Flight track pattern recognition based on few labeled data with outliers

Yuqi Fan,[a] **Guangming Shen**,[a] **Xiong Xu**,[b] **Juan Xu**,[a] and **Xiaohui Yuan**[c,*]

[a]Hefei University of Technology, School of Computer Science and Information Engineering, Hefei, Anhui, China

[b]State Key Laboratory of Complex Electromagnetic Environment Effects on Electronics and Information Systems, Luoyang, Henan, China

[c]University of North Texas, Department of Computer Science and Engineering, Denton, Texas, United States

Abstract. Gradually crowded, complex airspace makes it necessary to identify the flight track patterns of interested targets. Existing studies on radar-based target track recognition rarely consider the impact of outliers in the acquired data, which happens very often for small air vehicles such as drones. In addition, the performance achieved with a few labeled track examples has significant room for improvement. We propose a semisupervised target track recognition algorithm based on a semisupervised generative adversarial network (SSGAN) that learns a robust model from a few labeled target track examples with the presence of outliers. Our method identifies and eliminates the outliers in the data set and fills in for the removed data. The proposed method extracts a strong recognition flight feature from the basic flight features and forms the strong recognition flight feature combination (SRFFC) by integrating the advanced flight features. The SRFFC is fed into the SSGAN model to identify target track patterns. Experiments were conducted using simulated data sets. Our results demonstrate that the proposed method achieves a highly competitive target track recognition performance in terms of accuracy, precision, and recall in comparison with the state-of-the-art methods. The minimum accuracy of our proposed method is 97%, which achieves an improvement of 15.7% compared with the state-of-the-art methods. In addition, our method exhibits great robustness with respect to the number of labeled data and choice of parameters. © *2021 SPIE and IS&T* [DOI: 10.1117/1.JEI.30.3.031204]

Keywords: radar; target track recognition; outlier; few labeled data; generative adversarial network.

Paper 200530SS received Jul. 30, 2020; accepted for publication Nov. 23, 2020; published online Jan. 5, 2021.

1 Introduction

With the increasing popularity of unmanned aerial vehicles and the continuous development of the commercial airline industry, the airspace is becoming an increasingly crowded and complex environment.[1] According to the International Air Transport Association (IATA), the average number of registered aircraft flights per day was ∼10,000 in 2015, and this number will likely double by 2035.[2] Identifying and tracking potential threats of flight targets in the airspace is a great demand for the security and intelligence agencies. Anomalous flights could be potential hazards to other aircrafts in the airspace or, in the case of an extreme event, a hazard to thousands of civilians or military personnel.[3]

Although target tracking has been extensively studied and tracking objects on the ground and in the air processes a sequence of geolocations of the targets,[4] the differences among their movement characteristics are quite dramatic: (1) the movements of ground targets are usually constrained by physical environment restrictions, e.g., roads and traffic signals, whereas flight

*Address all correspondence to Xiaohui Yuan, xiaohui.yuan@unt.edu targets have greater freedom in movement; (2) the dynamic range of the moving speed differs greatly for flight and ground targets; and (3) the moving space of flight targets is much larger than that of ground targets. Therefore, the approaches for track recognition of ground targets cannot be directly applied to recognition of flight target track patterns.

A primary means of monitoring flight targets[5] is using active radar, which detects the distance, speed, flying direction, and other information of flight targets and enables flight target recognition based on radar data. There has been extensive research on radar-based flight target recognition, which mainly uses spectrum information in radar echoes, polarization characteristics, and high-resolution range profile (HRRP) to realize the recognition of target attributes and types. The track patterns imply the intention and mission of the target. For example, when the target is in an S maneuver, the target is likely avoiding some threat; when the target is circling in the air, the target could be performing surveillance and detection. The classification and recognition of target track patterns help with understanding the intention of flights and provide auxiliary decision-making information for airspace supervision. The radar data are prone to be distorted by noises such as recorder system errors, electromagnetic interference, random interference during the acquisition of track data, etc. The uncertainty increases for small flying objects, which results in outliers and hinders the analysis of track patterns.[6]

Another challenge faced in flight track pattern recognition is the lack of labeled examples despite the massive amount of radar target track data available for processing. It is expensive to annotate examples, which is a widely acknowledged issue faced in deep learning and motivates the study of semisupervised learning strategies to reduce the dependence on labeled data. The success of generative adversarial networks (GANs)[7] in the field of machine learning inspires researchers to increase training data by generating confidence maps of unlabeled data. In this paper, we propose a semisupervised target track recognition algorithm (SSTT) based on a semisupervised generation adversarial network (SSGAN) for few labeled radar target track data with outliers.

The main contributions of this paper are as follows. We investigate the problem of flight target track pattern recognition based on few labeled radar target track data with outliers and propose an efficient algorithm to recognize track patterns. Our proposed method introduces outlier recognition based on the improved Letts criterion (ORLC) to identify and eliminate the outliers and adopts missing point filling based on improved linear interpolation (MPFL). We define four basic flight features (BFFs) for the track data and select the strong recognition flight feature (SRFF) with the best track pattern recognition effect from the BFFs. Advanced flight features (AFFs) are extracted from BFFs and combined with the SRFF as a strong recognition flight feature combination (SRFFC) to the constructed SSGAN classification model. This enables accurate classification with few labeled radar data with outliers.

The rest of the paper is organized as follows. The related work is introduced in Sec. 2. The problem is defined in Sec. 3. We present our proposed algorithm in Sec. 4. In Sec. 5, we experimentally evaluate the proposed algorithm. Finally, we conclude the paper in Sec. 6.

2 Related Work

As an important tool of monitoring the status of flight targets, radar plays an irreplaceable role in the field of target recognition. Extensive research has been conducted on radar target recognition. Lee et al.[8] proposed a radar output recognition method based on the frequency-diversity radar cross section (RCS) and kernel scatter difference discrimination to reduce the space measurement time for solving the problem of low recognition accuracy when RCS contains random components. Ding et al.[9] designed a synthetic aperture radar (SAR) automatic target recognition method using an attributed scattering center matching approach that evaluates the global consistency and structural correlation between two attributed scattering center sets to provide a reliable and robust similarity measure for SAR automatic target recognition. Lee et al.[10] proposed a new target identification scheme for HRRP-based recognition that extracts scale-invariant features using the relaxation algorithm and the simple resampling process and constructs a nearest-neighbor classifier to determine the true target class. Chantasen et al.[11]

presented a radar target identification method of the coated object that uses poles extracted by a Cauchy method to determine the perfectly electrically conducting sphere. Guo et al.[12] adopted the least square support vector data description to alleviate the over-fitting of support vector data description (SVDD) for HRRP-based recognition. Karine et al.[13] proposed a SAR target recognition method based on the statistical model and the weighted sparse representation-based classification that uses the Kullback–Leibler divergence between the parametric statistical models of training and test sets to improve the recognition effect. However, these traditional radar target recognition methods are proposed for specific applications and are unable to be adapted to general and complex targets and environments.

In recent years, the exceptional performance of deep learning in the applications of image classification motivated researchers to apply deep networks to radar-based target recognition. Chen et al.[14] proposed a radar target recognition algorithm based on CNN to solve the recognition problem of limited training data through developing the convolution factor analysis probability generation model. Pan et al.[15] constructed a method of radar HRRP target recognition based on t-distribution stochastic neighbor embedding (t-SNE) and a depth belief network (DDBN) to solve the problem of imbalanced HRRP data by adopting t-SNE and synthetic sampling for preprocessing and constructing the DDBN for recognition. Xu et al.[16] designed a target-aware recurrent attentional network for HRRP radar target recognition to discover target regions through a recurrent neural network and attention mechanism. Although the aforementioned studies advanced our ability in radar target recognition, there is still room to improve in the recognition of track patterns. Note that the track of a radar target often implies the intention and mission of the target, and hence the recognition of track patterns is important to airspace supervision and threat assessments.

Methods have been developed for track recognition of ground targets. Choi et al.[17] combined a statistical hypothesis test and a random sampling method based on the renowned Monte-Carlo method to statistically categorize drivers for identifying the resting behavior pattern of drivers, which could be used to construct a proper resting strategy or a drowsy-driving avoidance strategy, resulting in fewer traffic accidents related to drowsy driving or loss of concentration while driving. Wu et al.[18] designed a vehicle-pedestrian near-crash identification method based on the tracks of vehicles and pedestrians extracted from roadside laser radar that was coded into an automatic procedure for near-crash identification.

The movement of flight targets is different from that of ground targets, which makes the existing target recognition approaches designed for ground track pattern recognition difficult to apply to the track recognition of flight targets. To our best knowledge, very few studies have been conducted on the radar target recognition of flight objects. Pan et al.[19] presented a sequential multifactor Hausdorff nearest-neighbor conformal multiclass classifier based on the conformal multiclass classifier and the multifactor nonconformity measure designed by the authors, which could online learn and classify flight target track patterns. Fan et al.[20] proposed a GAN-based approach to target track recognition with insufficient labeled data.

The outliers of radar data hinder the processing and analysis of track data such that track outlier detection plays an important role in data processing.[6] Deshmukh and Hwang[21] proposed a temporal-logic-based outlier detection algorithm for the air traffic surveillance data that can update the old models with the new overnight data to detect outliers in air traffic tracks. Habler and Shabtai[22] adopted the long short-term memory (LSTM) encoder–decoder algorithm to model ADS-B data and compared the simulated flight track generated by the LSTM model with the ADS-B data received to identify outliers. Puranik and Mavris[23] studied an outlier detection model composed of clustering and single-class classification algorithms that uses energy-based metrics as features to quickly and efficiently identify outliers in the approach and landing phase. However, the current research on track outlier detection fails to utilize the spatial position relationship between adjacent points in identifying outliers.

The existing research on track pattern recognition based on deep learning rarely considers the difficulty in labeling a massive amount of track data. To alleviate the negative impact of outliers and reduce the dependence on labeled data, we propose an SSTT recognition algorithm to achieve radar flight target track pattern identification based on few labeled radar data with outliers.

Table 1 Symbols and notations used in this paper.

Notation	Definition
L_m	Label of the *m*'th target track
$p_{i,m}$	The *i*'th point in the *m*'th target track
$l_{i,m}$, $\theta_{i,m}$, $\varphi_{i,m}$, $t_{i,m}$	The distance, elevation angle, azimuth, and time of point $p_{i,m}$, respectively
$k_{i,m}$	The coefficient of standard deviation of point $p_{i,m}$ during outlier identification
$fd_{i,m}$	Fluctuation degree of point $p_{i,m}$
$c(k_{i,m}, fd_{i,m})$	The correlation coefficient between $k_{i,m}$ and $fd_{i,m}$ for each point $p_{i,m}$
$x_{i,m}$, $y_{i,m}$, $z_{i,m}$, $t_{i,m}$	The *x*, *y*, *z* coordinates and the time stamp of point $p_{i,m}$, respectively
$x^f_{i,m}$, $y^f_{i,m}$, $z^f_{i,m}$	The *x*, *y*, *z* coordinates of the fitted point for point $p_{i,m}$
$x^d_{i,m}$, $y^d_{i,m}$, $z^d_{i,m}$	The absolute value of the difference between $(x^f_{i,m}, y^f_{i,m}, z^f_{i,m})$ and $(x_{i,m}, y_{i,m}, z_{i,m})$, i.e., $x^d_{i,m} = \lvert x^f_{i,m} - x_{i,m}\rvert$, $y^d_{i,m} = \lvert y^f_{i,m} - y_{i,m}\rvert$, and $z^d_{i,m} = \lvert z^f_{i,m} - z_{i,m}\rvert$, respectively
$\overline{x}^d_{i,m}$, $\overline{y}^d_{i,m}$, $\overline{z}^d_{i,m}$	The average of $x^d_{i,m}$, $y^d_{i,m}$, $z^d_{i,m}$, respectively.
$x^r_{i,m}$, $y^r_{i,m}$, $z^r_{i,m}$	The residue of $(x^d_{i,m}, y^d_{i,m}, z^d_{i,m})$ and the corresponding average value, i.e., $x^r_{i,m} = \lvert x^d_{i,m} - \overline{x}_{i,m}\rvert$, $y^r_{i,m} = \lvert y^d_{i,m} - \overline{y}_{i,m}\rvert$, and $z^d_{i,m} = \lvert z^d_{i,m} - \overline{z}_{i,m}\rvert$, respectively
$w_{i,m}$	The weight of interpolation point $p_{i,m}$ during the missing point fitting
FC'_a	The *a*'th fully connected layer in the generator G of SSGAN model
FC_a	The *a*'th fully connected layer of the discriminator D of SSGAN model

3 Problem Formulation

3.1 *Radar Target Point and Track*

The symbols and notations used in the paper are listed in Table 1. The target scanned by the radar corresponds to multiple points in the scanning area. Each point includes target information such as the position. Multiple points indicate the flying direction, speed, etc. A radar target track consists of multiple measurements (i.e., points), and a track can be regarded as a time-series of multiple radar target points that are sequential in time.

The position of a radar target track point is represented in the spherical coordinate system with the radar as the coordinate origin, as shown in Fig. 1. The radar is at the origin O of the coordinate system, and P denotes the target. The distance from P to O is l, and OL is the projection of OP on the XY coordinate plane. The elevation angle $\angle POL$ is denoted by θ, and the azimuth $\angle XOL$ is expressed as φ. The relative time when the target is located at the point is t. Assuming that there are M radar targets and each track includes N points, the i'th point of the m'th target is $p_{i,m}(l_{i,m}, \theta_{i,m}, \varphi_{i,m}, t_{i,m})$.

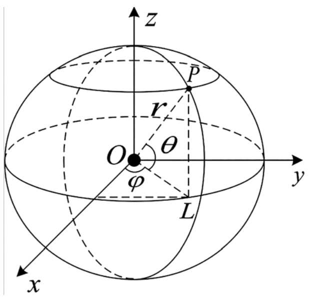

Fig. 1 Spherical coordinate system.

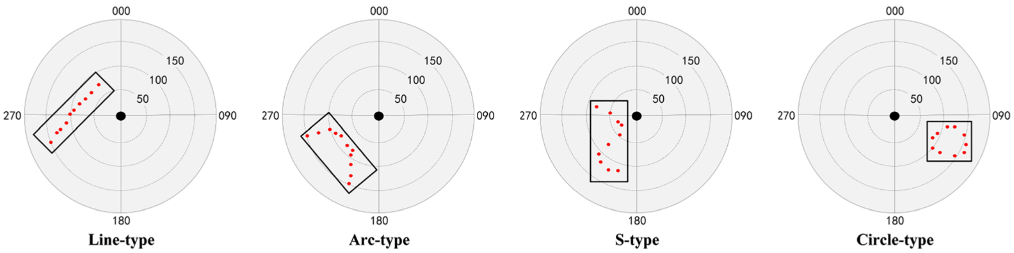

Fig. 2 Target track types.

3.2 *Radar Target Track Pattern Recognition*

Based on the analysis of a large number of track data, we classify the track patterns into four types: line-type, arc-type, S-type, and circle-type (encoded as 1, 2, 3, and 4, respectively) as shown in Fig. 2. A track m consists of N points, and the class label is as follows:

$$p_{1,m}, p_{2,m}, \dots , p_{i,m}, \dots p_{N,m}, L_m. \tag{1}$$

For track data with outliers, we deal with the outliers in the target track, extract track features, and construct an SSGAN model to classify the track pattern.

4 Flight Track Pattern Recognition Method

This section presents our proposed method, and Fig. 3 shows its overall flowchart. Our method consists of two modules: outlier processing and track pattern recognition. Outlier processing module: To reduce the negative impact of outliers in the original target track data on target track recognition, ORLC is introduced to recognize and eliminate the outliers in the target tracks. MPFL is adopted to fill the missing points caused by the outlier removal. Track pattern recognition module: To reduce the dependence on the labeled data, our method extracts four BFFs from the target track data and selects the SRFF with the best target track recognition effect from the BFFs. An AFF from the BFFs is extracted to improve the robustness of recognition, which is combined with the SRFF to obtain the SRFFC. The SRFFC is input into the constructed SSGAN model to recognize target track patterns.

4.1 *Outlier Processing*

Outlier processing consists of two stages: identifying and eliminating the outliers in the target tracks based on ORLC and filling the missing points after removing the outliers based on MPFL.

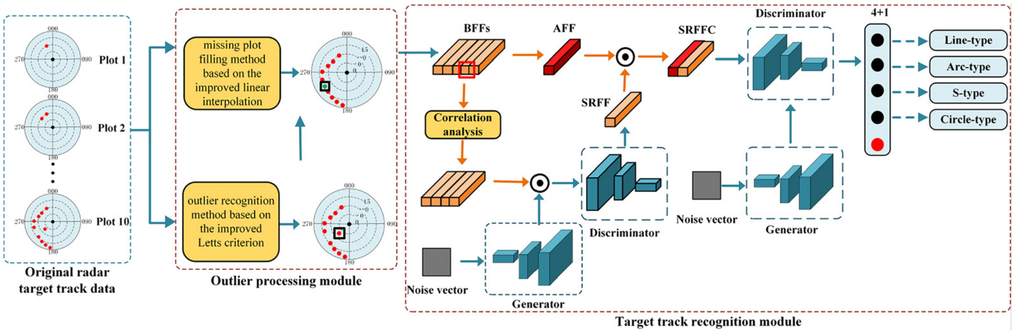
Plot 1
Plot 2
Plot 10
Original radar target track data
missing plot filling method based on the improved linear interpolation
outlier recognition method based on the improved Letts criterion
Outlier processing module
BFFs
AFF
SRFFC
Discriminator
4+1
Line-type
Arc-type
S-type
Circle-type
SRFF
Correlation analysis
Discriminator
Noise vector
Generator
Noise vector
Generator
Target track recognition module

Fig. 3 Diagram of algorithm SSTT.

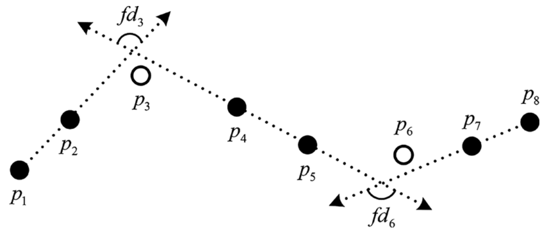

Fig. 4 Fluctuation degree.

4.1.1 *Stage 1: Outlier recognition based on improved Letts criterion*

According to Letts criterion,[24] when the target data follow a normal distribution, the probability that the residues between the target data and the arithmetic average fall within three times the standard deviation, i.e., $[-3\sigma, 3\sigma]$, is more than 99.7%. We regard the target data that fall outside the range as outliers. However, this probability varies for each data point in the original radar target track data with outliers. If the Letts criterion is used directly to identify outliers, we cannot recognize the outliers effectively. Therefore, for each point $p_{i,m}$, we set a range $[-k_{i,m} \cdot \sigma, k_{i,m} \cdot \sigma]$. If point $p_{i,m}$ falls within $[-k_{i,m} \cdot \sigma, k_{i,m} \cdot \sigma]$, $p_{i,m}$ is a normal point; otherwise, $p_{i,m}$ is an outlier.

Note that the spatial position of a normal track point $p_{i,m}$ is closely related to the angle between the lines formed by $p_{i-1,m}$, $p_{i-2,m}$ before point $p_{i,m}$ and $p_{i+1,m}$, $p_{i+2,m}$ after point $p_{i,m}$, as shown in Fig. 4. When the angle is large, the fluctuation of the track segment where point $p_{i,m}$ is located is small, and the fluctuation amplitude of point $p_{i,m}$ is smaller than that of the adjacent points. By contrast, when the angle is small, the track segment where point $p_{i,m}$ is located is large, and the fluctuation amplitude of point $p_{i,m}$ should be larger than that of the adjacent points. Therefore, we define fluctuation degree fd_i at point $p_{i,m}$ as the angle between the lines formed by the two points before point $p_{i,m}$ and the two points after point $p_{i,m}$ in time sequence, which is used to adjust $k_{i,m}$ at each track point $p_{i,m}$. When fd_i is large, we decrease $k_{i,m}$. When fd_i is small, we increase $k_{i,m}$. In particular, we set the relationship between $k_{i,m}$ and fd_i as $k_{i,m} = c(k_{i,m}, fd_{i,m})/fd_i$, where $c(k_{i,m}, fd_{i,m})$ is the correlation coefficient between $k_{i,m}$ and fd_i for each point $p_{i,m}$.

The input to algorithm ORLC includes the target track data containing N points $\{p_{1,m}, p_{2,m}, \dots, p_{N,m}\}$, set $\{k_{1,m}, k_{2,m}, \dots, k_{N,m}\}$, and set $\{fd_{1,m}, fd_{2,m}, \dots, fd_{N,m}\}$. We set $k_{1,m}$, $k_{2,m}$, $k_{N-1,m}$, and $k_{N,m}$ as the default value 3 of the Letts criterion. The ORLC algorithm is as follows (Algorithm 1):

Since the outliers are included in the calculation of the data standard deviation, we perform multiple iterations during the outlier identification, that is, the residue and standard deviation should be recalculated based on the data after the previous outlier recognition. Finally, algorithm ORLC outputs the radar track data after the outlier recognition and elimination.

4.1.2 *Stage 2: Missing point filling based on the improved linear interpolation*

The recognition and elimination of outliers lead to missing points at the corresponding positions, and we need to fill these missing points. As an interpolation method commonly used for data filling, linear interpolation can obtain good filling performance for linear track. However, when the point that needs to be interpolated is in the position where the direction of the track changes greatly, it is difficult to fit an accurate point using linear interpolation. Therefore, we propose an MPFL algorithm for missing point filling. While keeping the characteristics of the linear interpolation, algorithm MPFL considers the relationship between the fluctuation degree of the missing point and the predicted coordinate value, thereby improving the performance of data filling for the nonlinear track.

The input of algorithm MPFL (Algorithm 2) is the track containing N points $\{p_{1,m}(x_{1,m}, y_{1,m}, z_{1,m}, t_{1,m}), p_{2,m}(x_{2,m}, y_{2,m}, z_{2,m}, t_{2,m}), \dots, p_{N,m}(x_{N,m}, y_{N,m}, z_{N,m}, t_{N,m})\}$ after the outliers' removal.

Algorithm 1 Outlier recognition based on the improved Letts criterion.

Step 1: We use Eq. (2) to convert every track point $p_{i,m}(l_{i,m}, \theta_{i,m}, \varphi_{i,m}, t_{i,m})$ to the rectangular coordinate system in space, with the radar as the origin, the positive east direction as the x axis, the true north direction as the y axis, and the vertical upward direction as the z axis, to obtain $p_{i,m}(x_{i,m}, y_{i,m}, z_{i,m}, t_{i,m})$

$$\begin{cases} x_{i,m} = \sqrt{l_{i,m}^2 - \varphi_i^2} \cdot \cos\,\theta_{i,m} \\ y_{i,m} = \sqrt{l_{i,m}^2 - \varphi_i^2} \cdot \sin\,\theta_{i,m} \\ z_{i,m} = \varphi_{i,m} \end{cases}. \tag{2}$$

Step 2: To make the difference between the outliers and the normal points significant, we use the least square fitting[25] on the corresponding x axis coordinate value set $X_m = \{x_{1,m}, x_{2,m}, \dots, x_{N,m}\}$, y axis coordinate value set $Y_m = \{y_{1,m}, y_{2,m}, \dots, y_{N,m}\}$, and z axis coordinate value set $Z_m = \{z_{1,m}, z_{2,m}, \dots, z_{N,m}\}$ to obtain the corresponding fitting data set $X_m^f = \{x_{1,m}^f, x_{2,m}^f, \dots, x_{N,m}^f\}$, $Y_m^f = \{y_{1,m}^f, y_{2,m}^f, \dots, y_{N,m}^f\}$, and $Z_m^f = \{z_{1,m}^f, z_{2,m}^f, \dots, z_{N,m}^f\}$.

Step 3: We calculate the absolute value of the difference between the original data and the fitting data to obtain $X_m^d = \{x_{1,m}^d, x_{2,m}^d, \cdots, x_{N,m}^d\}$, $Y_m^d = \{y_{1,m}^d, y_{2,m}^d, \cdots, y_{N,m}^d\}$, and $Z_m^d = \{z_{1,m}^d, z_{2,m}^d, \cdots, z_{N,m}^d\}$, which are used as the target data for outlier recognition. For example, for each $x_{i,m}^d \in X_m^d$, $x_{i,m}^d = |x_{i,m} - x_{i,m}^f|$.

Step 4: The Letts criterion requires that the target data follow the normal distribution, and the Kolmogorov–Smirnov test (K-S test)[26] is often adopted to verify whether the data conform to the normal distribution. Therefore, we adopt the K-S test to verify the normal distribution of the target data before using the Letts criterion. For the data that do not follow the normal distribution, we apply logarithmic processing to the data to make the target data conform to the normal distribution.

Step 5: We calculate the average of the data in sets X_m^d, Y_m^d, and Z_m^d, respectively, to get $\overline{x}_m^d$, $\overline{y}_m^d$, and $\overline{z}_m^d$. For example, $\overline{x}_m^d$ is calculated with Eq. (3).

$$\overline{x}_m^d = \frac{1}{N}\sum_{i=1}^{N} x_{i,m}^d. \tag{3}$$

Step 6: We calculate residue sets $X_m^r = \{x_{1,m}^r, x_{2,m}^r, \cdots, x_{N,m}^r\}$, $Y_m^r = \{y_{1,m}^r, y_{2,m}^r, \dots, y_{N,m}^r\}$, and $Z_m^r = \{z_{1,m}^r, z_{2,m}^r, \dots, z_{N,m}^r\}$ for sets X_m^d, Y_m^d, and Z_m^d, respectively. For example, for every $x_{i,m}^r \in X_m^r$, $x_{i,m}^r = |x_{i,m}^d - \overline{x}_m^d|$.

Step 7: We calculate the standard deviation σ_x, σ_y, and σ_z for the data in sets X_m^d, Y_m^d, and Z_m^d, respectively.

Step 8: For each point $p_{i,m}$, we evaluate whether each of the i'th elements in sets X_m^r, Y_m^r, and Z_m^r falls within $[-k_{i,m} \cdot \sigma_x, k_{i,m} \cdot \sigma_x]$, $[-k_{i,m} \cdot \sigma_y, k_{i,m} \cdot \sigma_y]$, and $[-k_{i,m} \cdot \sigma_z, k_{i,m} \cdot \sigma_z]$, respectively. If yes for all three i'th elements, point $p_{i,m}$ is a normal data point; otherwise, point $p_{i,m}$ is an outlier, and we set the coordinate of $p_{i,m}$ as $(0,0,0,t_{i,m})$.

Step 9: We return $\{p_{1,m}(x_{1,m}, y_{1,m}, z_{1,m}, t_{1,m}), p_{2,m}(x_{2,m}, y_{2,m}, z_{2,m}, t_{2,m}), \dots, p_{N,m}(x_{N,m}, y_{N,m}, z_{N,m}, t_{N,m})\}$, where the x axis, y axis, and z axis coordinate values of the outliers are set as 0.

4.2 *Track Pattern Recognition*

The track pattern classification consists of five stages: (1) we extract four BFFs from the radar data for track pattern classification; (2) we select the SRFF which contributes the most to the target track classification from the BFFs using the SSGAN model; (3) we extract an AFF from the BFFs to improve the robustness of target track recognition; (4) we combine the SRFF and AFF to get the SRFFC, and we use the SRFFC as the input of the SSGAN model for track type classification; and (5) we construct the SSGAN model to achieve identification of track patterns using few labeled radar target track data. The detailed description of each stage is as follows.

4.2.1 *Stage 1: extraction of the BFFs*

The radar data contain several flight features such as speed, acceleration, track angle, etc. When the target is in flight, some features change, which differ significantly under different types of

Algorithm 2 Missing point padding based on the improved linear interpolation.

Step 1: Traverse each point $p_{i,m} \in T$ to find the missing points. Assuming that the adjacent points of the missing point $p_{i,m}$ in time sequence are $p_{i-1,m}$ and $p_{i+1,m}$, respectively. We calculate $p'_{i-1,m}$ and $p'_{i+1,m}$ via Eqs. (4) and (5):

$$\begin{cases} x'_{i-1,m} = x_{i-1,m} + v^{x}_{i-1,m} \cos \mu^{x}_{i-1,m}(t_{i,m} - t_{i-1,m}) \\ y'_{i-1,m} = y_{i-1,m} + v^{x}_{i-1,m} \sin \mu^{x}_{i-1,m}(t_{i,m} - t_{i-1,m}), \\ z'_{i-1,m} = z_{i-1,m} + v^{y}_{i-1,m}(t_{i,m} - t_{i-1,m}) \end{cases} \tag{4}$$

$$\begin{cases} x'_{i+1,m} = x_{i+1,m} + v^{x}_{i+1,m} \cos \mu^{x}_{i+1,m}(t_{i+1,m} - t_{i,m}) \\ y'_{i+1,m} = y_{i+1,m} + v^{x}_{i+1,m} \sin \mu^{x}_{i+1,m}(t_{i+1,m} - t_{i,m}) \; , \\ z'_{i+1,m} = z_{i+1,m} + v^{y}_{i+1,m}(t_{i+1,m} - t_{i,m}) \end{cases} \tag{5}$$

where $v^{x}_{i-1,m}$ and $v^{x}_{i+1,m}$ are the horizontal velocities at points $p_{i-1,m}$ and $p_{i+1,m}$, respectively; $v^{y}_{i-1,m}$ and $v^{y}_{i+1,m}$ are the vertical velocities at points $p_{i-1,m}$ and $p_{i+1,m}$, respectively; $t_{i,m}$ is the time at the interpolation point $p_{i,m}$; and $\mu^{x}_{i-1,m}$ and $\mu^{x}_{i+1,m}$ are the horizontal courses at points $p_{i-1,m}$ and $p_{i+1,m}$, respectively.

Step 2: We calculate the coordinates of $p^{A}_{i,m}$ as the weighted sum of the coordinates of $p'_{i-1,m}$ and $p'_{i+1,m}$, i.e., $x^{A}_{i,m} = w'_{i-1,m}x'_{i-1,m} + w'_{i+1,m}x'_{i+1,m}$, $y^{A}_{i,m} = w'_{i-1,m}y'_{i-1,m} + w'_{i+1,m}y'_{i+1,m}$, and $z^{A}_{i,m} = w'_{i-1,m}z'_{i-1,m} + w'_{i+1,m}z'_{i+1,m}$, where the weights of $p'_{i-1,m}$ and $p'_{i+1,m}$ are $w'_{i-1,m} = 1 - (t_{i,m} - t_{i-1,m})/(t_{i+1,m} - t_{i-1,m})$ and $w'_{i+1,m} = 1 - (t_{i+1,m} - t_{i,m})/(t_{i+1,m} - t_{i-1,m})$, respectively.

Step 3: Adopt linear interpolation to calculate $p^{A'}_{i,m}(x^{A'}_{i,m}, y^{A'}_{i,m}, z^{A'}_{i,m}, t_{i,m})$.

Step 4: To make the track data after missing point filling as consistent with the actual target flight state as possible, we take the weighted sum of $p^{A}_{i,m}$ and $p^{A'}_{i,m}$ as the interpolation point $p_{i,m}(x^{I}_{i,m}, y^{I}_{i,m}, z^{I}_{i,m}, t_{i,m})$. To be specific, $x^{I}_{i,m} = w^{A}_{i,m}x^{A}_{i,m} + w^{A'}_{i,m}x^{A'}_{i,m}$, $y^{I}_{i,m} = w^{A}_{i,m}y^{A}_{i,m} + w^{A'}_{i,m}y^{A'}_{i,m}$, and $z^{I}_{i,m} = w^{A}_{i,m}z^{A}_{i,m} + w^{A'}_{i,m}z^{A'}_{i,m}$. The weights $w^{A}_{i,m}$ and $w^{A'}_{i,m}$ of $p^{A}_{i,m}$ and $p^{A'}_{i,m}$ are related to the $fd_{i,m}$ of point $p_{i,m}$. When $fd_{i,m}$ is large, the fluctuation of the track segment where $p_{i,m}$ is located is small, and $w^{A}_{i,m}$ should be appropriately reduced, while $w^{A'}_{i,m}$ should be appropriately increased. When $fd_{i,m}$ is small, the fluctuation of the track segment where $p_{i,m}$ is located is large, $w^{A}_{i,m}$ should be appropriately increased, and $w^{A'}_{i,m}$ should be properly reduced. We set the relationship among $w^{A}_{i,m}$, $w^{A'}_{i,m}$, and $fd_{i,m}$ as $w^{A}_{i,m}/w^{A'}_{i,m} = c(w^{A}_{i,m}, w^{A'}_{i,m}, fd_{i,m})/fd_{i,m}$ for each point $p_{i,m}$, where the interpolation point weight correlation coefficient $c(w^{A}_{i,m}, w^{A'}_{i,m}, fd_{i,m})$ determines the relationship among $w^{A}_{i,m}$, $w^{A'}_{i,m}$, and $fd_{i,m}$.

Step 5: Fill the next missing point until all missing points in T are filled.

Step 6: Return the target track data after missing point filling.

target tracks. We use the least square method[25] to fit the track and extract five BFFs from the target track data, including distance (spatial distance between adjacent points), velocity, acceleration, track angle (the angle between longitude and the tangent of the track where the point is located), and azimuth.

The different correlations between BFFs may lead to feature redundancy which negatively impacts the track pattern recognition. To improve the independence of BFFs, we perform a correlation analysis on BFFs to evaluate the relationship between the BFFs, thereby removing the BFFs with high correlation. Spearman rank correlation coefficients are often used to measure the degree of correlation between the data without making any assumptions about the data distribution. We use Spearman rank correlation coefficients to measure the correlation between BFFs. The high correlation between the BFFs indicates the existence of redundant features that should be eliminated.

Table 2 illustrates the Spearman rank correlation coefficients between different pairs of BFFs, which shows that there are some correlations among the features. Specifically, there is a strong correlation between distance and velocity, which indicates that distance and velocity overlap with each other. Therefore, we remove distance from BFFs, and hence BFFs only include velocity, acceleration, track angle, and azimuth.

Table 2 Spearman rank correlation coefficients between different feature pairs.

Feature pair	Line-type	Arc-type	S-type	Circle-type
Distance-velocity	1.000	1.000	1.000	1.000
Distance-acceleration	0.714	0.714	−0.524	0.667
Distance-trackangle	0.381	−0.381	0.429	−0.333
Distance-direction	1.000	0.238	0.429	−0.048
Velocity-acceleration	0.714	0.714	−0.524	0.667
Velocity-trackangle	0.381	−0.381	0.429	−0.333
Velocity-direction	1.000	0.238	0.429	−0.048
Acceleration-trackangle	−0.024	−0.286	−0.929	−0.048
Acceleration-direction	0.714	−0.048	−0.929	0.000
Trackangle-direction	0.381	0.214	1.000	0.429

BFFs may differ greatly in the dynamic range. Therefore, we perform normalization on the BFFs to balance the contribution of the features. Assuming that the data that need to be normalized are $d_1, d_2, \dots, d_N$, the normalization is as follows:

$$d_i' = \frac{d_i - \overline{d}}{\sigma}, \quad i = 1,2, \dots, N, \tag{6}$$

where $\overline{d}$ and σ are the arithmetic average and the standard deviation of the data, respectively, and d_i' is normalized d_i.

4.2.2 *Stage 2: selection of the SRFF*

The recognition accuracy varies with different BFFs. The BFF with the highest recognition accuracy contains the most effective target track recognition information. Therefore, we define the BFF which contains the most effective identification information as the SRFF. By selecting SRFF from all of the BFFs, we obtain the BFF that contributes the most to the track pattern recognition.

We use the data after correlation analysis and normalization as the input of the SSGAN classification model and select the BFF with the highest target track recognition accuracy as the SRFF. As shown in Table 3, the track angle for target track recognition obtains the highest recognition accuracy among the four BFFs, indicating that the track angle contains the most effective identification information. Therefore, we select the track angle as the SRFF.

4.2.3 *Stage 3: extraction of the AFF*

Normally, the track angle of a target during a straight flight changes very little, so the fluctuation of the track angle of adjacent points is small and close to 0. When the target is performing an S maneuver, the track angle of adjacent points will show periodic fluctuations. When the track is of circle-type, the track angles of the adjacent points are constantly changing, but the variation remains mostly stable, as shown in Fig. 5. The variation of the track angles of adjacent points

Table 3 Comparison of recognition accuracy of different BFFs.

BFFs	Velocity	Acceleration	Track angle	Direction
Accuracy	94.4%	85.2%	94.8%	73.9%

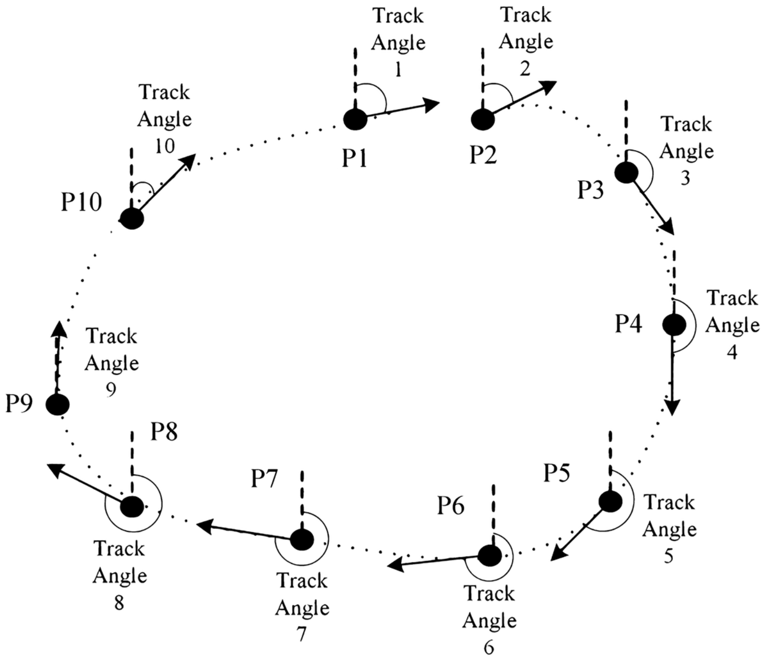

Fig. 5 Change of track angle under a circle-type track.

usually increases first and then decreases when the target track is of arc-type. We extract the track angles at the points and use the fluctuation residues of the track angles of adjacent points (TF) as the AFF.

4.2.4 *Stage 4: combination of SRFF and AFF*

We combine SRFF and AFF as the SRFFC. We take the SRFFC as the input of the SSGAN model for the target track recognition.

4.2.5 *Stage 5: construction of the SSGAN model*

To overcome the laborious labeling process of massive track data, we devise semisupervised learning that leverages GAN to learn from a few labeled data. GAN is an unsupervised generative model based on the game theory, which consists of two networks contest with each other in a game: generator (G) and discriminator (D). When constructing the SSGAN model, we need to make some changes based on GAN. The original GAN model uses unlabeled samples during training to generate samples without category information. The SSGAN uses labeled and unlabeled samples to jointly train and generate samples with category information. Therefore, we replace the discriminator D with a multiclassifier. The "fake" samples generated by the generator G through random noise are input into the discriminator D with the labeled samples and the unlabeled samples, and the discriminator D outputs $(n+1)$-dimensional classification results. Each dimension of the former n-dimensional output is the confidence of the corresponding class, and the $(n+1)$'th dimension represents the confidence of being classified as "fake."

The constructed SSGAN model is shown in Fig. 6. In each epoch, the noise vector is input to the generator G and continuously mapped to a high-dimensional vector which is consistent with the size of the real data through a four-layer fully connected layer. We add a batch normalization layer (BN) after FC_1' and FC_2' to improve the convergence and stability of the model. Meanwhile, the network structure of the discriminator D is implemented with a four-layer fully connected neural network. After receiving fake data and real data as input, the discriminator maps the input data through FC_1 and FC_2 to a 100×500-dimensional parameter space so the network can simulate the characteristics of the data distribution. The dimension is reduced through full connection layers FC_3 and FC_4. Finally, the discriminator D outputs the probability

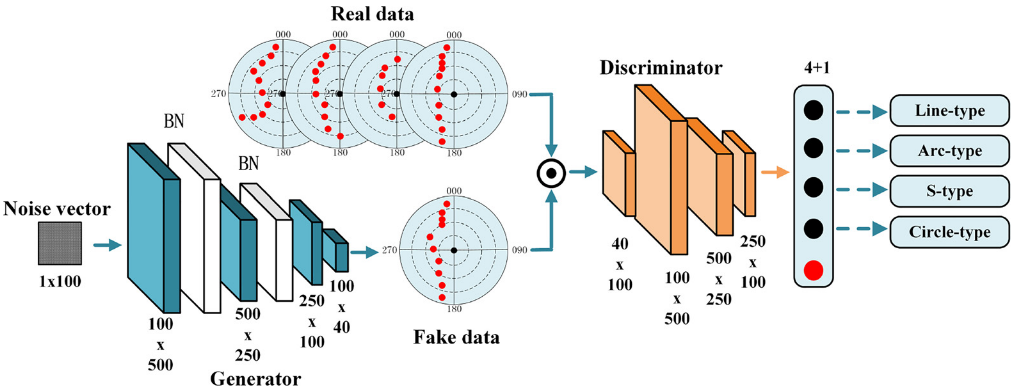

Fig. 6 The construction of SSGAN model.

distribution of each target track pattern. In particular, we initialize the parameters randomly according to the standard normal distribution during parameter initialization and perform normalization for each layer of the parameters in the model as follows:

$$\hat{\omega}_{ab} = \frac{\omega_{ab}}{\sqrt{\sum_{b=1}^{N_a} \omega_{ab}^2}}, \tag{7}$$

where ω_{ab} is the b'th model parameter in the a'th fully connected layer; N_a is the number of neurons in the a'th fully connected layer; and $\hat{\omega}_{ab}$ is the updated model parameter ω_{ab} after normalization. With normalization, we can make the dynamic floating ranges of parameters in the discriminator model consistent, thereby avoiding using different learning rates for different parameters due to the different dynamic ranges of parameters, which enables fast convergence during training.

5 Performance Evaluation

5.1 *Experiment Settings*

We generated synthetic radar returns of multiple targets following methods in Ref. 27, and examples of the simulated data are shown in Table 4. The data set includes four types of track patterns: line-type, arc-type, S-type, and circle-type. We created 5000 track sequences for each track pattern. Each radar track sequence lasts about 10 s, and the time interval between two adjacent points is 1 s, that is, each track sequence is composed of 10 consecutive points. Each point has four attributes: distance l, elevation θ, azimuth φ, and time t. The range of distances is [0 m, 100,000 m] (m denotes meter), the range of elevation is [0°, 90°], the range of azimuths is [0°, 360°], and the value of the time attribute is within [1000 s, 20,000 s] (s denotes second). Each track contains one to three outliers that are randomly perturbed.

Table 4 Schematic diagram of the original radar target track data.

track m	$l_{1,m}$	$\theta_{1,m}$	$\varphi_{1,m}$	$t_{1,m}$	$l_{2,m}$	$\theta_{2,m}$	$\varphi_{2,m}$	$t_{2,m}$		Label
1	1694.61	26.50	129.71	1113	1206.36	40.69	138.25	1114	…	Line-type
2	53,652.2	1.84	52.34	5582	53,576.68	1.80	56.41	5583	…	Arc-type
3	5133.54	14.99	240.81	7625	5414.66	14.23	229.46	7626	…	S-type
4	705.81	32.98	1.29	9432	754.54	33.42	29.27	9433	…	Circle-type
…	…	…	…	…	…	…	…	…	…	…

We split the dataset into two parts: a training set and a testing set. Specifically, we select 80% of each type of track data from the dataset by random sampling for training and use the remaining 20% of each type of track data as the test set[28] to evaluate the performance of algorithm SSTT. To make the simulation results robust to randomness, we randomly select the training set 30 times and take the average of the testing results as the final experimental result.

The parameters of algorithm SSTT are set as follows: $c(w^{A}_{i,m}, w^{A'}_{i,m}, fd_{i,m})$ determines the relationship among $w^{A}_{i,m}$, $w^{A'}_{i,m}$, and $fd_{i,m}$, and we set the default value of $c(w^{A}_{i,m}, w^{A'}_{i,m}, fd_{i,m})$ as 1; both discriminator D and generator G of the SSGAN model use Adam optimizer,[29] where the optimizer parameters β_1 and β_2 are set as 0.9 and 0.99, respectively; the learning rate is initialized as 0.001, the batch size is set as 100, and the number of epochs is 200.

We use three performance metrics commonly used in deep learning classification tasks to evaluate the performance of algorithm SSTT: accuracy, precision, and recall. Accuracy is the ratio of the correctly classified samples to the total samples and is calculated with Eq. (8), where TP is the amount of true positive samples, FP is the amount of false positive samples, FN is the amount of false negative samples, and TN is the amount of true negative samples:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{8}$$

Precision calculated with Eq. (9) is the proportion of all classified positive samples that are positive, and the value range of precision is [0, 1]:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{9}$$

Recall calculated via Eq. (10) is the proportion of positive samples that are classified as positive, and the value range of recall is [0, 1]:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{10}$$

5.2 *Experimental Results*

5.2.1 *Impact of the correlation between $k_{i,m}$ and $fd_{i,m}$*

We investigate the impact of the correlation coefficient between $k_{i,m}$ and $fd_{i,m}$ for each point $p_{i,m}$, denoted with $c(k_{i,m}, fd_{i,m})$, on the accuracy of track pattern recognition. Table 5 presents the results with the correlation coefficient ranging from 1 to 9. As the correlation coefficient increases up to four, the accuracy increases and reaches the maximum of 97.3%. As the correlation coefficient continues to increase, the accuracy reduces. When $c(k_{i,m}, fd_{i,m}) \geq 8$, the outlier recognition standard is too loose to identify some of the outliers, while the unrecognized outliers degrade the accuracy significantly. For example, some outliers with gentle fluctuations were not recognized. The accuracy reaches the maximum when $c(k_{i,m}, fd_{i,m})$ is 4, which means that the outlier processing achieves a good balance between large fluctuation degree and small fluctuation degree. In the remaining experiments, we set $c(k_{i,m}, fd_{i,m})$ as 4.

Table 5 Recognition accuracy with different $c(k_{i,m}, fd_{i,m})$.

$c(k_{i,m}, fd_{i,m})$	1	2	3	4	5	6	7	8	9
Accuracy	85.1%	92.3%	96.7%	97.3%	97.1%	93.2%	94.4%	89.8%	82.4%

Table 6 Recognition accuracy with respect to different $c(w_{i,m}^{A}, w_{i,m}^{A'}, fd_{i,m})$.

$c(w_{i,m}^{A}, w_{i,m}^{A'}, fd_{i,m})$	1	2	3	4	5
Accuracy	91.6%	95.8%	97.5%	96.4%	90.5%

5.2.2 *Impact of the correlation coefficient between $w_{i,m}^{A}$, $w_{i,m}^{A'}$, and $fd_{i,m}$*

We study the impact of different values of $c(w_{i,m}^{A}, w_{i,m}^{A'}, fd_{i,m})$, the correlation coefficient among $w_{i,m}^{A}$, $w_{i,m}^{A'}$, and $fd_{i,m}$ for each point $p_{i,m}$, on the target track pattern recognition accuracy. Table 6 illustrates that the accuracy of target track pattern recognition increases first and then decreases when $c(w_{i,m}^{A}, w_{i,m}^{A'}, fd_{i,m})$ increases from 1 to 5. When $c(w_{i,m}^{A}, w_{i,m}^{A'}, fd_{i,m})$ is the maximum value of five, the weight $w_{i,m}^{A}$ of the auxiliary interpolation point $p_{i,m}^{A}$ is too large, which leads to a large deviation when filling the line-type track and arc-type track, resulting in low accuracy. When $c(w_{i,m}^{A}, w_{i,m}^{A'}, fd_{i,m})$ is the minimum value of one, the weight $w_{i,m}^{A'}$ of the auxiliary interpolation point $p_{i,m}^{A'}$ is too large, which causes the curved track segment to be filled as a straight track segment such that the recognition accuracy is also low. When $c(w_{i,m}^{A}, w_{i,m}^{A'}, fd_{i,m})$ is three, algorithm MPFL balances the filling effects of different types of tracks, thereby obtaining the highest recognition accuracy. Therefore, we set the value of $c(w_{i,m}^{A}, w_{i,m}^{A'}, fd_{i,m})$ as 3 in the following experiments.

5.2.3 *Recognition accuracy w.r.t. the number of labeled tracks*

Table 7 reports the recognition accuracy with respect to the different number of labeled tracks used for training our model. As the total number of labeled data increases from 100 to 1000, the recognition accuracy increases. Note that the training data set includes examples for four types and each type has an equal number of examples. When the amount of labeled data is 400 or more, i.e., the amount of labeled data for each type of track is 100, the increment of accuracy reaches the plateau. In our experiments, when about 2.5% of the total radar sequences are used as examples in training, our proposed method achieved satisfactory performance. To balance the training complexity and performance, we set the amount of labeled data as 400 in the remaining experiments.

5.2.4 *Impact of different features and feature pairs*

We study the impact of different features and feature combinations on the performance of the radar target track pattern recognition. Figure 7 shows that the SRFFC (a combination of track angle and TF) achieves the best accuracy, which is above 98% and better than track angle or TF alone by 4% and 5.5%, respectively. The extracted SRFFC can increase the classification accuracy, and the introduced AFF can effectively improve the robustness of the algorithm. Figure 8 shows that the precision with the SRFFC is much higher than that with other features, indicating that the extracted SRFFC is effective and can significantly improve the precision of the track classification. Figure 9 shows that the recall of using the SRFFC is above 98%, which is much higher than that of using other features or feature combinations. In summary, the extracted SRFFC obtains the best performance in the three performance metrics.

Table 7 Track type identification accuracy with different amounts of labeled data.

Amount of labeled data	100	200	400	600	800	1000
Accuracy	91.8%	92.1%	98.0%	98.0%	97.9%	98.2%

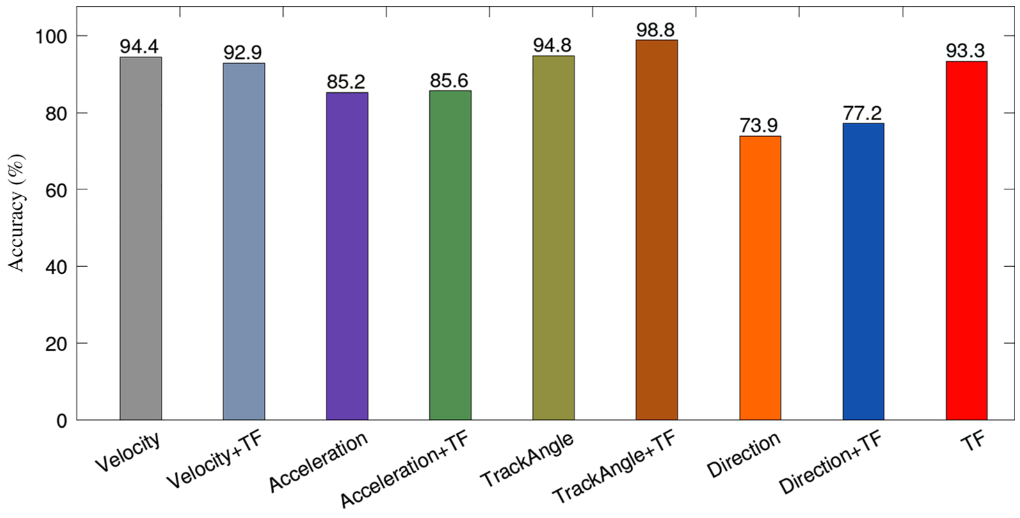

Fig. 7 Accuracy of track identification with different features.

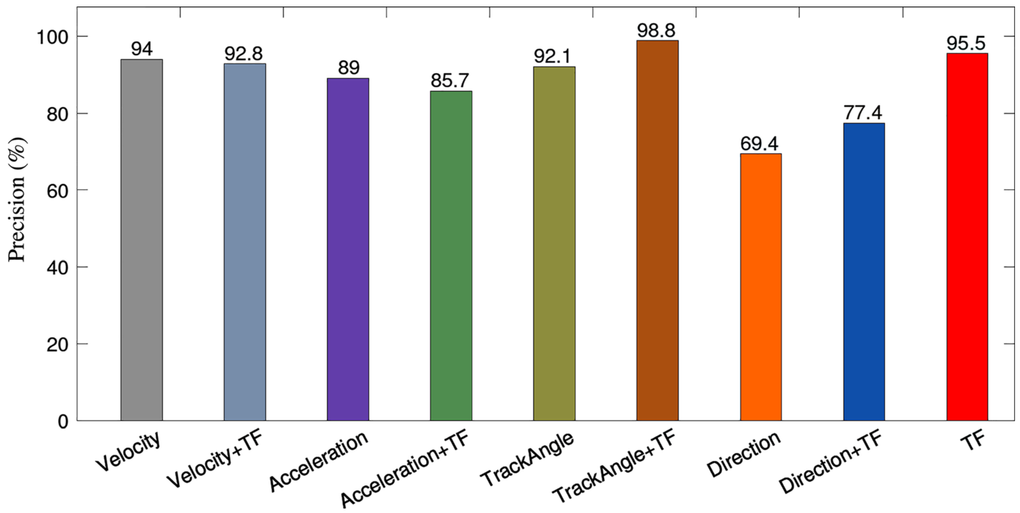

Fig. 8 Precision of track identification with different features.

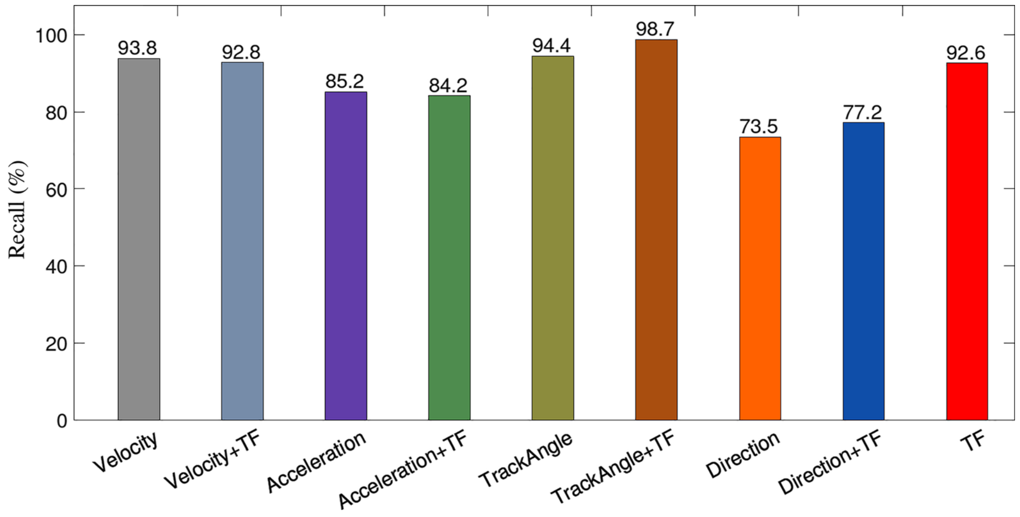

Fig. 9 Recall of track identification with different features.

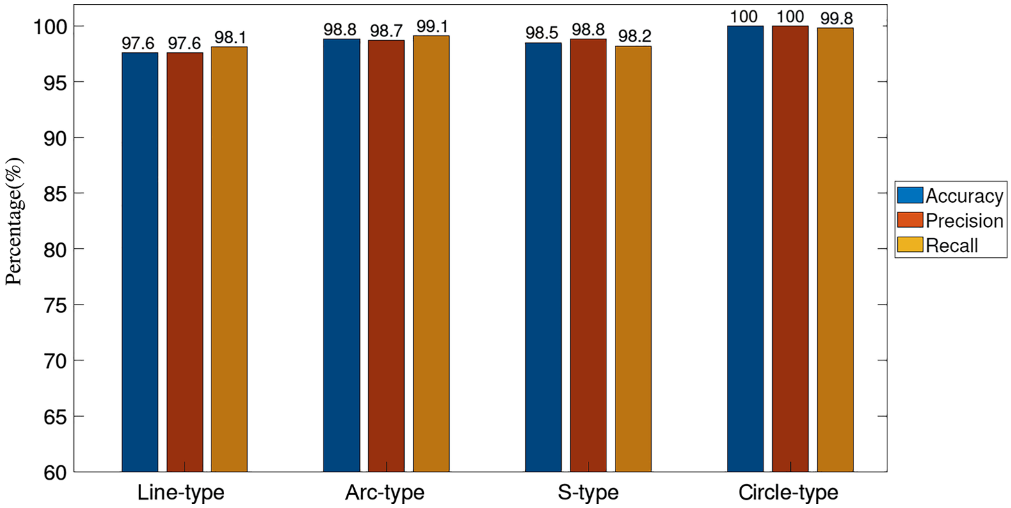

Fig. 10 Recognition performance on different target track types.

5.2.5 *Performance of different track patterns*

We study the recognition performance of algorithm SSTT on different types of target tracks. Figure 10 shows the recognition accuracy of each type of track pattern. The recognition accuracy is >97% in all cases, which demonstrates that our proposed method accurately identifies the track patterns. In particular, the recognition accuracy of circle-type tracks reaches 100%. In addition, the precision of algorithm SSTT for all four types of target tracks is higher than 97%, which shows that the precision of algorithm SSTT for all types of target tracks is high. For line-type and arc-type tracks, the recall of algorithm SSTT is stable at about 98%, and the recall for S-type and circle-type tracks is higher than 99%, illustrating that algorithm SSTT can achieve good performance in terms of recall. In summary, algorithm SSTT obtains good performance on each type of target track.

5.2.6 *Comparison of track pattern recognition algorithms*

We compare algorithm SSTT with other target track recognition algorithms, including CNN,[30] LSTM,[31] and GAN.[32] The experimental results in Fig. 11 show that algorithm SSTT achieves better performance than CNN, LSTM, and GAN in terms of the three performance metrics.

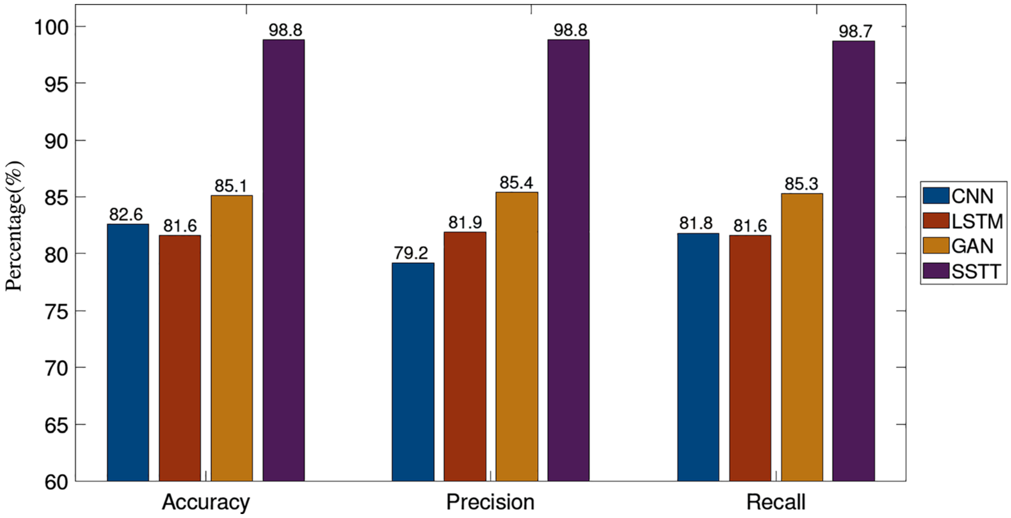

Fig. 11 Recognition performance of different algorithms.

Compared with the second-best, our proposed method achieves a minimum improvement of 15.7%. CNN extracts the features from the track data through the convolution operation for recognition, while ignoring the temporal relationship between the track points and the impact of insufficient labeled data. LSTM captures the features of track data in the time domain but fails to fully utilize the spatial features of track data. GAN solves the problem of insufficient labeled data while neglecting the negative impact of outliers in the track data. When there is only a small amount of labeled data, algorithm SSTT improves the performance with outlier processing and the SSGAN.

6 Conclusions

In this paper, we proposed a radar target track pattern recognition algorithm based on a SSGAN for few labeled radar target track data with outliers. Our proposed method contains two modules of outlier processing and target track pattern recognition. In the outlier processing module, we adopted ORLC to identify and eliminate the outliers existing in the original radar target track data and used MPFL to realize the filling of missing points. In the target track pattern recognition module, we extracted four types of BFFs, selected the SRFF from the BFFs, extracted the AFF from the BFFs, and combined the SRFF and the AFF as the SRFFC. We then input the SRFFC into the constructed SSGAN model to achieve accurate target track pattern recognition. The simulation results demonstrate that the proposed algorithm SSTT obtains good performance on each type of target track and achieves good and stable performance with the amount of labeled data for each type of track being only 100 and that the extracted SRFFC effectively improves the target track pattern recognition performance with outliers in terms of accuracy, precision, and recall.

Acknowledgments

This work was partly supported by the National Key Research and Development Plan under grant # 2018YFB2000505 and National Natural Science Foundation of China under grant # 61806067.

References

1. D. Feng and X. Yuan, "Automatic construction of aerial corridor for navigation of unmanned aircraft systems in class G airspace using LiDAR," *Proc. SPIE* **9828**, 98280I (2016).
2. IATA Annual Review 2017, https://www.iata.org/contentassets/c81222d96c9a4e0bb4ff6ced0126f0bb/iata-annual-review-2017.pdf (accessed November 2020).
3. L. Li, "Anomaly detection in airline routine operations using flight data recorder data," Massachusetts Institute of Technology (2013).
4. M. Etemad, A. Soares, and A. Hoseyni, "A trajectory segmentation algorithm based on interpolation-based change detection strategies," in *EDBT/ICDT Workshops* (2019).
5. W. S. Chen, J. Liu, and J. Li, "Classification of UAV and bird target in low-altitude airspace with surveillance radar data," *Aeronaut. J.* **123**(1260), 191–211 (2019).
6. F. Meng et al., "An overview on trajectory outlier detection," *Artif. Intell. Rev.* **52**(4), 2437–2456 (2019).
7. I. Goodfellow, J. Pouget-Abadie, and M. Mirza, "Generative adversarial networks," in *Adv. Neural Inf. Process. Syst.*, pp. 2672–2680 (2014).
8. K. Lee, "Radar target recognition by frequency-diversity RCS together with kernel scatter difference discrimination," *Prog. Electromagn. Res.* **87**, 137–145 (2019).
9. B. Ding et al., "Target recognition in synthetic aperture radar images via matching of attributed scattering centers," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **10**(7), 3334–3347 (2017).
10. S. J. Lee, et al., "Target identification using bistatic high-resolution range profiles," *IET Radar Sonar Navig.* **11**(3), 498–504 (2017).

11. N. Chantasen, A. Boonpoonga, and S. Burintramart, "Radar target identification of coated object using Cauchy method," in *IEEE Conf. Antenna Meas. and Appl.*, IEEE, pp. 1–3 (2015).
12. Y. Guo, H. Xiao, and Q. Fu, "Least square support vector data description for HRRP-based radar target recognition," *Appl. Intell.* **46**(2), 365–372 (2017).
13. A. Karine et al., "Target recognition in radar images using weighted statistical dictionary-based sparse representation," *IEEE Geosci. Remote Sens. Lett.* **14**(12), 2403–2407 (2017).
14. J. Chen, L. Du, and H. He, "Convolutional factor analysis model with application to radar automatic output recognition," *Pattern Recognit.* **87**, 140–156 (2019).
15. M. Pan, J. Jiang, and Q. Kong, "Radar HRRP target recognition based on t-SNE segmentation and discriminant deep belief network," *IEEE Geosci. Remote Sens. Lett.* **14**(9), 1609–1613 (2017).
16. B. Xu, B. Chen, and J. Wan, "Target-aware recurrent attentional network for radar HRRP target recognition," *Signal Process.* **155**, 268–280 (2019).
17. J. Choi et al., "Classification of inter-urban highway drivers resting behavior for advanced driver-assistance system technologies using vehicle trajectory data from car navigation systems," *Sustainability* **12**(15), 5936 (2020).
18. J. Wu et al., "A novel method of vehicle-pedestrian near-crash identification with roadside LiDAR data," *Accid. Anal. Prev.* **121**, 238–249 (2018).
19. X. Pan et al., "Online classification of frequent behaviours based on multidimensional trajectories," *IET Radar Sonar Navig.* **11**(7), 1147–1154 (2017).
20. Y. Fan et al., "Semi-supervised target track recognition based on few labeled radar data with outliers," in *Int. Conf. Urban Intell. and Appl.*, Springer (2020).
21. R. Deshmukh and I. Hwang, "Incremental-learning-based unsupervised anomaly detection algorithm for terminal airspace operations," *J. Aerosp. Inf. Syst.* **16**(9), 362–384 (2019).
22. E. Habler and A. Shabtai, "Using LSTM encoder-decoder algorithm for detecting anomalous ADS-B messages," *Comput. Secur.* **78**, 155–173 (2018).
23. T. G. Puranik and D. N. Mavris, "Anomaly detection in general-aviation operations using energy metrics and flight-data records," *J. Aerosp. Inf. Syst.* **15**(1), 22–36 (2018).
24. R. Laganierer, *OpenCV2 Computer Vision Application Programming Cookbook*, pp. 143–164, Packt Publishing, Birmingham (2011).
25. M. Kesaniemi and K. Virtanen, "Direct least square fitting of hyperellipsoids," *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(1), 63–76 (2018).
26. T. Cover and J. Thomas, *Elements of Information Theory*, John Wiley & Sons, New Jersey (1991).
27. B. R. Mahafza and A. Elsherbeni, *MATLAB Simulations for Radar Systems Design*, CRC Press, Florida (2003).
28. A. Alahi et al., "Social LSTM: human trajectory prediction in crowded spaces," *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 961–971 (2016).
29. A. Gupta et al., "Social GAN: socially acceptable trajectories with generative adversarial networks," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 2255–2264 (2018).
30. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM* **60**(6), 84–90 (2017).
31. C. Olah, "Understanding LSTM networks," 2015, https://web.stanford.edu/class/cs379c/archive/2018/class_messages_listing/content/Artificial_Neural_Network_Technology_Tutorials/OlahLSTM-NEURAL-NETWORK-TUTORIAL-15.pdf (accessed 11 November 2020).
32. T. Salimans et al., "Improved techniques for training GANs," in *Adv. Neural Inf. Process. Syst.*, 2234–2242 (2016).

Yuqi Fan is an associate professor in the School of Computer Science and Information Engineering at Hefei University of Technology, China. He received his BS and MS degrees in computer science and engineering from Hefei University of Technology in 1999 and 2003, respectively, and received his PhD in computer science and engineering from Wright State University in 2009. His research interests include blockchain, computer networks, cloud computing, and cyber-physical systems.

Guangming Shen received his BS degree in computer science from Zhejiang Normal University, China, in 2018. Currently, he is a graduate student in computer science at School of Computer and Information Engineering at Hefei University of Technology, Hefei, China. His research interests include deep learning and blockchain.

Xiong Xu is a research assistant at the State Key Laboratory of Complex Electromagnetic Environment Effects on Electronic and Information Systems (CEMEE). He received his PhD in computer science and engineering from University of Electronic Science and Technology of China in 2012. His research interests include signal processing and machine learning.

Juan Xu is currently an associate professor in the School of Computer Science and Information Engineering, Hefei University of Technology. She received her PhD in computer science from Hefei University of Technology, Hefei, Anhui, China in 2012. Her research interests include industrial IoT and intelligent fault diagnosis.

Xiaohui Yuan received his BS degree in electrical engineering from the Hefei University of Technology, China, in 1996 and his PhD in computer science from Tulane University in 2004. Currently, he is an associate professor at the University of North Texas. His research interests include computer vision, artificial intelligence, data mining, and machine learning. His research findings have been published in more than 170 peer-reviewed papers. He is the editor-in-chief of the *International Journal of Smart Sensor Technologies and Applications*, serves on the editorial board of several international journals, and as chairs several international conferences. He is a recipient of the Ralph E. Powe Junior Faculty Enhancement Award in 2008.