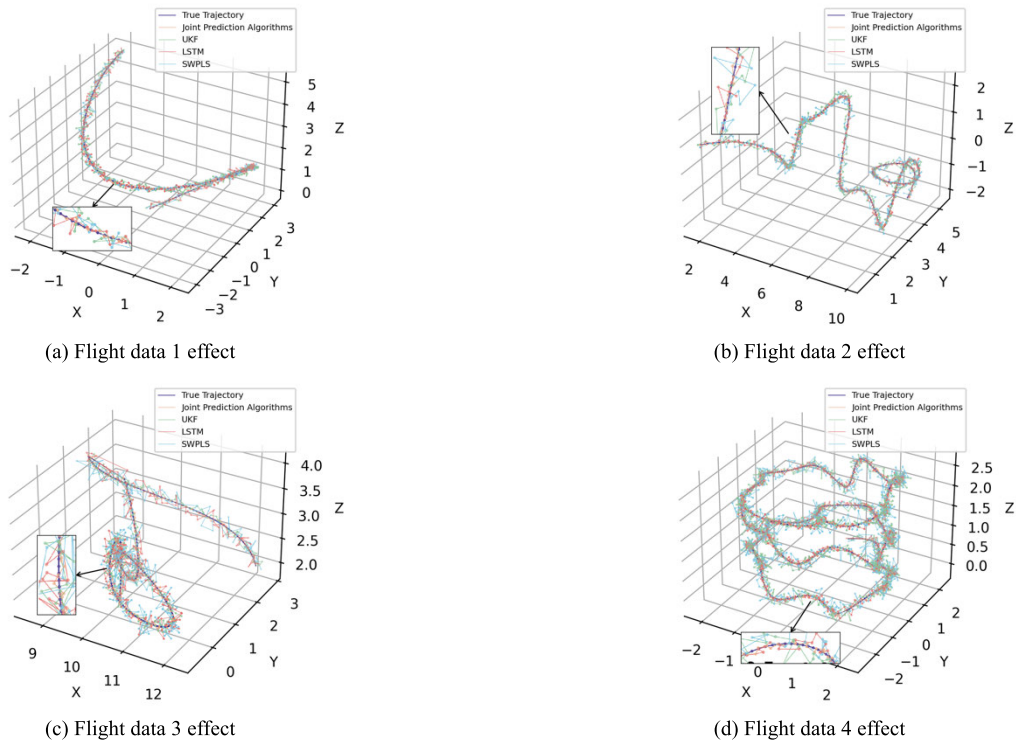


FIGURE 10. Predicted trajectories and true trajectories.

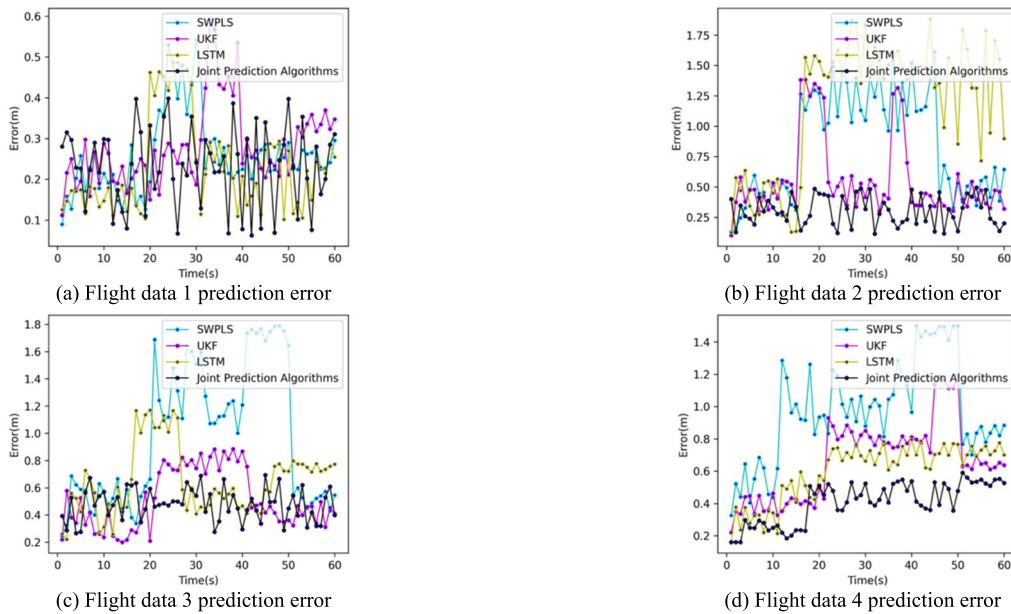


FIGURE 11. Comparison of prediction algorithm errors across different datasets.

four different flight datasets are 0.387s, 0.457s, 0.428s, and 0.466s, all of which are below the prediction time interval (0.5s) set in this paper. Therefore, the algorithm presented in this paper can generally meet the real-time prediction requirements. Although the average time delay of the joint prediction algorithm in this paper is not the lowest, it is lower

than that of UKF and LSTM. While SWPLS has the lowest average time delay, the MSE for this algorithm is greater than that of the joint prediction algorithm proposed in this paper for all four datasets. When considering both time delay and MSE, the trajectory prediction algorithm in this paper outperforms SWPLS, UKF, and LSTM.

TABLE 7. Comparison of prediction algorithm results.

Prediction model	Joint Prediction Algorithm		SWPLS		UKF		LSTM	
	MSE(m)	Average delay(s)	MSE(m)	Average delay(s)	MSE(m)	Average delay(s)	MSE(m)	Average delay(s)
Flight data 1	<b>0.309</b>	0.387	0.372	0.207	0.383	0.503	0.378	0.423
Flight data 2	<b>0.322</b>	0.457	1.025	0.265	0.862	0.624	1.284	0.767
Flight data 3	<b>0.431</b>	0.428	1.235	0.417	0.603	0.472	0.974	0.628
Flight data 4	<b>0.421</b>	0.466	0.983	0.467	0.879	0.678	0.512	0.823

## V. CONCLUSION

(1) Based on the experimental results of motion state identification, this paper proposes a SVM-BTS classification and identification method to realize the vertical motion, pitch motion, transverse motion, roll motion, and transitional motion states. This method performs two feature extractions on complex tracks. Compared with the traditional Binary Tree SVM, the accuracy is increased by 10%. The experimental results show the effectiveness of this method.

(2) From the trajectory prediction experiment, under the prediction time interval of 0.5s, the mean square error of prediction is 0.322m; when the prediction time interval exceeds 5s, the mean square error exceeds 4m. It can be seen that the joint prediction algorithm in this paper has great advantages for short-term trajectory prediction. Furthermore, while considering different complex trajectories, the proposed prediction approach in this paper exhibits the lowest mean square error compared to SWPLS, UKF, and LSTM. Crucially, real-time predictability can be ensured while preserving predictive reliability.

(3) The joint prediction algorithm proposed in this paper has a significant drop in prediction accuracy for prediction times exceeding 5 seconds. To make precise long-term predictions about the UAV, it is essential to have access to the UAV's flight intention data. In the future, the intent information of the UAV can be added, such as the UAV's flight plan. In addition, since small-sample data classification training is used, the generalization of the model needs to be improved. Future work will focus on large-sample data classification training for UAV flights to achieve better results.

This study addresses the issue of low accuracy in predicting complex UAV trajectories using a single prediction model and proposes a joint prediction algorithm. Firstly, different motion states in complex trajectories are classified and identified, and then SWPLS, UKF, and LSTM are respectively employed to predict each motion state. By evaluating the mean square error and average delay of the three prediction algorithms under different motion states, the most suitable prediction algorithm is selected for each motion state. The jointly proposed prediction algorithm is particularly applicable to civilian drones with complex trajectories. Through experiments comparing with other prediction algorithms, the

proposed algorithm in this study exhibits the lowest mean square error, significantly improves the accuracy of trajectory prediction, and demonstrates good average delay performance, meeting real-time prediction requirements.

## REFERENCES

- [1] K. Nonami, "Research and development of drone and roadmap to evolution," *J. Robot. Mechatronics*, vol. 30, no. 3, pp. 322–336, Jun. 2018.
- [2] S. Jayanthi, "Evolution and significance of unmanned aerial vehicles," *Unmanned Aerial Vehicle Cellular Commun.*, vol. 12, no. 2, pp. 287–311, Oct. 2022.
- [3] S. A. H. Mohsan, N. Q. H. Othman, Y. Li, M. H. Alsharif, and M. A. Khan, "Unmanned aerial vehicles (UAVs): Practical aspects, applications, open challenges, security issues, and future trends," *Intell. Service Robot.*, vol. 16, no. 1, pp. 109–137, Jan. 2023, doi: [10.1007/s11370-022-00452-4](https://doi.org/10.1007/s11370-022-00452-4).
- [4] A. Otto, N. Agatz, J. Campbell, B. Golden, and E. Pesch, "Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey," *Networks*, vol. 72, no. 4, pp. 411–458, Mar. 2018.
- [5] R. P. Sishodia, R. L. Ray, and S. K. Singh, "Applications of remote sensing in precision agriculture: A review," *Remote Sens.*, vol. 12, no. 19, p. 3136, Sep. 2020.
- [6] D. Sadykova, D. Pernebayeva, M. Bagheri, and A. James, "IN-YOLO: Real-time detection of outdoor high voltage insulators using UAV imaging," *IEEE Trans. Power Del.*, vol. 35, no. 3, pp. 1599–1601, Jun. 2020.
- [7] P. Nooralishahi, C. Ibarra-Castanedo, S. Deane, F. López, S. Pant, M. Genest, N. P. Avdelidis, and X. P. V. Maldague, "Drone-based non-destructive inspection of industrial sites: A review and case studies," *Drones*, vol. 5, no. 4, p. 106, Sep. 2021.
- [8] L. Page, "Drone trespass and the line separating the national airspace and private property," *Geo. Wash. L. Rev.*, vol. 86, no. 4, p. 1152, Jul. 2018.
- [9] C. Decker and P. Chiambaretto, "Economic policy choices and trade-offs for unmanned aircraft systems traffic management (UTM): Insights from Europe and the United States," *Transp. Res. A, Policy Pract.*, vol. 157, pp. 40–58, Mar. 2022, doi: [10.1016/j.tra.2022.01.006](https://doi.org/10.1016/j.tra.2022.01.006).
- [10] M. Huang, W. Y. Ochieng, J. J. E. Macias, and Y. Ding, "Accuracy evaluation of a new generic trajectory prediction model for unmanned aerial vehicles," *Aerosp. Sci. Technol.*, vol. 119, Dec. 2021, Art. no. 107160.
- [11] G. Xie and X. Chen, "Efficient and robust online trajectory prediction for non-cooperative unmanned aerial vehicles," *J. Aerosp. Inf. Syst.*, vol. 19, no. 2, pp. 143–153, Feb. 2022.
- [12] W. Zeng, X. Chu, Z. Xu, Y. Liu, and Z. Quan, "Aircraft 4D trajectory prediction in civil aviation: A review," *Aerospace*, vol. 9, no. 2, p. 91, Feb. 2022, doi: [10.3390/aerospace9020091](https://doi.org/10.3390/aerospace9020091).
- [13] Z. Wu, J. Li, J. Zuo, and S. Li, "Path planning of UAVs based on collision probability and Kalman filter," *IEEE Access*, vol. 6, pp. 34237–34245, 2018.
- [14] C. Sakthivel, B. Sureshkumar, R. Ramkumar, and R. Yokeswaran, "Detection and path prediction of aircraft based on acoustics and vibration," *Mater. Today, Proc.*, vol. 21, pp. 588–591, Sep. 2020.
- [15] Y. Lin, B. Yang, J. W. Zhang, and H. Liu, "Approach for 4D trajectory management based on HMM and trajectory similarity," *J. Mar. Sci. Technol.*, vol. 27, no. 3, pp. 246–256, May 2019.

- [16] X. Zhang and W. Yu, "Research on the application of Kalman filter algorithm in aircraft trajectory analysis," in *Proc. 7th Int. Conf. Intell. Comput. Signal Process. (ICSP)*, Apr. 2022, pp. 196–199.
- [17] L. Deng, D. Li, and R. Li, "Improved IMM algorithm based on RNNs," *J. Phys., Conf. Ser.*, vol. 1518, no. 1, Apr. 2020, Art. no. 012055.
- [18] R. Dalmau, M. Pérez-Batlle, and X. Prats, "Real-time identification of guidance modes in aircraft descents using surveillance data," in *Proc. IEEE/AIAA 37th Digit. Avionics Syst. Conf. (DASC)*, Sep. 2018, pp. 1–10.
- [19] H. Bin, P. Fang, and Q. Xiaoxia, "Research on trajectory prediction algorithm for low-altitude emergency rescue aircraft," in *J. Phys.: Conf. Ser.*, vol. 1510, Mar. 2020, Art. no. 012018.
- [20] X. Liu, W. He, J. Xie, and X. Chu, "Predicting the trajectories of vessels using machine learning," in *Proc. 5th Int. Conf. Control, Robot. Cybern. (CRC)*, Wuhan, China, Oct. 2020, pp. 66–70.
- [21] Z.-J. Wu, S. Tian, and L. Ma, "A 4D trajectory prediction model based on the BP neural network," *J. Intell. Syst.*, vol. 29, no. 1, pp. 1545–1557, Aug. 2019, doi: [10.1515/jisys-2019-0077](https://doi.org/10.1515/jisys-2019-0077).
- [22] Y. Pang and Y. Liu, "Probabilistic aircraft trajectory prediction considering weather uncertainties using dropout as Bayesian approximate variational inference," in *Proc. AIAA SciTech Forum*, Jan. 2020, p. 1413.
- [23] X. Zhang and S. Mahadevan, "Bayesian neural networks for flight trajectory prediction and safety assessment," *Decis. Support Syst.*, vol. 131, Apr. 2020, Art. no. 113246, doi: [10.1016/j.dss.2020.113246](https://doi.org/10.1016/j.dss.2020.113246).
- [24] J. Zhang, Y. Wu, and S. Jiao, "Research on trajectory tracking algorithm based on LSTM-UKF," in *Proc. 7th IEEE Int. Conf. Netw. Intell. Digit. Content (IC-NIDC)*, Beijing, China, Nov. 2021, pp. 61–65.
- [25] H.-C. Choi, C. Deng, and I. Hwang, "Hybrid machine learning and estimation-based flight trajectory prediction in terminal airspace," *IEEE Access*, vol. 9, pp. 151186–151197, 2021.
- [26] S. T. Kanneganti, P. B. Chilson, and R. Huck, "Visualization and prediction of aircraft trajectory using ADS-B," in *Proc. NAECON IEEE Nat. Aerosp. Electron. Conf.*, Dayton, OH, USA, Jul. 2018, pp. 529–532.
- [27] Z. Shi, M. Xu, Q. Pan, B. Yan, and H. Zhang, "LSTM-based flight trajectory prediction," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Rio de Janeiro, Brazil, Jul. 2018, pp. 1–8.
- [28] S. T. Barratt, M. J. Kochenderfer, and S. P. Boyd, "Learning probabilistic trajectory models of aircraft in terminal airspace from position data," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 9, pp. 3536–3545, Sep. 2019.
- [29] L. Ma and S. Tian, "A hybrid CNN-LSTM model for aircraft 4D trajectory prediction," *IEEE Access*, vol. 8, pp. 134668–134680, 2020.
- [30] B. Wang, D. Liu, W. Wang, and X. Peng, "A hybrid approach for UAV flight data estimation and prediction based on flight mode recognition," *Microelectron. Rel.*, vol. 84, pp. 253–262, May 2018.
- [31] S. J. Claessens, "Efficient transformation from Cartesian to geodetic coordinates," *Comput. Geosci.*, vol. 133, Dec. 2019, Art. no. 104307.
- [32] Rivlin and Theodore J, *Chebyshev Polynomials*. Mineola, NY, USA: Courier Dover, 2020, pp. 155–166.



**LIN BAO** is currently pursuing the master's degree with the School of Automation and Electrical Engineering, Zhejiang University of Science and Technology. Her research interests include pattern identification, data mining, machine learning, and deep learning.



**HUIHUI BAI** received the Ph.D. degree in control theory and control engineering from Xiamen University, in 2018. She is currently a Lecturer with the School of Automation and Electrical Engineering, Zhejiang University of Science and Technology. Her current research interests include robust control, nonlinear control, spacecraft attitude control, and aircraft control.



**HUIJIE NIU** is currently pursuing the master's degree with the School of Automation and Electrical Engineering, Zhejiang University of Science and Technology. His research interests include image processing, deep learning, and optimization algorithms.



**CHAO HAN** received the master's degree in engineering from the Zhejiang University of Science and Technology, in 2023. His research interests include data processing, classification, data mining, and deep learning.



**QINGCHUN JIAO** received the master's degree in information and communication engineering from Zhejiang University. He is currently a Professor with the Zhejiang University of Science and Technology. His research interests include security and prevention technology, intelligent building systems, and the development of smart cities. He is also a Standardization Expert with the Zhejiang Provincial Market Supervision Administration, an Expert with the Zhejiang Provincial Department of Science and Technology, and the Chairperson of the Standardization Committee within the Zhejiang Provincial Security Association.