Detecting Anomalous Trajectories via Recurrent Neural Networks ⋆

Cong Ma[1], Zhenjiang Miao[1], Min Li[1], Shaoyue Song[1], and Ming-Hsuan Yang[2]

[1] Beijing Jiaotong University, China
[2] University of California, Merced, USA
{13112063, zjmiao, 16112066, 14112060}@bjtu.edu.cn, mhyang@ucmerced.edu

Abstract. Detecting anomalies from trajectory data is an important task in video surveillance. However, it is difficult to give a precise definition of this term since trajectory data obtained from different camera views may vary in shape, direction, and spatial distribution. In this paper, we propose trajectory distance metrics based on a recurrent neural network to measure similarities and detect anomalies from trajectory data. First, we use an autoencoder to capture the dynamic features of a trajectory. The distance between two trajectories is defined by the reconstruction errors based on the learned models. We then detect anomalies based on the nearest neighbors using the proposed metric. As such, we can deal with various kinds of anomalies in different scenes and detect anomalous trajectories in either a supervised or unsupervised manner. Experiments show that the proposed algorithm performs favorably against the state-of-the-art anomaly detections on the benchmark datasets.

1 Introduction

Anomaly detection is important in numerous vision tasks including traffic monitoring, motion analysis, and public safety. Trajectories of moving targets are good representations of object behaviors in video data and useful in detecting unusual events. On one hand, compared with appearance and low-level motion features, trajectories can provide object-level long-term information of target behaviors and motion patterns. On the other hand, compared with raw image data, trajectories are more compact and require less computational resources for motion analysis. In addition, thanks to the recent rapid development of object detection and tracking algorithms, fairly accurate trajectories can be extracted and analyzed for event analysis. Anomalies are usually defined as irregular patterns that are different from the mainstream. However, in real-world video surveillance scenarios, due to the difference of camera positions, sampling rates, and

⋆ This work is supported in part by NSFC (No. 61672089, 61273274, and 61572064), National Key Technology R&D Program of China 2012BAH01F03, the Fundamental Research Funds for the Central Universities 2017YJS043, the NSF CAREER Grant (No. 1149783), and gifts from Adobe and Nvidia. Cong Ma and Shaoyue Song are supported by a scholarship from China Scholarship Council.

scene structures, the obtained trajectories usually vary in temporal and spatial characteristics. Therefore, it is challenging to properly define and analyze trajectory properties and detect anomalous patterns. Furthermore, in some cases, we can obtain a video segment that contains only normal patterns for building regular models, and yet in other cases, we only have a test video to detect anomalies without any prior knowledge of normal patterns. A flexible method that facilitates anomaly detection in both cases is of great interest for numerous applications. Existing trajectory anomaly detection methods [13, 14, 19] usually construct statistical path models based on clustering to learn normal patterns and determine deviated samples as irregular ones. However, when the number of samples in the dataset is small, it is difficult to construct reliable statistical models. Other approaches [17, 18, 22] learn regular models based on the normal distributions and detect outliers. These algorithms usually rely on labeled data to estimate the distributions. Constructing reliable statistical models usually require a fair amount of data. In contrast, non-parametric models offer more flexibility when only scarce and rare samples are at our disposal. If we treat each normal sample as a single regular model, we can avoid this problem and detect the anomalies with a k-nearest neighbor based method, as shown in [16] and [9]. For unlabeled datasets, we can also measure the anomaly level of a trajectory by computing the distance to its nearest neighbor. A similar strategy is developed in [28]. However, the trajectory distances used in these methods are less sensitive to some anomalous patterns and thus limit performance and application scenarios.

In this paper, we use the simple but effective nearest neighbor method to deal with the trajectory anomaly detection problem in different settings. In order to obtain robust detection results in challenging scenes, we propose a new trajectory similarity measure based on an autoencoder constructed from recurrent neural networks (RNNs) to compute the distances between trajectories and facilitate the distance-based anomaly detector. Our contributions can be summarized as follows:

- We propose an RNN autoencoder based similarity measure for two, three or higher dimensional trajectory data.
- We demonstrate that the proposed similarity measure is effective for distinguishing various types of trajectory anomalies.
- We show that the proposed anomaly detection method performs favorably against the state-of-the-art algorithms in both supervised and unsupervised settings.

2 Related Work

2.1 Trajectory Anomaly Detection

In order to detect anomalous trajectories, numerous methods aim to learn a regular model first. Clustering based methods [13–15, 19] learn path models for normal patterns via grouping trajectory data in an unsupervised manner. Hu et al. [13] cluster trajectories based on spatial and temporal information using Gaussian distributions. For a trajectory sample, if the probabilities under known patterns are lower than a threshold, it is detected as an anomaly. Jiang et al. [14] represent trajectories with hidden Markov models (HMMs) and propose a dynamic hierarchical clustering method to learn normal patterns. HMMs are also used in [19] to model activity paths and evaluate the likelihoods of trajectory samples. However, it is difficult for these methods to handle the ambiguities that are clustered near the boundaries of normal and anomalous patterns. A few methods learn regular models when some labeled normal samples are available. Trajectories that deviate from the learned model are considered as anomalies. Li et al. [17] use the sparse reconstruction analysis and construct a dictionary based on normal samples. For a test trajectory, when the reconstruction error on the normal dictionary is lower than a threshold, it is identified as an anomaly. In [20] Piciarelli et al. learn the regular model with a one-class SVM and use the novelty detection method to find anomalies. Additionally, they propose an approach to tune the SVM parameters without labeled data. A recent method developed in [18] is based on the concept of tube and droplets. For anomaly detection, it first constructs thermal transfer fields based on the training set and then evaluates test samples by the derived droplet vectors. Nearest neighbor based methods are simple but usually effective for anomaly detection. Such approaches require few parameters and can be applied to both supervised and unsupervised settings. Yankov et al. [28] propose an efficient method based on the Euclidean distance to discover unusual samples in huge unlabeled datasets. This method performs favorably against the SVM-based scheme by Piciarelli et al. [20]. For online and sequential anomaly detection, Laxhammar [16] propose a detector based on the Hausdorff distance, nearest neighbor principle, and non-conformity measure. The detector can be used with or without supervisory signals. More recently, Ergezer and Leblebicioğlu [9] apply sparse representations to the nearest neighbor method based on the covariance trajectory descriptors and achieve favorable results. In these methods, it is critical to compute sample distances accurately. Therefore, trajectory similarity measures with better sensitivity to anomalies are likely to improve the detection performance and studied in this work.

2.2 Trajectory Similarity Measures

Numerous trajectory distances or similarity measures have been proposed in the literature. On one hand, warping based distance metrics take the temporal dimension into consideration. The simplest one is the Euclidean distance that averages the differences between each pair of corresponding points [21,30]. In this case, lengths of the two trajectories are required to be the same. In addition, the DTW [2] and LCSS [25] methods have been used for time sequences to measure similarities. When these metrics are applied to trajectory data, the distances between each time steps are computed based on the Euclidean distance. Furthermore, Chen et al. design ERP [4] and EDR [5] distance metrics for time series to support local time shifts and deal with data noises. On the other hand, shape-based distance metrics such as the Hausdorff [11] and Fréchet [1] measures have been applied to measure the similarities between spatial trajectory sequences. A more recent shape-based distance metric is the SSPD introduced in [3] for trajectory clustering. Different from these methods, a set of HMM-based distances are proposed in [21]. The distance between two trajectories is defined with the probabilities of each sample to be generated by the models learned on each other. Experiments in [21] show that such a model-based distance is able to handle more unrestricted trajectories and has strong discriminative abilities. However, distances based on more powerful trajectory models remain unexplored.

2.3 RNN-based Autoencoder

An autoencoder can be seen as a special encoder-decoder architecture where the target is the same as the input. The autoencoders based on RNNs are introduced in [6] and [24] to address the machine translation problem. The RNN autoencoders have also been used to construct representations for videos, such as in [23] and [27]. Hasan et al. [10] use convolutional autoencoders to learn regularity models in videos and detect anomalies. In addition, the spatio-temporal autoencoders have been developed for anomaly detection in [7]. These two approaches use autoencoders as a one-class learning approach and detect frame-level anomalies based on raw image data, rather than trajectories. In addition, Yao et al. [29] learn fixed-size representations of geographical trajectories with sequence-to-sequence autoencoders for clustering. However, to the best of our knowledge, autoencoders based on RNNs have not been applied to measure trajectory distances for anomaly detection.

3 Proposed Method

3.1 RNN Autoencoder based Trajectory Distance

Consider a trajectory T formed by a moving object in an image sequence, at each time step t the state of T is represented by the spatial coordinates (x, y). Thus we have the sequence notation $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$, where N is the time duration. In order to measure the similarity between two trajectories, we propose a model-based method.

We first train an RNN autoencoder model for each trajectory sample in a sequence-to-sequence manner. The autoencoder consists of an encoder and a decoder while the network input and the decoding target are identical. In the encoder part, we use a GRU-based RNN [8] to capture the dynamic characteristics of the input trajectory sequence. In practice, the state of the input sequence at each time step can be multi-dimensional. To emphasize the sequential information, we explicitly embed a time axis starting from 0 to the input sequence during training. Thus at each time step k, the state of a sequence is denoted as (x_k, y_k, t_k). The final hidden states of the encoder are fed to the decoder network as initial hidden states. In the decoder part, we use a GRU layer and a linear layer to reconstruct the input sequence along the time dimension. The two parts of the autoencoder model are trained together by minimizing the reconstruction error. Since the output sequence and the target are in the same dimensions, we use mean squared error (MSE) as the loss function. The training stage ends when the loss is small enough (we use a threshold of 10^{-5}). In addition, we normalize the data values of the input sequence to $[0, 1]$ for a better convergence performance. We define the distance between two trajectories based on the trained models. For a pair of trajectory sequences T_a and T_b, we denote the autoencoder models learned from each trajectory as M_a and M_b. When we take trajectory T_b as the input of model M_a, we compute the reconstruction error $e(T_b, M_a)$. Therefore, the one-way distance from T_a to T_b can be computed by:

$$d(T_a, T_b) = |e(T_b, M_a) - e(M_a)|, \tag{1}$$

where $e(M_a)$ denotes the final loss of model M_a during training. Conversely, the one-way distance $d(T_b, T_a)$ from T_b to T_a can be computed by $|e(T_a, M_b) - e(M_b)|$.

From the definition above we know that $d(T_a, T_b) > 0$ and $d(T_a, T_a) = 0$. However, due to the randomness and noise during model training, we cannot obtain $d(T_a, T_b) = d(T_b, T_a)$. To have a symmetric distance measure, we define the bidirectional dissimilarity between trajectory pair T_a and T_b as:

$$D(T_a, T_b) = |e(T_b, M_a) + e(T_a, M_b) - e(M_a) - e(M_b)|. \tag{2}$$

Since RNN models can process sequences with variable time steps, trajectories T_a and T_b need not be of the same lengths.

Furthermore, considering that the autoencoder model can be trained with a batch of samples at the same time, we can expand the application of this model to computing the distance from a cluster of trajectories C_T to a query trajectory T_q. Similarly, this one-way distance can be defined as:

$$D_C(C_T, T_q) = |e(T_q, M_C) - e(M_C)|, \tag{3}$$

where M_C denotes the model learned from the cluster of trajectories.

The set of distances defined above can be naturally applied to the nearest neighbor based anomaly detection depending on different data settings.

3.2 Distance based Anomaly Detection

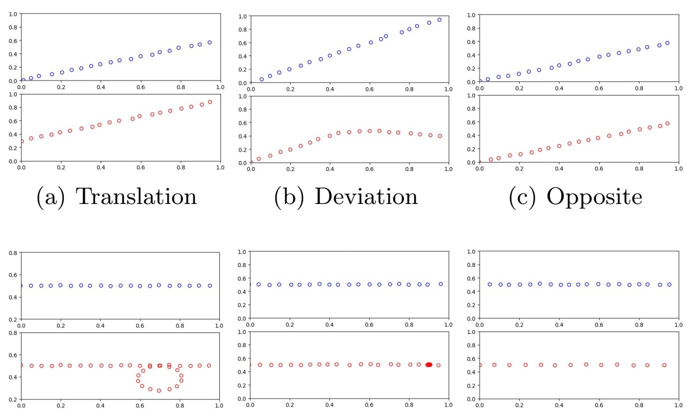
In trajectories obtained from surveillance videos, anomalies tend to show irregular motion patterns. However, it is difficult to differentiate between anomalies and samples with noise. When there is a training set indicating the normal patterns as a limited supervision, it is easier to define and detect the anomalies. Instead of the statistical methods that may require a lot of training samples, we use a nearest neighbor detector based on the proposed one-way autoencoder distance (Eq. 1). Considering each training sample as an individual normal model, the anomaly score can be defined by the directional distance of a test sample from its nearest training sample.

However, labeled data are not always available. In unsupervised scenes, researchers usually use novelty detection methods to find anomalies. A nearest neighbor based detector can still perform by measuring the deviation of a trajectory sample to its neighbors. We use the bidirectional distance definition (Eq. 2) in this case to obtain a symmetric distance matrix. Since the proposed trajectory similarity measure is sufficiently robust, we use the one-nearest neighbor distance as the anomaly score for a test trajectory.

4 Experimental Results

4.1 Comparison of Distances on Different Trajectory Patterns

In order to evaluate the sensitivity of a trajectory distance to different kinds of anomalies, we measure the similarities of several trajectory pattern pairs inspired by [21]. In a surveillance video scene, an anomalous object trajectory may be different from the normal ones in several aspects including position, orientation, and speed. Therefore, we list six pattern pairs including: *Translation*, *Deviation*, *Opposite*, *Loop*, *Wait*, and *Speed*, as shown in Fig. 1.



(a) Translation (b) Deviation (c) Opposite

(d) Loop (e) Wait (f) Speed

Fig. 1. Different pattern pairs illustrating various types of anomalies

For each pair of trajectories T_a and T_b, considering T_a as a normal sample and T_b as an anomaly, we compute the distance $D(a, b)$ between them. We then construct another pair of trajectories T_a and T_c, where T_c is built by adding T_a with a Gaussian noise. The distance $D(a, c)$ between this pair is computed as a baseline. Therefore, we can evaluate the distance sensitivity $S_{pattern}$ to a certain pattern by computing the ratio between $D(a, b)$ and $D(a, c)$ as:

$$S_{pattern} = D(a, b)/(D(a, c) + \epsilon), \tag{4}$$

where ϵ is the machine epsilon for avoiding division by zero. A ratio close to 1.0 indicates that it is difficult for the distance to distinguish a pattern from noises. On the other hand, a higher ratio value indicates that the similarity measure is more sensitive to this pattern.

We generate a synthetic dataset consisting of 100 pair examples for each pattern. The length of each normal trajectory sample is set to 50. Due to the pattern definition, in the last three cases *Loop*, *Wait*, and *Speed*, the lengths of anomalous samples are different from those of the normal ones. This inequality also contributes to the data variety. In addition, a Gaussian white noise is added to each sample for a better simulation of real situations.

We use nine trajectory distances to compare with the proposed method. They are: Euclidean, DTW [2], LCSS [25], ERP [4], EDR [5], Hausdorff [11], Fréchet [1], SSPD [3], and the HMM-based distance [21]. Experimental results are shown in Table. 1.

From the table we can know that most of the distances cannot identify the temporal anomalies *Wait* and *Speed*, while non-directional distances SSPD and Hausdorff (undirected) are not sensitive to the *Opposite* case. Among these, the ERP metric performs well over all the patterns. Compared to other distances, our method is sensitive to all the six anomalous pattern cases.

Table 1. Sensitivity scores of different trajectory distances on the six anomaly patterns.

Distances	Translation	Deviation	Opposite	Loop	Wait	Speed
Euclidean	42.47	54.73	82.42	15.10	5.34	1.10
DTW	40.88	54.73	82.42	6.86	1.51	1.38
SSPD	62.35	75.06	1.01	3.52	0.98	0.99
LCSS	10.12	8.00	9.12	1.00	0.94	1.04
EDR	10.04	8.00	9.04	1.78	2.57	3.63
ERP	42.47	54.73	56.06	30.29	47.99	35.73
Frechet	22.25	72.68	83.97	8.50	1.01	1.05
Hausdorff	22.08	69.36	1.00	10.71	1.01	1.05
HMM	750.64	2919.38	1249.74	98.86	2.43	1.34
Ours (One-way)	4679.26	10544.19	14865.79	1050.85	746.84	584.86
Ours (Bidirectional)	198125.56	693815.05	641080.44	44645.88	27131.89	20250.79

4.2 Anomaly Detection Performances

Based on the proposed trajectory distance, we implement the anomaly detection task with the nearest neighbor detector as described in Sec. 3.2. In order to

Table 2. Dataset properties: a brief view.

Datasets	Supervision	Dataset size	Scene number	Data source
CROSS [19]	Yes	Big	1	Synthetic
TRAFFIC [18]	Yes	Small	1	Realistic
VMT [12]	No	Big	1	Realistic
DETRAC [26]	No	Medium	31	Realistic

show the wide applicability of our method, we conduct experiments on four datasets that are under quite different configurations, as summarized in Table 2. Since the anomaly detection results vary with different threshold values, we use the Receiver Operating Characteristic (ROC) curve and the Area Under Curve (AUC) to evaluate the overall performance of the detector.

CROSS Dataset. This is a widely used dataset for motion pattern learning and anomaly detection under a cross-traffic scene, introduced in [19]. The training set contains 1900 traffic trajectory samples divided to 19 classes. The test set has 9700 samples while 200 of them are labeled as anomalous. In order to make a fair comparison and reduce noise effects, trajectory data in this and the following experiments are normalized to the same length of 12 points using least-squares cubic spline curves approximation [17, 22].

First, we compare the performances of several different trajectory distances using the same nearest neighbor anomaly detector. Since there exists a training set, we use the one-way distance as defined in Eq. 1. In order to make use of the class labels, we construct 19 normal models with the average trajectory of each class. The anomaly score is defined with the distance of a test sample from the nearest normal model. With the scores obtained, we can vary the threshold and draw ROC curves as shown in Fig. 2. The AUC scores are summarized in Table 3. The distances compared are the same as those in Sec. 4.1. We can see that the proposed distance metric performs favorably against the others when applied to anomaly detection.

Table 3. AUCs of anomaly detectors based on different trajectory distances on the CROSS dataset. The best result is in bold and the second best results are underlined.

Methods	Euclidean	DTW	SSPD	LCSS	EDR	ERP	Fréchet	Hausdorff	HMM	Ours
AUCs	<u>0.9985</u>	0.9976	0.9520	0.9916	0.9920	<u>0.9985</u>	0.9923	0.9961	0.8600	**0.9996**

Then, the comparison with several state-of-the-art methods is shown in Table 4. Since the available results in the literature are evaluated by the detection rates (DR) and abnormality false positive rates (FPR) [18, 19], we report our re-

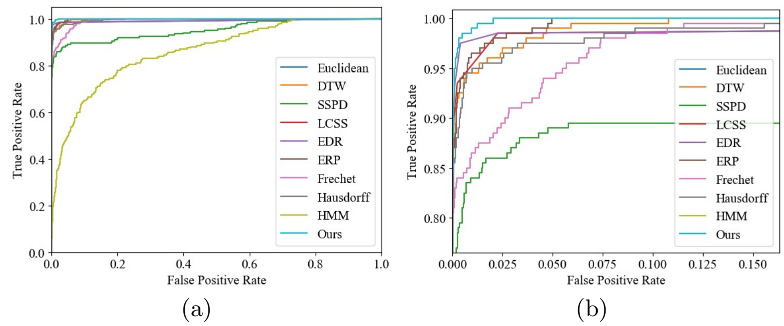


Fig. 2. ROC curves of anomaly detection based on different trajectory distances on the CROSS dataset. Sub-figure (b) is the zoomed out version of the left top part of (a)

sults under the same metrics. The results are obtained directly from [18]. From Table 4 we can know that our method achieves a significantly low false alarm rate while keeping a high detection rate. It is worth mentioning that the listed methods [12, 18, 19] are not specially designed for anomaly detection, so their results may not seem competitive.

Table 4. Comparison of anomaly detection results on the CROSS dataset.

Methods	DR (%)	FPR (%)
tDPMM [12]	91.0	23.3
3SHL [19]	85.5	23.5
3D Tube [18]	91.3	23.5
Ours	**96.0**	**9.9**

TRAFFIC Dataset. This dataset is introduced in a recent work [18]. It contains 300 trajectories collected from a real-world cross-traffic scene. As shown in [18], the small sample number, large data variations, and oblique camera view all make the TRAFFIC dataset challenging, especially for learning reliable normal models. We also use the nearest neighbor detector based on our RNN autoencoder distance measure to handle this situation. We keep the same experimental settings with [18], namely: equally dividing the dataset into training/test sets and running the random division for four times to obtain an average score. For each test trajectory, we compute the nearest one-way distance from the training samples as the anomaly score. Since the results are averaged, we do not present ROC curves. The average AUCs computed from different distances are shown in Table 5. From the experimental results, we can know the best performing distance metrics are Hausdorff, Fréchet and ours. This indicates that the anomalies in the TRAFFIC dataset are best distinguished by shape features.

Table 5. AUCs of anomaly detectors based on different trajectory distances on the TRAFFIC dataset.

Methods	Euclidean	DTW	SSPD	LCSS	EDR	ERP	Fréchet	Hausdorff	HMM	Ours
AUCs	0.9987	0.9987	0.9977	0.9937	0.9909	0.9987	**1.0000**	<u>0.9999</u>	0.9922	**1.0000**

VMT Dataset. In order to evaluate our method on more real-world data, we employ the Vehicle Motion Trajectory (VMT) dataset [12] that contains 1500 trajectory samples. Since the dataset is not labeled with anomalies, we manually label 24 anomalous trajectories that deviate from the others in spatial or shape features. Some examples of labeled anomalies are shown in Fig. 3. We do not label normal data and directly run our method in an unsupervised manner.

As discussed in Sec. 3.2, we use the symmetric version of the proposed distance metric (Eq. 2) in this experiment. The ROC curves of different trajectory distances are shown in Fig. 4 and AUCs are summarized in Table 6. Due to the fact that the anomalies in this dataset are more difficult to distinguish, we can see that the overall results are not so good, but the detector based on our distance metric still performs the best.

Fig. 3. Illustration of some anomalous trajectory samples in the VMT dataset

DETRAC Dataset. This is a video dataset originally introduced as a large-scale real-world benchmark for object detection and multi-object tracking [26]. We generate trajectories based on the tracking ground-truth of the training set. We divide the behaviors of vehicle trajectories to several types including *Driving through*, *Changing lane*, *Turn*, *Wait* and some ambiguities. In order to define a proper amount of anomalous samples, we label the trajectories with *Turn* and

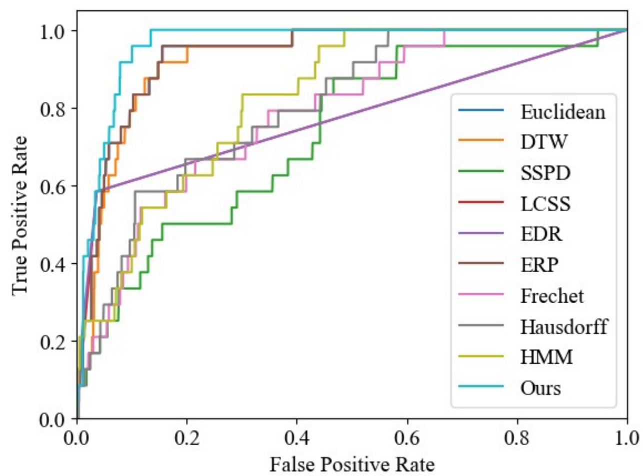


Fig. 4. ROC curves of anomaly detectors based on different trajectory distances on the VMT dataset.

Table 6. AUCs of anomaly detectors based on different trajectory distances on the VMT dataset.

Methods	Euclidean	DTW	SSPD	LCSS	EDR	ERP	Fréchet	Hausdorff	HMM	Ours
AUCs	0.9339	0.9261	0.7300	0.7734	0.7744	0.9339	0.7871	0.8049	0.8246	**0.9608**

Wait behaviors as anomalies. Finally, there are 31 scenes labeled with anomalies out of the 60 training videos. The number of trajectory samples in each scene ranges from 8 to 89 and there are 3 anomalies out of 43 trajectories in each scene on average. Therefore, the labeled trajectory dataset is challenging due to the sparse data, different vehicle behaviors and various scenes. For this dataset, we use the nearest neighbor based anomaly detector without the supervision of normal data. The AUCs of the nearest neighbor detector based on different distances are listed in Table 7. The results are averaged over the 31 scenes of the dataset. Overall, the HMM-based distance metric performs the best on this dataset. The Euclidean distance and our method also achieve comparable results. The experimental results above show that our method is suitable for

Table 7. AUCs of anomaly detectors based on different trajectory distances on the DETRAC dataset.

Methods	Euclidean	DTW	SSPD	LCSS	EDR	ERP	Fréchet	Hausdorff	HMM	Ours
AUCs	0.8956	<u>0.9044</u>	0.8981	0.8976	0.8979	0.8956	0.8557	0.8572	**0.9071**	0.9022

both supervised and unsupervised anomaly detection and sufficiently flexible to obtain satisfactory results on datasets with various properties. The results indicate the robustness and wide applicability of our RNN autoencoder based trajectory distance. All the source code and datasets will be made available to the public.

5 Conclusion

In this paper, we present trajectory distance metrics based on an RNN-based autoencoder to measure the similarity between trajectories and apply them to anomaly detection. Experimental results show that the proposed metrics are sensitive to anomalies and perform favorably in the trajectory anomaly detection task when incorporated in a nearest neighbor based detector. We also demonstrate an anomaly detector based on our metrics is flexible and effective under various dataset configurations. In practice, the proposed distance metrics can also be applied to other distance-based detectors. We will address these issues in our future work.

References

1. Alt, H., Godau, M.: Computing the fréchet distance between two polygonal curves. International Journal of Computational Geometry & Applications **5**, 75–91 (1995)
2. Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: Workshop on Knowledge Discovery in Databases (1994)

3. Besse, P., Guillouet, B., Loubes, J.M., François, R.: Review and perspective for distance based trajectory clustering. arXiv preprint arXiv:1508.04904 (2015)
4. Chen, L., Ng, R.: On the marriage of lp-norms and edit distance. In: International Conference on Very Large Data Bases (2004)
5. Chen, L., Ozsu, M.T., Oria, V.: Robust and fast similarity search for moving object trajectories. In: ACM SIGMOD International Conference on Management of Data (2005)
6. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
7. Chong, Y.S., Tay, Y.H.: Abnormal event detection in videos using spatiotemporal autoencoder. arXiv preprint arXiv:1701.01546 (2017)
8. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
9. Ergezer, H., Leblebicioğlu, K.: Anomaly detection and activity perception using covariance descriptor for trajectories. In: Hua, G., Jégou, H. (eds.) ECCV 2016 Workshops. LNCS. pp. 728–742. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48881-3_51
10. Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A.K., Davis, L.S.: Learning temporal regularity in video sequences. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)
11. Hausdorff, F.: Grundz uge der mengenlehre (1914)
12. Hu, W., Li, X., Tian, G., Maybank, S., Zhang, Z.: An incremental dpmm-based method for trajectory clustering, modeling, and retrieval. IEEE Trans. Pattern And. Mach. Intell. **35**(5), 1051–1065 (2013)
13. Hu, W., Xiao, X., Fu, Z., Xie, D., Tan, T., Maybank, S.: A system for learning statistical motion patterns. IEEE Trans. Pattern And. Mach. Intell. **28**(9), 1450–1464 (2006)
14. Jiang, F., Wu, Y., Katsaggelos, A.K.: A dynamic hierarchical clustering method for trajectory-based unusual video event detection. IEEE Trans. Image Process. **18**(4), 907–913 (2009)
15. Kumar, D., Bezdek, J.C., Rajasegarar, S., Leckie, C., Palaniswami, M.: A visual-numeric approach to clustering and anomaly detection for trajectory data. The Visual Computer **33**(3), 265–281 (2017)
16. Laxhammar, R., Falkman, G.: Online learning and sequential anomaly detection in trajectories. IEEE Trans. Pattern And. Mach. Intell. **36**(6), 1158–1173 (2014)
17. Li, C., Han, Z., Ye, Q., Jiao, J.: Visual abnormal behavior detection based on trajectory sparse reconstruction analysis. Neurocomputing **119**, 94–100 (2013)
18. Lin, W., Zhou, Y., Xu, H., Yan, J., Xu, M., Wu, J., Liu, Z.: A tube-and-droplet-based approach for representing and analyzing motion trajectories. IEEE Trans. Pattern And. Mach. Intell. **39**(8), 1489–1503 (2017)
19. Morris, B.T., Trivedi, M.M.: Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach. IEEE Trans. Pattern And. Mach. Intell. **33**(11), 2287–2301 (2011)
20. Piciarelli, C., Micheloni, C., Foresti, G.L.: Trajectory-based anomalous event detection. IEEE Trans. Circuits Syst. Video Technol. **18**(11), 1544–1554 (2008)
21. Porikli, F.: Trajectory distance metric using hidden markov model based representation. In: ECCV 2004, PETS Workshop (2004)
22. Sillito, R.R., Fisher, R.B.: Semi-supervised learning for anomalous trajectory detection. In: British Machine Vision Conference (2008)

23. Srivastava, N., Mansimov, E., Salakhudinov, R.: Unsupervised learning of video representations using lstms. In: International Conference on Machine Learning (2015)
24. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Neural Information Processing Systems (2014)
25. Vlachos, M., Kollios, G., Gunopulos, D.: Discovering similar multidimensional trajectories. In: IEEE International Conference on Data Engineering (2002)
26. Wen, L., Du, D., Cai, Z., Lei, Z., Chang, M., Qi, H., Lim, J., Yang, M., Lyu, S.: UA-DETRAC: a new benchmark and protocol for multi-object detection and tracking. arXiv preprint arXiv:1511.04136 (2015)
27. Yang, H., Wang, B., Lin, S., Wipf, D., Guo, M., Guo, B.: Unsupervised extraction of video highlights via robust recurrent auto-encoders. In: IEEE International Conference on Computer Vision (2015)
28. Yankov, D., Keogh, E., Rebbapragada, U.: Disk aware discord discovery: Finding unusual time series in terabyte sized datasets. In: IEEE International Conference on Data Mining. pp. 381–390 (2007)
29. Yao, D., Zhang, C., Zhu, Z., Huang, J., Bi, J.: Trajectory clustering via deep representation learning. In: International Joint Conference on Neural Networks (2017)
30. Zhang, Z., Huang, K., Tan, T.: Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. In: International Conference on Pattern Recognition (2006)