

Research Article

Learning to Track Multiple Radar Targets with Long Short-Term Memory Networks

Fei Song ^{1,2} Yong Li ¹ Wei Cheng ¹ and Limeng Dong ¹

¹School of Electronic Information, Northwestern Polytechnical University, Xi'an 710072, China

²School of Electronic Engineering, Xi'an Aeronautical Institute, Xi'an 710077, China

Correspondence should be addressed to Yong Li; ruikel@nwpu.edu.cn

Received 12 August 2022; Revised 30 September 2022; Accepted 14 October 2022; Published 15 February 2023

Academic Editor: Mingqian Liu

Copyright © 2023 Fei Song et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Radar multitarget tracking in a dense clutter environment remains a complex problem to be solved. Most existing solutions still rely on complex motion models and prior distribution knowledge. In this paper, a new online tracking method based on a long short-term memory (LSTM) network is proposed. It combines state prediction, measurement association, and trajectory management functions in an end-to-end manner. We employ LSTM networks to model target motion and trajectory associations, relying on their strong learning ability to learn target motion properties and long-term dependence of trajectory associations from noisy data. Moreover, to address the problem of missing appearance information of radar targets, we propose an architecture based on the LSTM network to calculate similarity function by extracting long-term motion features. And the similarity is applied to trajectory associations to improve their robustness. Our proposed method is validated in simulation scenarios and achieves good results.

1. Introduction

In an environment with much clutter, multitarget tracking based on radar detection results is a challenging task. An important branch of its solution is the multitarget tracking algorithm based on data association, which can be generally decomposed into the following three aspects: modeling the motion of the tracked target to estimate trajectory parameters and filter measurements, associating the measurements with the motion trajectory to distinguish the measurements from different targets or background noise and clutter interference, and effectively managing different motion trajectories to determine the birth, retention identity, and termination of different target trajectories.

For modeling the motion of the tracked target, the traditional method is based on the Bayesian filtering theory [1, 2]. Earlier, the Kalman filter is used to obtain unbiased optimal estimation of Gaussian linear targets. For nonlinear motion targets, the following developed methods consist of extended Kalman filter (EKF) [3], unsensitive Kalman filter (UKF) [4],

interactive multiple model (IMM) [5] algorithm, and so on. All of these methods require limited motion models in advance. Particle filtering (PF) [6], which does not need the aforementioned assumptions, however, has low computational efficiency and sample poverty problem.

For tracking trajectory association and trajectory management, it can theoretically be regarded as a maximum matching problem of a bipartite graph. For this problem, the Hungarian algorithm (HA) [7] and Kuhn and Munkres algorithm (KM) are first used. Later, many more complex algorithms have appeared, including multiple hypothesis tracker (MHT) [8], which establishes a potential tracking hypothesis tree for each candidate target and calculates the tracking probability to select the most likely tracking combination. Another popular technique is joint probabilistic data association (JPDA) [9]. By building a validation matrix, it can calculate all feasible joint association event probabilities and associate targets based on the obtained scores. However, these traditional methods not only require knowing the number of targets, starting positions, and clutter distribution

but also result in “combination explosions” increase of calculation when the number of targets or clutter is enlarged. This will not only reduce the tracking accuracy but also may lead to trace association errors.

Recently, deep learning has made great progress in classification and detection applications in many fields. For example, it is widely used in identifying interference in wireless communication systems [10] and lightweight radio frequency fingerprint identification (RFFID) systems [11]. Especially in computer vision, the target tracking methods in video based on deep learning have also improved, such as pedestrian monitoring [12, 13], car driving monitoring [14, 15], and biological sequence tracking [16]. However, deep learning-based multitarget tracking algorithms that are applied to radar targets are relatively rare. This may be caused by several reasons. First, deep learning requires a large amount of annotated training data, but real data is difficult to obtain in radar multitarget scenarios. Secondly, the measurement of radar targets nearly has no apparent information, which prevents the appearance detection information commonly used in video target tracking. Thirdly, radar target tracking and detection have real-time requirements, and the tracking algorithm generally needs to be used online.

The main contributions of this paper are as follows:

- (1) To overcome these problems in traditional radar multitarget tracking based on data association, inspired by deep learning ideas, we propose an end-to-end multitarget tracking structure based on recurrent neural networks. We use LSTM networks to model the state prediction and measurement association parts of the multitarget tracking algorithm, respectively. Relying on the powerful learning ability of LSTM networks, it is able to obtain the long-term dependence of multiple tracking target motion states and measurement association by learning from a large amount of training data. Then, it combines status update and tracking management into a unified network structure and finally realizes multitarget tracking. The advantages of this method are that the target motion models and clutter distribution are not required to be known in advance, and the combination explosion problem in the traditional data association method is alleviated. Meanwhile, the tracking process runs online without caching any future frame information
- (2) To solve the problem of missing appearance information when the neural network is applied to the radar multitarget tracking algorithm, we creatively design a motion feature extraction LSTM to extract motion features from the speed of the targets and calculate the similarity scores, which are used to learn long-dependent measurement associations
- (3) To address the problem of insufficient training data, we propose a method to obtain extensive training data from simulation models. Our multitarget track-

ing architecture is also confirmed by a simulation analysis of the simulated data

2. Related Work

Lately, target tracking in the video application field has been extended from a single target to multitarget tracking [17]. And the specific methods have also been extended from constructing complex appearance models to focusing on the motion model and interaction of the targets simultaneously [18]. In these applications, deep learning methods play an important role, mainly reflected in the following three aspects:

- (1) Using the deep neural network to extract high-order features of the tracking targets, especially for the appearance model, this can effectively improve the target tracking performance. Earlier, Reid [8] use AlexNet for extracting deep features from targets for the use of the MHT framework. Leal-Taixé et al. [19] use the Siamese network structure to extract deep features. By combining the depth features and motion information with a gradient enhancement algorithm, the tracking problem is expressed as linear programming and solved effectively. However, there is no apparent information in the radar signal, so these methods are not suitable for extracting the depth features of the radar signal
- (2) Deep neural networks are used to directly learn key components of a multitarget tracking framework. Chen et al. [20] construct two CNN-based classifiers using the features of fasterRCNN [21] from the VGG-16 model [22] as input and combining the confidence degree of the classifier as particle weights to obtain tracking results by particle filtering. Xiang et al. [23] propose a CNN based on triplet state loss to learn the distance measure between tracker and detection and thus constitute the cost of the bipartite graph, which can be effectively solved by the Hungarian algorithm. However, radar targets are different from video targets. It often has a poor performance to extract radar target features and classify them using the CNN network
- (3) Deep neural networks are designed directly in an end-to-end manner to obtain multitarget tracking results. The multitarget tracking task involves many interwoven components and is difficult to be modeled as a whole for learning. There have been some preliminary researches recently. Milan et al. [24] unify target state prediction, state update, and existence probability calculation under a whole RNN. Also, they design a set of LSTM networks for data matching matrix calculation and act on the state update process. These architectures achieved good results. Sadeghian et al. [25] design a more complex RNN structure, which is divided into three subnetworks to extract multiframe appearance features, motion features, and interaction features, respectively, and integrate them into a top-

level RNN for time-dependent comprehensive inferring, so as to obtain the final matching probability. Experimental results show that this method is more robust

Our architecture is similar to that of [24] but different in three key aspects: First, RNN is used in [24] to deal with motion state prediction. If the time interval is too long, the time dependence will be weakened. Instead, we use the LSTM network for motion state prediction, which enhances the long-term dependence learning ability of target states. Secondly, in the measurement association part, we use the motion feature extraction network based on LSTM to calculate the similarities for data associations, which has stronger anti-interference ability than the method in [24] only using the Euclidean distance between the target and each measurement to carry out data associations. Thirdly, we use different loss functions due to different network architectures.

3. Our Approach

For addressing the aforementioned radar target tracking problem, we propose an online end-to-end radar multitarget tracking network architecture based on LSTM (see Figure 1). Our architecture implements target state prediction, measurement association, state update, and tracking path management under a unified LSTM network architecture, and all model learning is completed in an online end-to-end manner.

3.1. Notations. In our application scenario, the target state vector we care about is $x_t \in R^{N \cdot D}$, including x coordinate, y coordinate, x axis velocity v_x , and y axis velocity v_y , as represented (x, y, v_x, v_y) , $D = 4$, N represents the maximum number of targets tracked simultaneously in a frame. The measurement value vector is $z_t \in R^{M \cdot D}$, whose dimension D is the same as the target state vector, and M represents the maximum number of detected values in a frame, including the targets and clutter. The association probability matrix of the measurement association part is expressed as $A_t \in [0, 1]^{N \cdot (M+1)}$, each row of which represents the probability that each measurement value belongs to a certain target. That is, A_t^{ij} presents the probability of the j th measurement value belonging to the i th target. The added column indicates the probability that the target lacks measurement, so that each row is satisfied $\forall_i : \sum_j A_t^{ij} = 1$. The path management section uses a probability vector $\eta_t \in [0, 1]^N$ to express the probability of each target's existence, which can accurately describe the target generation and termination processes.

3.2. Motion-Association Multitarget Tracking with LSTMs (MA-LSTM). In our architectural design (see Figure 1), we partition the state prediction, state update, and path management into a target motion module and measurement association into another target association module. This primarily deals with the diversity of measurement association algorithms and facilitates the replacement of this part for

tracking effectiveness comparison. At the same time, the two modules are trained separately for easy convergence.

3.2.1. State Prediction in Motion Module. State prediction is the first submodule of the motion module. The task of this module is to learn a complex motion pattern, which can be noisy and nonlinear, and predict the future motion states based on the trend of the past motion states. This is actually a time series prediction problem, which we designed to achieve by using a LSTM network structure, as shown in Figure 2. We train the network on noisy motion trajectories so that the network can completely learn the long-term dependent motion pattern from the training data without prior knowledge.

More specifically, at the current moment t , both the hidden state h_t and the cell state c_t of the prediction LSTM come from the learning of the previous target motion state, and the input x_t is the target motion state at the current moment. Through the internal processing of the LSTM network, as described below, the predicted motion state \hat{x}_{t+1} at the next moment as well as the hidden state h_{t+1} and cell state c_{t+1} can be obtained to continue the transmission.

The formula is expressed as follows:

$$\begin{aligned} h' &= W_{hh'} h_t + W_{xh'} x_t, \\ c_{t+1} &= \sigma(h') c_t + \sigma(h') \tanh(h'), \\ h_{t+1} &= \sigma(h') \tanh(c_{t+1}), \\ \hat{x}_{t+1} &= W_{ho} h_{t+1}, \end{aligned} \quad (1)$$

where $\sigma()$ is the sigmoid and $\tanh()$ is the hyperbolic tangent activation functions, and the learnable parameters are expressed as $W_{hh'} W_{xh'} W_{ho}$.

3.2.2. State Update and Track Management in Motion Module. Status update and track management is the second submodule of the motion module. The task of this module is to update the motion state of the target and identify and judge the start and end of the target trajectory on the basis of the obtained target prediction, measured values, and association matrix. This is an important step in the multitarget tracking architecture, for taking into account the clutter interference in the multitarget measurements and the association of the measured data. The network structure we designed is shown in Figure 3.

In detail, the measured values z_{t+1} at $t+1$ moment are combined with the predicted values \hat{x}_{t+1} output by the state prediction submodule, as $[\hat{x}_{t+1}; z_{t+1}]$. Then, using the dot product with the association matrix A_{t+1} output by the target association module, the possible state of the target is $[\hat{x}_{t+1}; z_{t+1}] \cdot A_{t+1}$. Then, it is multiplied by the target existence probability η_t . With the multiplied result and the hidden state of the prediction module h_{t+1} , the updated motion state x_{t+1}^* is obtained after a nonlinear transformation, and the target existence probability η_{t+1}^* is also calculated.

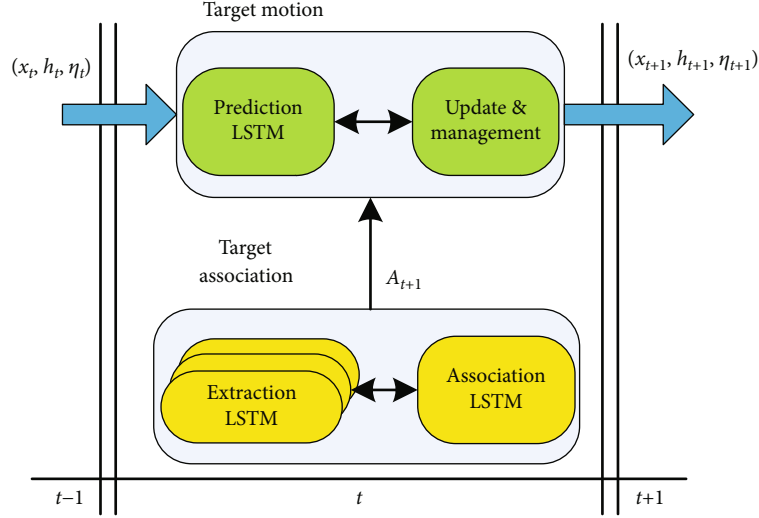


FIGURE 1: Overview of our architecture. In a time step t , the architecture is composed of two parts as target motion module and target association module (see Section 3.2 for details). The input can be expressed as the following: at the current moment t , the target motion state x_t , the hidden state h_t , and the probability vector η_t (see below) come from the learning of the previous moment $t-1$. After the processing of our architecture, the output is the estimated target motion state x_{t+1} , the hidden state h_{t+1} , and the probability vector η_{t+1} of the next moment $t+1$.

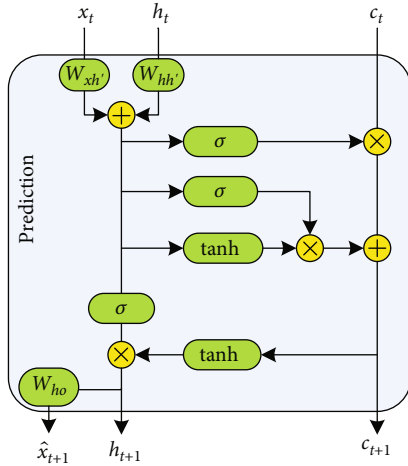


FIGURE 2: Prediction submodule of motion module on the basis of LSTM is used to estimate target state without the help of measurement.

The formula is expressed as follows:

$$\begin{aligned}
 \tilde{x}_{t+1} &= [\tilde{x}_{t+1}; z_{t+1}] \cdot A_{t+1}, \\
 \tilde{h}_{t+1} &= h_{t+1} + W_{x\tilde{h}}(\tilde{x}_{t+1}\eta_t), \\
 x_{t+1}^* &= W_{x^*o} \tanh(\tilde{h}_{t+1}), \\
 \eta_{t+1}^* &= \sigma(W_{\eta^*o} \tanh(\tilde{h}_{t+1})),
 \end{aligned} \tag{2}$$

where $\sigma(\cdot)$ is the sigmoid and $\tanh(\cdot)$ is the hyperbolic tangent activation functions, and the learnable parameters are expressed as $W_{x\tilde{h}}, W_{x^*o}, W_{\eta^*o}$.

3.2.3. *Loss of Motion Module.* In our architecture, we are interested in tracking performance-related losses and propose the following loss functions that meet our needs:

$$\begin{aligned}
 L(\hat{x}_{t+1}, x_{t+1}^*, \eta_{t+1}^*, x_{t+1}, \eta_{t+1}) &= \alpha \sum \|\hat{x}_{t+1} - x_{t+1}\|^2 \\
 &+ \beta \sum \|x_{t+1}^* - x_{t+1}\|^2 + \mu L_\eta + \psi \bar{\eta}_{t+1},
 \end{aligned} \tag{3}$$

$$L_\eta(\eta_{t+1}^*, \eta_{t+1}) = \eta_{t+1} \log \eta_{t+1}^* + (1 - \eta_{t+1}) \log (1 - \eta_{t+1}^*), \tag{4}$$

$$\bar{\eta}_{t+1} = \text{abs}(\eta_{t+1}^* - \eta_t), \tag{5}$$

where \hat{x}_{t+1} is the predicted value of the prediction module, x_{t+1}^* is the predicted value of the update module, and η_{t+1}^* is the predicted value of the existence probability. x_{t+1}, η_{t+1} are true values.

This loss function in formula (3) we designed consists of four parts. The first part is to predict the motion state \hat{x}_{t+1} of the targets without measured values, and we take the mean square error (MSE) of the predicted values and the real values. The second part is the state prediction x_{t+1}^* after the measurement update. We take the mean square error of the updated values and the real values. The third part is to predict the probability of the target's existence η_{t+1}^* at each moment, and we use binary cross entropy loss function, as shown in formula (4). The fourth part is the additional smoothing variable $\bar{\eta}_{t+1}$, in order to smooth the absolute difference between two consecutive η values, so as to prevent the loss of the target in the tracking process.

3.2.4. *Target Association Module.* The target association module is the most challenging and creative part of the multitarget tracking architecture. We design it as an independent association module and further subdivide it into two parts: motion extraction LSTM and association LSTM.

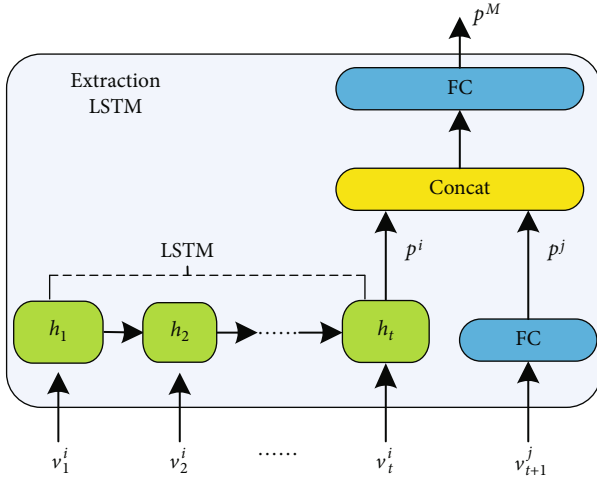


FIGURE 5: The extraction LSTM submodule of association module is used to extract the velocity features of the target's long-term motion and calculate the similarity between the new measurement and the existing trajectory.

our framework. The learning rate is initially set at 0.001 decreasing by 10% for every 10 periods. We set the maximum number of iterations to 10,000, which is enough to achieve convergence. Training for these network architectures takes about 10 hours on a GPU.

3.3.1. Training Data. As mentioned above, deep learning network learning requires a large amount of training data. However, there are very few open marker datasets for multi-target tracking of radar signals. Therefore, we generate training data from radar motion simulation models. The training trajectories we generated are set with multiple targets, whose birth time and lifetime are random, and incorporate a large number of random uniformly distributed clutter. The initial position and velocity of each target are randomly distributed within a certain range, and the conventional constant-velocity motion model and constant acceleration motion model of the radar target are adopted. The radar measurements are sampled every 2 seconds, and a Gaussian measurement noise is added, while the annotations for multitarget associations are manually added.

4. Experiments

We have proposed a radar multitarget tracking network architecture. To demonstrate its functionality, we first present experiments on simulated data and then give more insights and analysis of our performance.

In the simulation experiment scene, we choose 2D radar to track multiple moving targets, assuming that the target movements are independent of each other, and trajectory crossing may occur. The equation of motion for each target is

$$X_k = F_{k-1}X_{k-1} + G_{k-1}w_{k-1}, \quad (7)$$

where $X_k = (x, y, v_x, v_y, a_x, a_y)$ is the target state. W_{k-1} is

process noise representing random acceleration in the x and y axis direction, $w_k \sim N(0, \sigma_w^2 I_2)$, $\sigma_w = 10 \text{ m/s}^2$.

We assume that the target follows the constant acceleration (CA) model and the constant velocity (CV) model [27]. Their process equation F and process noise G are

$$F_{CA} = \begin{bmatrix} 1 & 0 & T & 0 & \frac{T^2}{2} & 0 \\ 0 & 1 & 0 & T & 0 & \frac{T^2}{2} \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$G_{CA} = \begin{bmatrix} \frac{T^2}{2} & 0 & T & 0 & 1 & 0 \\ 0 & \frac{T^2}{2} & 0 & T & 0 & 1 \end{bmatrix}, \quad (8)$$

$$F_{CT} = \begin{bmatrix} 1 & 0 & T & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$G_{CT} = \begin{bmatrix} \frac{T^2}{2} & 0 & T & 0 & 1 & 0 \\ 0 & \frac{T^2}{2} & 0 & T & 0 & 1 \end{bmatrix},$$

where T is the sampling period.

The target observation equation is

$$Z_k = H_k X_k + v_k, \quad (9)$$

where H_k is the observation matrix; the measurement noise v_k is the Gaussian white noise with zero mean and covariance matrix R_k .

The observation vector of radar is $Z_k = (\theta_k, r_k)$, where θ_k is the azimuth of the sensor observation target and r_k is the distance of the sensor observation target. The target observation model is

$$\begin{bmatrix} \theta_k \\ r_k \end{bmatrix} = \begin{bmatrix} \arctan \frac{y_k - y}{x_k - x} \\ \sqrt{(x_k - x)^2 + (y_k - y)^2} \end{bmatrix} + \begin{bmatrix} v_\theta \\ v_r \end{bmatrix}, \quad (10)$$

where $v_\theta \sim N(0, \sigma_\theta^2)$, $\sigma_\theta = 1 \times (\pi/180) \text{ rad/s}$ and $v_r \sim N(0, \sigma_r^2)$, $\sigma_r = 10 \text{ m}$.

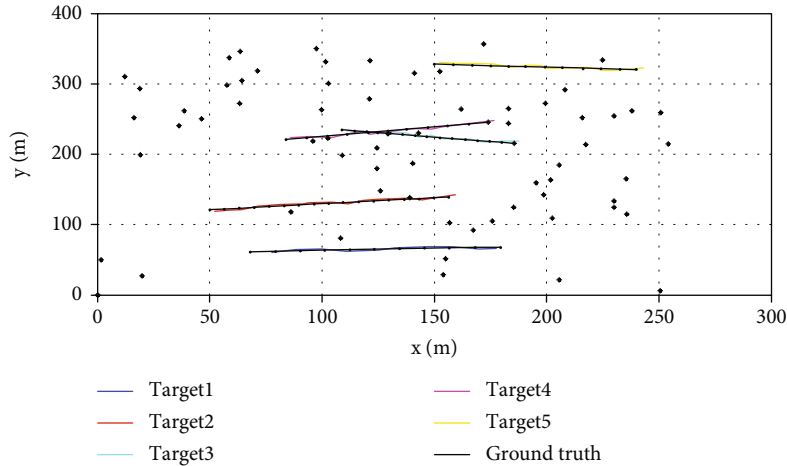


FIGURE 6: The tracking results of our method on 5 targets. The different color curves represent the different predicted target trajectories. The black curve is the ground truth.

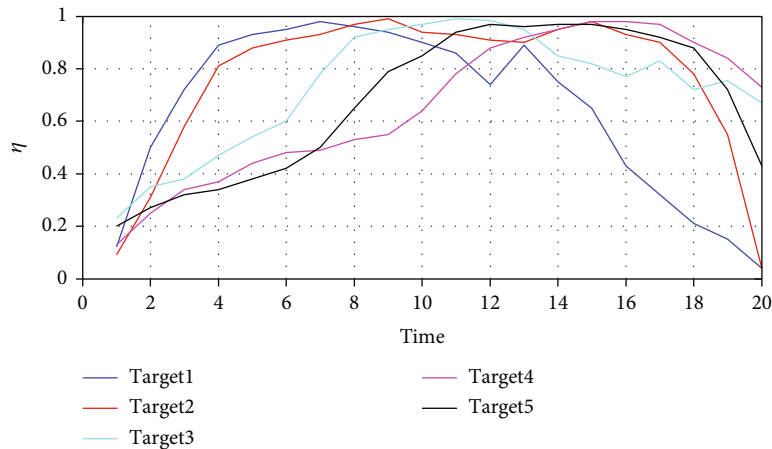


FIGURE 7: The predicted value of the existence probability η of the 5 targets in the time frame. The threshold is set as 0.6.

TABLE 1: OSPA distance of the four methods.

Method	OSPA ↓	OSPA Loc ↓	OSPA card ↓
KF-HA	83.79	77.48	8.09
JPDA	79.86	75.39	8.22
M-HA	79.12	73.15	12.44
MA-LSTM	70.08	65.78	7.86

On this basis, we set an observation area of $1 \text{ km} \times 1 \text{ km}$, 5 radar tracking targets with random birth and death time. Each target has detected probability $P_D = 0.99$. In the observed region, the clutter follows a uniform Poisson distribution, and the clutter intensity is set as $\lambda_k = 30$.

To generate large amounts of training data, the initial position of the target is randomly distributed within the observation area, the initial linear velocity is randomly distributed from 100 m/s to 300 m/s , and the acceleration is randomly distributed from 0 m/s^2 to 8 m/s^2 . The times of birth and death for each target are randomly distributed between 3 and 18 s. At the same time, the number of targets

at each sampling time is manually annotated. From this, we produce 10,000 random paths and their observation data, respectively. These are sufficient to train our network architecture.

The results of the tracking are shown in Figures 6 and 7. From this, we can qualitatively conclude that our method can track multiple moving radar targets in a large number of clutter environments, although the trajectories of these targets may be crossed. Furthermore, we also see that the predicted trajectories always start and end one or two points later than the real trajectories, which is difficult to avoid with the online method.

For quantitative analysis, we compare our proposed method (MA-LSTM) with three baseline methods. The first baseline method is the traditional KF-HA, which employs a combination of Kalman filter and data association via the Hungarian algorithm. The second baseline approach is JPDA mentioned above. The third comparison is comprised of our motion module and the data association using the Hungarian algorithm (M-HA). The optimal submode allocation (OSPA) [28] distance for each method is calculated.

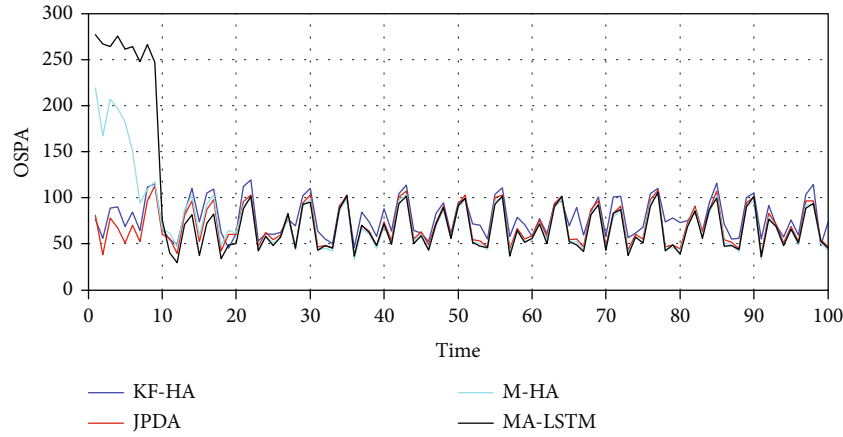


FIGURE 8: The OSPA values for different methods.

OSPA is a consistency measurement method for the overall performance evaluation of the target tracking system, which can be used to measure the error between real track and estimated track and can separate the total error into distance errors and correlation errors. For the tunable parameter distance sensitivity parameter p and associated sensitivity parameter c , we set $p = 20$ and $c = 350$, respectively.

The calculation results are shown in Table 1. MA-LSTM outperforms the other methods in all three aspects. It has great advantages in overall OSPA and associated OSPA and is slightly better than JPDA in terms of distance OSPA. It is worth noting that compared with the HA method for data association, the proposed method of using LSTM for data association and using motion velocity features to calculate similarity has a positive impact on the overall tracking effect, especially on the reduction of association error.

As shown in Figure 8, this intuitively shows the trend of different method OSPA values over time under a long period of test data. It can be seen that in the early stage of the test, the OSPA value of the learning method is high and then decreases rapidly after that, because of the learning of motion properties in the early stages of tracking. The OSPA value of our method is significantly reduced from the 10th frame, because the speed feature extraction sequence length in the data association module is set to 10. That is, the role of data association can be fully played after 10 frames; after that, the performance of our method is better than other methods.

5. Conclusion

This paper presents an LSTM-based network architecture for radar multitarget tracking. This architecture can effectively solve the problems of state prediction, measurement association, and trajectory management for radar multitarget tracking under much clutter. In addition, we propose to use the motion extraction LSTM to extract motion features to calculate similarity scores and use it to learn long-dependent target associations, which achieves good results. Our architecture is able to accomplish tracking tasks online and has been verified in simulation scenarios. In future

work, we plan to expand it into the video tracking field and make association strategies more robust by combining more clues to achieve better performance.

Data Availability

The experimental data used to support the findings of this study are available from the corresponding authors upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This research was supported by the Fundamental Research Funds for the Central Universities under Grant 3102019ZX015 and in part by the Fundamental Research Funds for the Central Universities under Grant D5000220131.

References

- [1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
- [2] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," *Proceedings of SPIE-The International Society for Optical Engineering*, vol. 3068, pp. 182–193, 1999.
- [3] H. Kushner, "Approximations to optimal nonlinear filters," *IEEE Transactions on Automatic Control*, vol. 12, no. 5, pp. 546–556, 1967.
- [4] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *IEEE Proceedings of 1995 American Control Conference-ACC'95*, pp. 1628–1632, Seattle, WA, USA, 1995.
- [5] X. R. Li, X. Zhi, and Y. Zhang, "Multiple-model estimation with variable structure. V. Likely-model set algorithm," *IEEE*

- Transactions on Aerospace and Electronic Systems*, vol. 36, no. 2, pp. 448–466, 2000.
- [6] A. Doucet, S. Godsill, and C. Andrieu, “On sequential Monte Carlo methods for Bayesian filtering,” *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.
- [7] H. W. Kuhn, “The Hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
- [8] D. B. Reid, “An algorithm for tracking multiple targets,” *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.
- [9] T. Fortmann, Y. B. Shalom, and M. Scheffe, “Sonar tracking of multiple targets using joint probabilistic data association,” *IEEE Oceanic Engineering*, vol. 8, no. 3, pp. 173–184, 1983.
- [10] M. Liu, Z. Liu, W. Lu, Y. Chen, X. Gao, and N. Zhao, “Distributed few-shot learning for intelligent recognition of communication jamming,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 3, pp. 395–405, 2022.
- [11] M. Liu, J. Wang, N. Zhao, Y. Chen, H. Song, and F. R. Yu, “Radio frequency fingerprint collaborative intelligent identification using incremental learning,” *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 5, pp. 3222–3233, 2021.
- [12] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, “Multiple object tracking using k-shortest paths optimization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1806–1819, 2011.
- [13] Y. Li, C. Huang, and R. Nevatia, “Learning to associate: hybrid boosted multi-target tracker for crowded scene,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2953–2960, Miami, FL, USA, 2009.
- [14] A. A. K. Runyoro, V. A. Ndume, Z. Mganilwa, and I. Kambira, “A smart approach for vehicle speed monitoring and accidents trend at black spot areas in Tanzania,” *Open Access Library Journal*, vol. 9, no. 3, pp. 1–14, 2022.
- [15] J. R. Perello-March, C. G. Burns, R. Woodman, M. T. Elliott, and S. A. Birrell, “Driver state monitoring: manipulating reliability expectations in simulated automated driving scenarios,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 99, pp. 1–11, 2021.
- [16] N. Chenouard, I. Bloch, and J. C. Olivo-Marin, “Multiple hypothesis tracking for cluttered biological image sequences,” *IEEE Transactions on Software Engineering*, vol. 35, no. 11, pp. 2736–2750, 2013.
- [17] Y. Xiang, A. Alahi, and S. Savarese, “Learning to track: online multi-object tracking by decision making,” in *IEEE International Conference on Computer Vision*, pp. 4705–4713, Santiago, Chile, 2015.
- [18] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, “Social lstm: human trajectory prediction in crowded spaces,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 961–971, Las Vegas, NV, USA, 2016.
- [19] L. Leal-Taixé, C. Canton-Ferrer, and K. Schindler, “Learning by tracking: Siamese CNN for robust target association,” in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 418–425, Las Vegas, NV, USA, 2016.
- [20] L. Chen, H. Ai, C. Shang, Z. Zhuang, and B. Bai, “Online multi-object tracking with convolutional neural networks,” in *IEEE International Conference on Image Processing (ICIP)*, pp. 645–649, Beijing, China, 2017.
- [21] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [22] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large scale image recognition,” 2015, <http://arxiv.org/abs/1409.1556>.
- [23] J. Xiang, G. Zhang, J. Hou, N. Sang, and R. Huang, “Multiple target tracking by learning feature representation and distance metric jointly,” 2018, <http://arxiv.org/abs/1802.03252>.
- [24] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, “Online multi-target tracking using recurrent neural networks,” in *31st AAAI Conference on Artificial Intelligence (AAAI 2017)*, pp. 4225–4232, San Francisco, USA, 2017.
- [25] A. Sadeghian, A. Alahi, and S. Savarese, “Tracking the untrackable: learning to track multiple cues with long-term dependencies,” in *IEEE International Conference on Computer Vision (ICCV)*, pp. 300–311, Venice, Italy, 2017.
- [26] A. Karpathy, J. Johnson, and F. F. Li, “Visualizing and understanding recurrent networks,” 2015, <http://arxiv.org/abs/1506.02078>.
- [27] C. Wang, P. Wu, S. He, and P. Yun, “Robust CPHD algorithm for maneuvering targets tracking via airborne pulsed Doppler radar,” *Optik-International Journal for Light and Electron Optics*, vol. 178, pp. 285–296, 2019.
- [28] B. Ristic, B. N. Vo, D. Clark, and B. T. Vo, “A metric for performance evaluation of multi-target tracking algorithms,” *IEEE Transactions on Signal Processing*, vol. 59, no. 7, pp. 3452–3457, 2011.