ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20230325000479
Original scientific paper

Pattern Anomaly Detection based on Sequence-to-Sequence Regularity Learning

Yuzhen CHENG*, Min LI

Abstract: Anomaly detection in traffic surveillance videos is a challenging task due to the ambiguity of anomaly definition and the complexity of scenes. In this paper, we propose to detect anomalous trajectories for vehicle behavior analysis via learning regularities in data. First, we train a sequence-to-sequence model under the autoencoder architecture and propose a new reconstruction error function for model optimization and anomaly evaluation. As such, the model is forced to learn the regular trajectory patterns in an unsupervised manner. Then, at the inference stage, we use the learned model to encode the test trajectory sample into a compact representation and generate a new trajectory sequence in the learned regular pattern. An anomaly score is computed based on the deviation of the generated trajectory from the test sample. Finally, we can find out the anomalous trajectories with an adaptive threshold. We evaluate the proposed method on two real-world traffic datasets and the experiments show favorable results against state-of-the-art algorithms. This paper's research on sequence-to-sequence regularity learning can provide theoretical and practical support for pattern anomaly detection.

Keywords: anomaly detection; autoencoders, sequence-to-sequence; vehicle trajectories

1 INTRODUCTION

Detecting anomalous vehicle behaviors from surveillance videos is an important task in traffic monitoring. Anomalies in the video are usually defined as events or behaviors that deviate from the mainstream, such as irregular routes, uncommon speeds, and entering restricted areas. Existing methods carry out anomaly detection based on different features. Some approaches [1-4] directly extract features from video frames and find anomalous patterns at the frame or region level, while others [5-10] detect unusual behaviors based on target trajectories. For the vehicles in traffic scenes, the anomalies that need to be detected are usually traffic violation or improper driving behaviors. Since the vehicle trajectories extracted from surveillance videos contain motion pattern information, they are good representations of vehicle routes and behaviors. Compared with features extracted from image pixels, the trajectory data is compact and computationally efficient, especially in large-scale long-term surveillance. At the trajectory level, some clutter in the spatial or temporal details of the scene can be alleviated. In addition, in recent years, the performance of object detection and visual tracking algorithms has been significantly improved and it is easier to obtain high-quality trajectories for motion analysis. Therefore, in this paper, we focus on detecting anomalous trajectories to analyze vehicle behaviors.

Existing anomalous trajectory detection approaches tend to find anomalous sequence samples based on particular sequence features. This is usually effective when processing sequences in specific application scenarios. However, the anomaly detection task based on traffic trajectories is challenging. On the one hand, the traffic rules may be different across scenes and due to the various positions, viewpoints, and intrinsic parameters of surveillance cameras, the criteria of anomalies may also be different. On the other hand, complicated motion patterns of moving targets and the complexity of scenes make it difficult to distinguish anomalous samples from normal ones. In addition, in real-world surveillance, label data is not always at our disposal. It would be best if the algorithm could directly detect anomalies from data containing anomalous samples.

In order to tackle the above-stated problems in anomaly detection based on trajectories, some researchers propose semi-supervised or unsupervised methods such as in [7, 8] and learn major normal motion patterns in the trajectory data. Samples that deviate from the regularities are detected as anomalies. These approaches are flexible and show promising results. While in other cases, when a dataset containing labeled normal samples is available, it is possible to learn regularity models with partial supervision and determine the samples that do not conform to the model as anomalies [6, 9, 10]. Overall, by learning the data regularity, these methods detect anomalies effectively without requiring the manual definition of an anomaly in specific scenes.

Existing methods for detecting anomalous trajectories usually use spatial coordinates or hidden Markov models to represent sequences. However, the performance of these representations might be limited in more complicated cases. The recent development of neural networks inspired us to study newer and more powerful sequence models for learning trajectory regularities. In this paper, we propose to detect anomalous trajectories via learning the regularity in vehicle trajectory data with a sequence-to-sequence auto encoder. Specifically, in order to capture the dynamic temporal features of the trajectory data, we construct a model based on recurrent neural networks (RNNs) under the encoder-decoder architecture with the constraint that the input and output data are identical trajectory sequences. The encoder and decoder in the model are trained together on all the samples to learn an overall data regularity. If a sample deviates from the learned model, we determine it as an anomaly. Under the assumption that anomalies are rare in real-world events, the proposed method can learn the normal patterns from unlabeled data, not requiring a training set labeled with normal samples.

In addition, there could be multiple cues in the trajectory anomaly definition. For example, a fast-moving vehicle may be considered as an anomaly (speeding), but the velocity of targets is not explicitly represented with the spatial coordinates. Therefore, we propose a new function to compute the difference between the output sequence and the input sequence, considering the spatial location, velocity, and direction of the moving targets in trajectories. When we use this function to compute the training loss, the model can be optimized in a better way. At the inference stage, this function is also used to evaluate the difference between the reconstructed trajectory and the input sample, thus making the proposed method sensitive to multiple cues. The main points of this paper can be summarized as follows:

·We present a sequence-to-sequence approach based on RNNs under the autoencoder architecture to detect anomalous vehicle trajectories.

·We propose a new function that takes target location, velocity, and direction into account to compute the training loss and evaluate the anomaly levels.

·We demonstrate that the proposed method can learn regularities from unlabeled real-world data and performs favorably against the state-of-the-arts on anomaly detection in large-scale trajectory datasets.

2 RELATED WORK

Anomaly detection in videos has been one of the popular research topics in video analysis over the past decade. A standard solution for detecting anomalies is constructing regularity models based on normal data and detecting deviated samples. There are mainly two different approaches to the modeling and evaluating of video regularities. One is analyzing the low-level motion and appearance features extracted from local video data and the other is modeling the patterns of moving target trajectories.

Based on local video features, Kim and Grauman [1] utilize mixtures of probabilistic PCA to learn the normal patterns and detect anomalies. Xu et al. [11] train one-class SVM models on deep representations and detect outliers. Feng et al. [12] construct deep Gaussian mixture models for normal event patterns. In addition, sparse coding analysis is used in [2, 3] for representing regularities. Autoencoders are also applied to learn normal models based on visual features [4, 13]. The anomalies are determined by reconstruction errors.

For trajectories, Sillito and Fisher [6] propose an incremental algorithm based on Gaussian mixture models to learn normal patterns. Piciarelli et al. [5] use one-class SVMs to detect outliers in trajectory data. Jiang et al. [7] present a dynamic hierarchical clustering method based on hidden Markov models (HMMs) and detect anomalous trajectories. HMMs are also used for sequence modeling in [8, 14]. Moreover, similar to the methods based on local video features, the sparse representation of normal dataset is computed in [9, 10] and reconstruction errors are used for anomaly detection.

In addition, anomalous trajectories can also be detected based on other algorithms without learning an overall regularity model. Distance-based novelty detection approaches such as [15, 17, 18] use the nearest neighbor based detectors and can deal with unlabeled datasets. Other methods such as Gaussian process regression flow [19], incremental Dirichlet process mixture model [20], and 3D-tube-based model [21] focus on trajectory analysis and propose effective representations for trajectory classification, clustering, prediction or retrieval. These methods present applications of anomaly detection based on the conformity to their trajectory models.

In the aspect of sequence modeling, recurrent neural networks have been popular in recent years. The sequence-to-sequence learning is proposed by [22, 23]. They introduce an encoder-decoder structure based on RNNs and have been successful in areas of machine translation [22, 23], speech recognition [24], and so on. Auto encoders can be viewed as a special encoder-decoder architecture which reconstructs input to the output. This structure helps to learn compact representations of high-dimensional data. Yao et al. [25] propose to learn deep sequence representations with sequence-to-sequence auto encoders for trajectory clustering. Ma et al. [18] use auto encoders based on RNNs to compute the distances between trajectory samples and perform distance-based anomaly detection. Different from the above methods, we do not compute distances between specific sequence samples or learn normal patterns from clean, labeled datasets, but learn the regularity from all the trajectory data based on a sequence-to-sequence auto encoder and estimate the anomaly scores. Detailed method description is in the next section.

3 PROPOSED METHOD

Raw trajectory data are usually sequences of spatial coordinates. In surveillance videos, a trajectory can be formed by associating the pixel coordinates of target centroids across frames. This is carried out via tracking algorithms. For each time step, we denote the state of the target as $s = (x, y, t)$, where x and y are spatial coordinates and t denotes the temporal axis. Therefore, a trajectory can be represented as $T = s_1, s_2, \ldots, s_t, \ldots, s_L$, where t is still for time and L is the sequence length. To suppress noises, we use the Least-squares Cubic Spline Curves Approximation (LCSCA) to smooth and normalize the trajectories to a fixed length, as in [6, 9]. Finally, a trajectory dataset consisting of N samples has the shape $(N; L; 3)$.

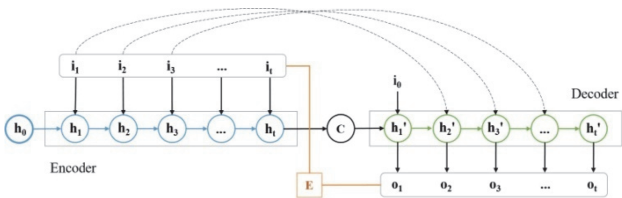


Figure 1 Illustration of the model structure. ***C*** is the context vector and *E* is the reconstruction error. The dashed lines represent the teacher-forcing strategy

A. Regularity Learning.

We construct an encoder-decoder sequence to sequence model. The model structure is illustrated in Fig. 1. Both the encoder and decoder are RNNs for processing sequence data. Specifically, the encoder is a single-layer GRU-based RNN [22] that summarizes the input trajectory and encodes it to a fixed length context vector c. The update of hidden states of the encoder RNN at each time step t can be represented as:

$$h_t = f\left(h_{t-1}, i_t\right) \tag{1}$$

where i_t is the input vector at time t. The hidden state h_L at the final time step is taken as the context vector c.
Then, we use another GRU-based RNN to decode the representation c. The decoder is initialized with the context vector as the hidden state and a zero vector as the input. In the forwarding procedure through time, we denote i as the input vector and o as the output vector. At each time step t, the decoder cell takes the output o_{t-1} of last step $t-1$ as input i_t and produces the next output vector. This can be described by:

$$h_t = f\left(h_{t-1}, o_{t-1}, c\right) \tag{2}$$

However, errors may accumulate in the sequential iteration. To obtain more reliable information from the trajectory data, we apply the teacher-forcing technique at the training stage, using the ground-truth T_{t-1} instead of o_t as the input i_t. Therefore, the update of hidden parameters in the decoder can be described as:

$$h_t = f\left(h_{t-1}, T_{t-1}, c\right) \tag{3}$$

For an autoencoder model, the output data can be seen to be reconstructed from the input sample by the model. Therefore, the autoencoder is usually trained with the loss function of a defined reconstruction error, for example, mean squared error (MSE), between the input and output sequences and force them to be identical. Generally, we define the reconstruction error as:

$$error = dist\left(\ T - T'\right) \tag{4}$$

where T denotes the input sequence and T' is the reconstructed sequence.

Finally, at the inference stage, we feed the test samples to the learned model and compute an anomaly score for each sample based on the reconstruction *error*. For an input sequence T, the model generates sequence T' from a zero input vector starting at time t_0. Intuitively, the more trajectory T deviates from the major patterns in data, the more sequence T' will be different from T, since it is generated based on the learned regularity.

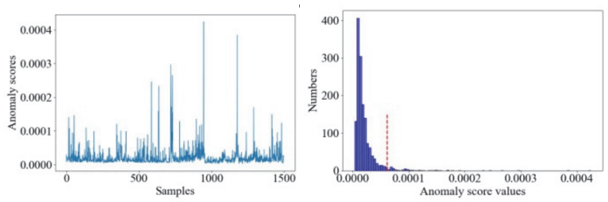


Figure 2 Illustration of the detection decision based on analyzing the anomaly scores. Sub-figure (a) is the scores computed on the VMT dataset and (b) shows the histogram of scores. The dashed line shows the threshold found by the three-sigma rule. Samples with scores at the right side of the line are considered as anomalies

B. Anomaly Detection.

With the anomaly scores for each sample, we use a threshold to determine if the trajectory is identified as an anomaly. However, in numerous cases, we need to deal with trajectories obtained from surveillance videos without labeled normal data and the threshold is scene-specific. Considering that anomalies are rare in the real world, we assume that the anomalous samples which need to be detected are sparse in the whole dataset. Therefore, we compute an adaptive threshold by analyzing the 1D sequence of anomaly scores.

The anomalies in the dataset usually have uncommon high score values. Identifying them by anomaly scores can be seen as another novelty detection problem. We first compute the mean value and standard variation of scores (trimmed by removing the 2% extreme high values) and then utilize the three-sigma rule to find the upper bound of normal scores as the threshold. Anomaly scores on the VMT dataset and their histogram are shown in Fig. 2. The computed threshold is depicted as a red dash line in the figure. Samples with scores over that threshold are determined as anomalies.

C. Reconstruction Error.

Specifically, for the application to anomalous vehicle trajectory detection, we can evaluate the reconstruction error with more cues. In addition to the spatial location, we propose to take the motion velocity and direction into account and define a new function to measure the differences between two trajectory sequences. First, the difference e_{loc} between spatial locations is defined as:

$$e_{loc} = \left\| T - \hat{T} \right\|_2^2 = \frac{\sum_i^L \sqrt{\left(x_i - \hat{x}_i\right)^2 + \left(y_i - \hat{y}_i\right)^2}}{L} \tag{5}$$

Second, since the trajectories are obtained in surveillance videos, each time step of the sequence corresponds to a frame in the video. We can compute the instant velocity v_t at each time t with:

$$v_t = \frac{\sqrt{\left(x_{t+1} - x_t\right)^2 + \left(y_{t+1} - y_t\right)^2}}{\Delta time} \tag{6}$$

As such, we can obtain 1D velocity sequences V and $\hat{V}$ from the input and output sequences, respectively. Since the time gap $\Delta time$ is same between two consecutive frames in the video, we can omit the time gap and represent the velocity error e_{vol} as:

$$e_{vol} = \left\| V - \hat{V} \right\|_2^2 = \frac{\sum_i^{L-1} \sqrt{\left(v_i - \hat{v}_i\right)^2}}{L-1} \tag{7}$$

Third, we consider the change of direction between two consecutive time steps. Fig. 3 shows the two displacement vectors across three consecutive time steps. The direction of target motion at time t can be represented by the angle θ_t between these two vectors.

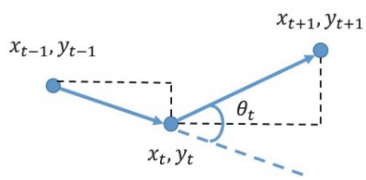


Figure 3 The change of motion direction of the target at time *t*, which can be represented by the angle between the two vectors before and after time *t*

Therefore, we can compute the change of target motion direction at time t with the cosine formula:

$$\theta_t = \\ = \frac{(x_t - x_{t-1})\cdot(x_{t+1} - x_t) + (y_t - y_{t-1})\cdot(y_{t+1} - y_t)}{\sqrt{(x_t - x_{t-1})^2 + (x_{t+1} - x_t)^2}\cdot\sqrt{(y_t - y_{t-1})^2 + (y_{t+1} - y_t)^2}} \quad (8)$$

For a sequence of spatial coordinates, the changes of direction at each time step can also form a sequence of 1-D direction changes Θ. Therefore, the difference between input and output sequences in terms of direction changes e_{ang} can be measured as follows:

$$e_{ang} = \left\|\Theta - \hat{\Theta}\right\|_2^2 = \frac{\sum_i^{L-2}\sqrt{(\Theta_i - \hat{\Theta}_i)^2}}{L-2} \quad (9)$$

Finally, we can combine the three Eq. (4), Eq. (6), and Eq. (8) as:

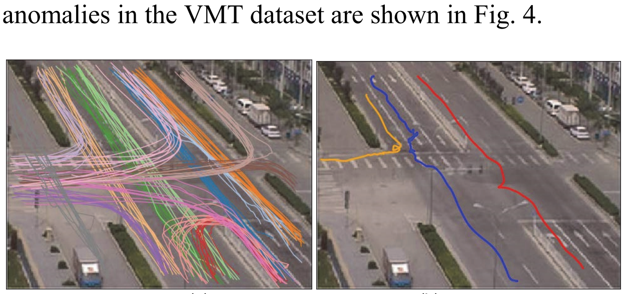
$$error = e_{loc} + e_{vel} + e_{ang} \quad (10)$$

This is the proposed criterion of differences between the input and output sequences. We use it to compute the reconstruction error, which is also the definition of anomaly score e.

4 EXPERIMENTAL VERIFICATION

A. Datasets.

In order to evaluate the proposed method, we conduct experiments on two real-world datasets and make comparison with existing algorithms. We use two trajectory datasets that are collected by tracking vehicles from real-world traffic surveillance videos. The first one is the Vehicle Motion Trajectory (VMT) dataset. It is introduced in [20] and is publicly available. It contains 1500 trajectories of vehicles monitored at a road junction, the routes include turning left and right, going through, and making U-turns. The ground-truth anomaly annotation comes from [18], where there are 24 anomalous behaviors containing waiting in the road, twisted trajectories, and uncommon routes. Examples of the normal samples and anomalies in the VMT dataset are shown in Fig. 4.



(a) (b)

Figure 4 Illustration of the VMT dataset. Sub-figure (a) depicts the traffic scene with normal samples while (b) shows three anomalous trajectories

The other dataset is the vehicle trajectories in the DETRAC dataset [26]. This is a large-scale traffic dataset containing 10 hours of videos as a public benchmark for object detection and multi-target tracking. We use the vehicle trajectories collected from the labeled tracking results and the ground-truth anomaly annotations in [18]. There are 31 scenes that contain trajectories labeled as anomalies, providing a variety of scene configurations. The dataset contains 1348 trajectory samples in total and 103 of them are labeled as anomalies. The anomalous patterns annotated are irregular routes and uncommon longtime stopping. Samples in several scenes of the DETRAC dataset are shown in Fig. 5. Neither of the datasets contains training data. Challenges include the various motion patterns, unorganized anomaly behaviors, and clutters and noises from real scenes.

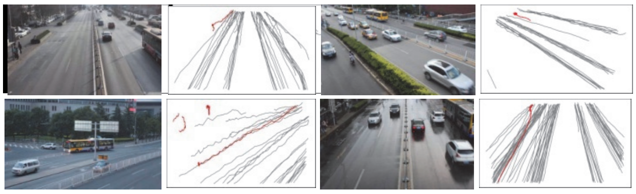


Figure 5 Illustration of the image views and trajectory samples of four scenes in the DETRAC dataset. Anomalous trajectories are in red

B. Experimental Setup.

We implement the proposed method with the PyTorch framework in Python 3. All the experiments run on an Ubuntu 16.04 workstation with 2.2 GHz CPU, NVIDIA-P40 GPU, and 64G memory. The use of the GPU significantly accelerates the computation procedure. As stated in previous sections, the input trajectories are normalized to a length of 12 with the Least-squares cubic spline curves approximation method. The values in trajectory data are scaled to [0, 1] so that the model could converge better. For our model, we set the size of hidden layers as 50 in both encoder and decoder networks. We use a learning rate at 0:001 and the batch size at 200 during training.

C. Anomaly Detection Performance.

Since the anomaly detection task can be viewed as a binary classification problem, we consider the anomalies as positive samples and normal trajectories as negative samples. Then, we can first evaluate the method performance with detection rate and false alarm rate, as in [8, 21]. Since we focus on unsupervised regularity learning, we compare the proposed method with algorithms that are able to run without normal data labels. The compared algorithms are the sequence-model based method (denoted by HMM in the result tables), the one-class SVM based method [5] (denoted by OCSVM), the sparse-reconstruction-based method (denoted by Sparse) and the distance-based methods (Euclidean and RNN-based [18] distances, denoted by Dist-Euc and Dist-RNN). All these methods can run on unlabeled datasets. The results are shown in Tab. 1.

From the table, we can see that the proposed method outperforms the other algorithms in both metrics. However, we notice that the detection rates of classical regularity learning methods (such as the HMM and one-class SVM method) are quite low. This may be due to the definition of anomalies in the test trajectory set. In real-world vehicle trajectories, the anomalies tend to be uncommon driving behaviors and are not significantly different from normal ones in routes or locations. Therefore, the detection task is of great challenge. In addition, the selection of thresholds will influence the detection performance. In order to control the variables, we use the single-peak statistical adaptive threshold method introduced in the previous section to detect anomalies in the anomaly scores obtained by different models. However, the distributions of anomaly values estimated by different models may be different, and the thresholds may need to be further adjusted to achieve the best results.

Table 1 Performance Comparison of Anomaly Detection Methods on The VMT and DETRAC Dataset. The Best Results are in Bold

Datasets	Methods / Metrics	HMM	OCSVM	Sparse	Dist-Euc	Dist-RNN	Ours
VMT	Detection Rate	0.1250	0.1250	0.1250	0.4167	0.5833	**0.9167**
	False Alarm Rate	0.0738	0.0589	0.0949	0.0332	0.0393	**0.0318**
DETRAC	Detection Rate	0.1637	0.1821	0.1759	0.3865	0.4522	**0.5122**
	False Alarm Rate	0.0269	0.0257	0.0392	0.0353	0.0387	**0.0247**

Therefore, in order to carry out a more comprehensive evaluation, we change the thresholds and use the Receiver Operation Characteristic (ROC) curve and the Area Under Curve (AUC) value to make further comparisons. The anomaly scores produced by the compared methods are used for evaluation.

Table 2 AUCS Of Detection Methods on the VMT and DETRAC Dataset. The Best Results are in Bold

Methods	HMM	OCSVM	Sparse	Dist-Euc	Dist-RNN	Ours
VMT	0.5642	0.5163	0.6089	0.9339	0.9608	**0.9925**
DETRAC	0.7279	0.8959	0.7598	0.8956	0.9022	**0.9342**

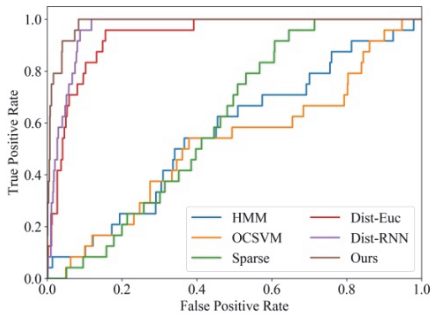


Figure 6 ROC curves of anomaly detection algorithms on the VMT dataset

We present the AUC values on both datasets in Tab. 2 and draw the ROC curves on the VMT dataset as in Fig. 6. We can see that the proposed method still performs favorably against the other algorithms. Since the DETRAC dataset contains 31 scenes, we do not provide detailed ROC curves for each subset. In addition, we can know from the table that the distance-based methods perform better than the methods based on classical regular models. A possible reason may be that it is difficult for the regular models to distinguish the anomalous samples from the normal ones in the real-world data, while the distance based methods are more sensitive to small differences between samples. However, our model shows higher performance. This may benefit from the strong nonlinear-fitting ability of the recurrent neural networks. The overall performance on the real-world traffic datasets with multiple scenes shows the wide applicability of our method.

Table 3 Comparisons Between the Proposed Method and The Baseline on the VMT and DETRAC Dataset. The Best Results are in Bold

Datasets	Methods	Detection Rate	False Alarm Rate	AUC
VMT	Baseline	0.7917	0.0332	0.9841
	Proposed	0.9167	0.0318	0.9925
DETRAC	Baseline	0.5032	0.0183	0.9116
	Proposed	0.5122	0.0247	0.9342

Furthermore, in order to evaluate the contribution of the proposed reconstruction error function (Eq. (9)) to the regularity learning and anomaly detection, we implement a baseline method that uses the basic Mean Squared Error (MSE) function to compute training loss and reconstruction errors. Other settings for the model are kept the same. Comparisons between the proposed method and the baseline on detection rate, false alarm rate, and the AUC values on both datasets are shown in Tab. 3.

We can know from the table that, in general, the proposed reconstruction error function can lead to more accurate anomaly detection results. This indicates that the proposed function is more effective than MSE loss and more sensitive to anomalous vehicle trajectories.

D. Limitations and future work.

However, the proposed method cannot always obtain favorable results without limitations. Since we consider traffic trajectory features including location, velocity, and direction, the proposed method cannot deal with anomalous behaviors that cannot be described in these ways. Other sequence describing dimensions, such as duplicates and periodic patterns, can be explored in future work.

5 CONCLUSION

In this paper, we propose to detect anomalous trajectories for vehicle behavior analysis via learning regularities in data. We utilize a sequence-to-sequence model under the auto encoder architecture to capture the main motion patterns in unlabeled data. The anomaly scores of samples are computed based on the conformity to the learned regularity model. In addition, a new function for the reconstruction error computation is proposed to deal with various anomalous trajectory patterns. Experiments on two large-scale real-world traffic datasets show that our method achieves favorable results against other algorithms and shows promising capabilities for detecting anomalous vehicle behaviors in wide application scenarios.

Acknowledgment

The authors would like to thank all the editors and reviewers for their time, efforts, and constructive comments.

6 REFERENCES

[1] Kim, J. & Grauman, K. (2009). Observe locally, infer globally: a space-time MRF for detecting abnormal activities with incremental updates. *IEEE Conference on Computer*

Vision and Pattern Recognition. 2921-2928. https://doi.org/10.1109/CVPR.2009.5206569

[2] Cong, Y., Yuan, J., & Liu, J. (2011). Sparse reconstruction cost for abnormal event detection. *IEEE Conference on Computer Vision and Pattern Recognition*. 3449-3456. http://dx.doi.org/10.1109/CVPR.2011.5995434

[3] Lu, C., Shi, J., & Jia, J. (2013). Abnormal event detection at 150 fps in Matlab. *IEEE International Conference on Computer Vision*, 2720-2727. https://doi.org/10.1109/ICCV.2013.338

[4] Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A. K., & Davis, L. S. (2016). Learning temporal regularity in video sequences. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 733-742. https://doi.org/10.1109/CVPR.2016.86

[5] Piciarelli, C., Micheloni, C., & Foresti, G. L. (2008). Trajectory-based anomalous event detection. *IEEE Transactions on Circuits and Systems for video Technology*, *18*(11), 1544-1554. https://doi.org/10.1109/TCSVT.2008.2005599

[6] Sillito, R. R. & Fisher, R. B. (2008). Semi-supervised Learning for Anomalous Trajectory Detection. *BMVC*, *1*, 035-1. https://doi.org/10.5244/C.22.103

[7] Jiang, F., Wu, Y., & Katsaggelos, A. K. (2009). A dynamic hierarchical clustering method for trajectory-based unusual video event detection. *IEEE Transactions on Image Processing*, *18*(4), 907-913. https://doi.org/10.1109/TIP.2008.2012070

[8] Morris, B. T. & Trivedi, M. M. (2011). Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach. *IEEE transactions on pattern analysis and machine intelligence*, *33*(11), 2287-2301. https://doi.org/10.1109/TPAMI.2011.64

[9] Li, C., Han, Z., Ye, Q., & Jiao, J. (2013). Visual abnormal behavior detection based on trajectory sparse reconstruction analysis. *Neuro computing*, *119*, 94-100. https://doi.org/10.1016/j.neucom.2012.03.040

[10] Chen, Z. J., Wu, C. Z., Zhang, Y. S., Huang, Z., Jiang, J. F., Lyu, N. C., & Ran, B. (2016). Vehicle behavior learning via sparse reconstruction with $\ell_ {2}-\ell_ {p} $ minimization and trajectory similarity. *IEEE Transactions on Intelligent Transportation Systems*, *18*(2), 236-247. https://doi.org/ 10.1109/TITS.2016.2587814

[11] Xu, D., Ricci, E., Yan, Y., Song, J., & Sebe, N. (2015). Learning deep representations of appearance and motion for anomalous event detection. *arXiv preprint arXiv:1510.01553*. https://doi.org/10.5244/C.29.8

[12] Feng, Y., Yuan, Y., & Lu, X. (2017). Learning deep event models for crowd anomaly detection. *Neuro computing*, *219*, 548-556. https://doi.org/ 10.1016/j.neucom.2016.09.063

[13] Zhao, Y., Deng, B., Shen, C., Liu, Y., Lu, H., & Hua, X. S. (2017). Spatio-temporal autoencoder for video anomaly detection. *Proceedings of the 25th ACM international conference on Multimedia*, 1933-1941. https://doi.org/10.1145/3123266.3123451

[14] Biswas, S. & Babu, R. V. (2017). Anomaly detection via short local trajectories. *Neuro computing*, *242*, 63-72. https://doi.org/10.1016/j.neucom.2017.02.058

[15] Yankov, D., Keogh, E., & Rebbapragada, U. (2008). Disk aware discord discovery: finding unusual time series in terabyte sized datasets. *Knowledge and Information Systems*, *17*, 241-262. https://doi.org/10.1109/ICDM.2007.61

[16] Laxhammar, R. & Falkman, G. (2013). Online learning and sequential anomaly detection in trajectories. *IEEE transactions on pattern analysis and machine intelligence*, *36*(6), 1158-1173. https://doi.org/10.1109/TPAMI.2013.172

[17] Ergezer, H. & Leblebicioğlu, K. (2016). Anomaly detection and activity perception using covariance descriptor for trajectories. *Computer Vision-ECCV 2016 Workshops: Amsterdam, Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II 14*, 728-742. Springer International Publishing. https://doi.org/10.1007/978-3-319-48881-3_51

[18] Ma, C., Miao, Z., Li, M., Song, S., & Yang, M. H. (2019). Detecting anomalous trajectories via recurrent neural networks. *Computer Vision-ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2-6, 2018, Revised Selected Papers, Part IV*, 370-382. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-20870-7_23

[19] Kim, K., Lee, D., & Essa, I. (2011). Gaussian process regression flow for analysis of motion trajectories. *2011 International Conference on Computer Vision*, 1164-1171. https://doi.org/10.1109/ICCV.2011.6126365

[20] Hu, W., Li, X., Tian, G., Maybank, S., & Zhang, Z. (2013). An incremental DPMM-based method for trajectory clustering, modeling, and retrieval. *IEEE transactions on pattern analysis and machine intelligence*, *35*(5), 1051-1065. https://doi.org/10.1109/TPAMI.2012.188

[21] Lin, W., Zhou, Y., Xu, H., Yan, J., Xu, M., Wu, J., & Liu, Z. (2016). A tube-and-droplet-based approach for representing and analyzing motion trajectories. *IEEE transactions on pattern analysis and machine intelligence*, *39*(8), 1489-1503. https://doi.org/10.1109/TPAMI.2016.2608884

[22] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*. https://doi.org/10.3115/v1/D14-1179

[23] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, *27*.

[24] Chiu, C. C., Sainath, T. N., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z., & Bacchiani, M. (2018). State-of-the-art speech recognition with sequence-to-sequence models. *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 4774-4778. https://doi.org/10.1109/ICASSP.2018.8462105

[25] Yao, D., Zhang, C., Zhu, Z., Huang, J., & Bi, J. (2017). Trajectory clustering via deep representation learning. *2017 international joint conference on neural networks (IJCNN)*, 3880-3887. https://doi.org/ 10.1109/IJCNN.2017.7966345

[26] Wen, L., Du, D., Cai, Z., Lei, Z., Chang, M. C., Qi, H., & Lyu, S. (2020). UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. *Computer Vision and Image Understanding*, *193*. https://doi.org/10.1016/J.CVIU.2020.102907

Contact information:

Yuzhen CHENG
(Corresponding author)
Beijing Jiaotong University,
No.3, Shangyuan Village, Haidian District, Beijing, China
E-mail: yzcheng@bjtu.edu.cn

Mi LI
North China University of Technology,
No.5, Jinyuanzhuang Road, Shijingshan District, Beijing,China
E-mail: limin@ncut.edu.cn