Research on Detection Method of Abnormal Trajectory of Port Operation Vehicles

Shuaihui Li
School of Computer Science and Technology,
Wuhan University of Technology
Wuhan, China
Email: wiki_hui@qq.com

Hongxing Liu,Hanbing Yao
Hubei Provincial Key Laboratory of Transportation
Internet of Things Technology
Wuhan, China
Email: liuhongxing@whut.edu.cn,
yaohb@whut.edu.cn

Abstract*—Port operation vehicles are mainly responsible for the transshipment of goods. If there is inadequate supervision in the process of transshipment it is easy to cause such behaviors as cargo leakage, cargo theft and illegal parking of drivers, which cause economic losses to the port. In order to discover such behaviors in time, the unsupervised anomaly detection algorithm Self-encoder-based Deep Feature Fusion Model(S-DFFM) is proposed to judge whether the trajectory of operation vehicles is abnormal or not. The method comprehensively characterizes the trajectory by fusing the shallow features of low-dimensional trajectory and the deep features of high-dimensional trajectory, which frees the trajectory from the limitation of spatial attributes. The experimental data adopts the real trajectory data of one month (7,547 trips) of operating vehicle trajectories of a port in Chongqing, and the experimental results show that S-DFFM can better represent the trajectory features, and the accuracy of trajectory abnormality detection using S-DFFM is as high as 96%.

***Keywords*—Trajectory; Anomaly detection; Feature fusion; Unsupervised detection;**

I. Introduction

The port is the hub center of water and land transportation, and the loading and unloading operation of the port is usually the transit of goods by operating vehicles. In the process of transshipment, due to the complex roads in the port, various operation types, harsh yard environment, and uncontrollable human factors, the process is prone to some abnormal operation behaviors such as cargo leakage (cargo not passing through the weighing equipment), cargo theft (transferring cargo from owner A to owner B's yard), and illegal parking (eating snack during operation). The operating vehicle will generate massive trajectory data in the process of operation, and by analyzing the trajectory data[1] and detecting the abnormal trajectory, the real behavior pattern of the operating vehicle hidden in the normal operation mode can be found, and the abnormal intention of the operating vehicle operation can be exposed, so that the supervisors can make corresponding measures.

Knorr et al[2] transformed the trajectory into some representative features consisting of position, direction and velocity, instead of just viewing the trajectory as a series of points, and then later detected the trajectory anomaly by comparison of distances. This method only compares the overall characteristics of the trajectory and ignores the local characteristics, and the problems exposed by this method become more and more obvious as the trajectory becomes longer. To solve the problems caused by comparing whole trajectories, Lee et al[3] designed a division and detection framework and proposed the TRAOD(Trajectory Outlier Detection) algorithm accordingly.Zhu et al[4] proposed a time-dependent popular path trajectory outlier detection algorithm TPRO(Time-dependent Popular Routes), which takes into account temporal anomalies but requires a lot of time computation during preprocessing.Laxhammar et al[5] proposed an online supervised trajectory anomaly detection algorithm SHNN-CAD(Sequential Hausdorff Nearest-Neighbor Conformal Anomaly Detector), which applies a consistent variance detector[6] to calculate the statistical confidence values of trajectories.

In this paper, considering the limitations of existing distance or density-based algorithms and the special characteristics of in-port trucking operations (some road networks are dense, operation types and operation goods are diverse, there are stationary and mobile states in operation links, and operation drivers can be changed, etc.),we propose the Self-encoder-based Deep Feature Fusion Model (S-DFFM), which converts shallow feature sequences into deep feature sequences and splices shallow feature sequences with deep feature sequences into fused feature sequences to comprehensively represent the features of trajectory segments from both deep and shallow layers.In the anomaly detection stage, the anomalous trajectory segments are extracted by comparing the similarity indices between the fused feature sequences, and then the percentage of the anomalous trajectory segments on the trajectory as a whole is considered to finally detect the anomalous trajectory.

II. Operating Vehicle Trajectory Feature Extraction And Conversion

Deep neural networks have been shown to have a strong ability to learn verbal data[7], textual data[8], and image data[9]. Trajectory data is three-dimensional data consisting of two-dimensional coordinates and timestamps, and it has been shown through empirical studies that it is not feasible to simply input three-dimensional data of the original trajectory into

2473-3636/21/$31.00 ©2021 IEEE
DOI 10.1109/DCABES52998.2021.00023

deep learning algorithms[10]. Therefore, in order to make the trajectory data suitable for deep neural network algorithms, trajectory features with stronger semantic expressions need to be extracted from the original trajectory data and the original 3D trajectory data is replaced by a representation of feature sequences.

As shown in Fig.1, the core of feature extraction and transformation is divided into two steps. First, the geometric features are extracted from a number of original trajectory fragments of variable length after division, and the geometric features are transformed into a shallow feature sequence F of fixed length after dimensional expansion; second, the shallow feature sequence is transformed into a fused feature sequence FM using a deep feature fusion model based on self-encoder, i.e., the shallow feature sequence F is stitched with the deep trajectory feature sequence FD to obtain the fused feature sequence FM.

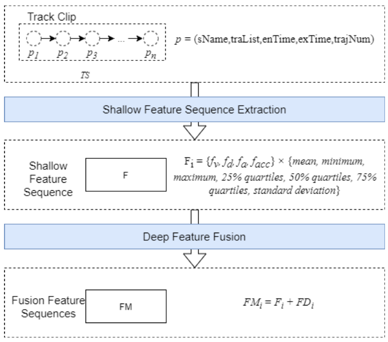

Fig. 1. Feature Extraction and Transformation.

A. Shallow Feature Extraction

Definition 1-1. Shallow feature sequence: the geometric features of the expanded trajectory are called the shallow feature sequence of the trajectory, and the set Q composed of n shallow trajectory feature sequences is called the set of shallow feature sequences, denoted as $Q = \{F_1, F_2, F_3 \ldots F_n\}$,where each shallow feature sequence F_i represents a trajectory segment $TS_i(1 \leq i \leq n)$.The dimensionality expansion requires the calculation of mean, minimum, maximum, 25%, 50% and 75% quartiles, and standard deviation[11] for each trajectory segment feature, and the obtained calculation result is the shallow feature sequence of each trajectory segment.According to Definition 1-1, the velocity change f_v, distance change f_d, angle change f_a and acceleration change f_{acc} of the trajectory analyzed in this section, the shallow feature sequence F is expressed as $F = \{f_v, f_d, f_a, f_{acc}\} \times \{$ mean, minimum, maximum, 25% quartiles, 50% quartiles, 75% quartiles, standard deviation $\}$, It is obvious that the shallow feature sequence F is a $28(4 \times 7)$ dimensional feature sequence. The shallow feature sequence can visually reflect the trajectory fragments and quantify the original GPS trajectory data at the same time. The set of shallow feature sequences Q is used to represent the trajectory fragment set TS, which is convenient for subsequent analysis and processing.

B. Deep Feature Fusion

After obtaining the shallow feature sequence, the depth feature fusion model based on the self-encoder is used to convert the shallow feature sequence into the deep feature sequence, and the shallow feature sequence is stitched with the deep feature sequence into the fused feature sequence to comprehensively represent the features of the trajectory segments from both deep and shallow aspects.

1) Self-Encoder: Fig.2 shows a three-layer self-encoder model, where L_1 is the input layer, L_2 is the hidden layer, and L_3 is the output layer. The input layer L_1 and the hidden layer L_2 constitute the encoder, which is responsible for extracting potential features. The hidden layer L_2 and the output layer L_3 form the decoder, which is responsible for reconstructing the data input from the hidden layer. Denote the data in input layer L_1, hidden layer L_2 and output layer L_3 by x, h and y $(x, y \in [0,1]^n, \quad h \in [0,1]^m)$. W is the weight matrix$(W \in R^{m \times n})$, b_1 $(b_1 \in R^m)$ denotes the bias of the input layer L_1 and b_2 $(b_2 \in R^n)$ denotes the bias of the hidden layer L_2.

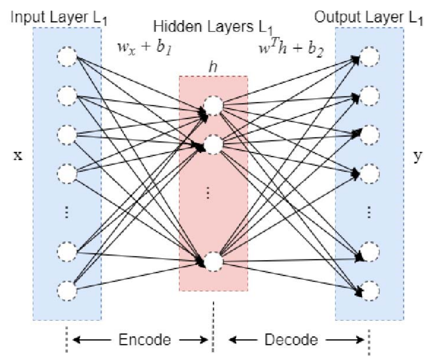

Fig. 2. Self-Encoder.

The value h in the hidden layer L_2 and the value y in the output layer L_3 are calculated by equation (1) and equation (2), where $\sigma()$ is the sigmod activation function.

$$h = f(x) = \sigma(Wx + b_1) \tag{1}$$

$$y = g(h) = \sigma\left(W^T h + b_2\right) \tag{2}$$

The purpose of using the self-encoder is to extract the salient high semantic features of the original data from the hidden layer, while having the function of dimensionality and data noise reduction. In order to make the learned hidden layer meaningful, the data output from the encoder needs to enter the decoder again for decoding, and through repeated iterations, the parameters of input and output are updated so that the values of y and x are close to each other, i.e., the self-encoder tries to learn the target optimization function of $y = F_{w.b}(x) \approx x$. The target optimization function can be expressed by the reconstruction error function $L(x, y)$, as shown in equation (3). According to the reconstruction error function, the loss functionloss can be obtained, and the $loss$ function is calculated by equation (4).

$$L(x, y)=-\sum_{i=1}^{n}\left(x_i \log \left(y_i\right)+\left(1-x_i\right) \log \left(1-y_i\right)\right) \quad (3)$$

$$\operatorname{loss}=\sum_{x \in Q} L(x, g(f(x))) \quad (4)$$

The weight matrix W and the biases b_1 and b_2 are updated as shown in equation (5), equation (6) and equation (7). Where,α denotes the learning rate.

$$W^{\text{new}}=W^{\text{old}}-\alpha \frac{\partial loss}{\partial W^{\text{old}}} \quad (5)$$

$$b_1^{new}=b_1^{\text{old}}-\alpha \frac{\partial loss}{\partial b_1^{\text{old}}} \quad (6)$$

$$b_2^{\text{new}}=b_2^{\text{old}}-\alpha \frac{\partial \text{loss}}{\partial b_2^{\text{old}}} \quad (7)$$

2) Fusion Feature Sequence Conversion: Using the deep feature fusion model based on self-encoder, the shallow feature sequences and deep feature sequences are transformed into fused feature sequences, which takes into account the detailed information of the shallow features and combines the semantic information of the deep features, and can highlight both the shallow and deep features of the trajectory segments, and the model structure is shown in Fig.3.

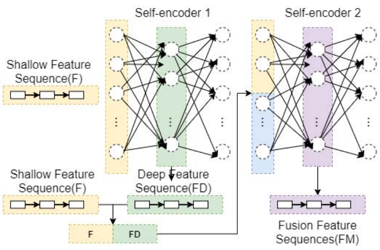

Fig. 3. Deep Feature Fusion Model.

The dimension of the shallow feature sequence at the input is 28. Considering the calculation time consumption and the size of the reconstruction error, the dimensions of the hidden layer L_2 of the self-encoder 1 and the self-encoder 2 are set to 10 and 15 respectively. When the 28-dimensional F is input to the model, the 10-dimensional FD is obtained through auto-encoder 1, and then spliced with F and sent to auto-encoder 2, and finally 15-dimensional FM is obtained.

C. Unsupervised Learning Based Feature Sequence Clustering

DBSCAN(Density-Based Spatial Clustering of Applications with Noise) can find clusters of arbitrary shapes and irregular clusters without prior knowledge of the number of clusters, but DBSCAN is too sensitive to parameters, and the clustering effect may be completely different under different parameters, which poses a great challenge to the parameter adjustment in experiments.OPTIC(Ordering Points to Identify the Clustering Structure) is an improved algorithm of DBSCAN, which can effectively solve the parameter sensitivity problem.

1) Similarity Index Anomaly Detection: Definition 1-2. Similarity index: The similarity matrix S is obtained by fusing the cosine similarity between each element of the set of feature sequences, and the set consisting of the similarity indices of all trajectory segments is the similarity indices A(The sum of the i-th row of S $ano_i=\sum s_i(1 \leq i \leq n)$)

Definition 1-3. anomalous trajectory fragment: given a similarity index threshold ρ, for a trajectory fragment TS, if its similarity index is less than the threshold ρ, then the trajectory is said to be fragment TS is an anomalous trajectory fragment.

Definition 1-4. Abnormal trajectory: Given an abnormal length ratio threshold τ, a trajectory T is said to be abnormal if the length ratio of its abnormal trajectory segment TS is greater than the threshold τ.

After clustering, the cluster set Cl containing each cluster can be obtained. firstly, the cosine similarity matrix of each cluster is calculated; secondly, the similarity index set is calculated according to definitions 1-2; after that, according to definitions 1-3, the anomalous trajectory fragment can be obtained; finally, the anomalous trajectory is finally determined by definitions 1-4.

III. Experiment and analysis

A. Experimental Data Set

The trajectory data set used in this paper is the trajectory data of one month (2018-12-19 00:00:00 to 2019-01-18 23:59:59) of the internal transport vehicle operations obtained from a port in Chongqing. Using the vehicle-mounted mobile terminal, data were collected with a sampling frequency of 8s/time, and a total of 1,239 operation instructions, 7,547 trips, and 3,415,835 trajectory points were collected.

B. Results and analysis

The parameters involved in the trajectory anomaly detection algorithm include open angle threshold ω, learning rate, nearest neighbor threshold ε, minimum number threshold MinPts, anomaly index threshold ρ and anomaly trajectory segment length ratio threshold τ. The settings of these parameters directly affect the results of trajectory anomaly detection.

The setting of the learning rate directly affects the optimization effect of the loss function in the self-encoder. The six curves from loss1 to loss6 are shown in Fig.4, which correspond to the loss functions with learning rates of 0.05, 0.1, 0.3, 0.5, 0.7 and 0.9, respectively.

The settings of the nearest neighbor threshold ε and the minimum number of MinPts have a direct impact on the clustering. However, the OPTIC algorithm is not sensitive to the parameters, and the reference value of the nearest neighbor threshold ε is given in the clustering, which greatly reduces the difficulty of parameter adjustment. After engineering experience and several experiments, the minimum number threshold MinPts is in the range of 15-25, and the nearest neighbor threshold ε is in the range of 0.5-0.6, which is more suitable.

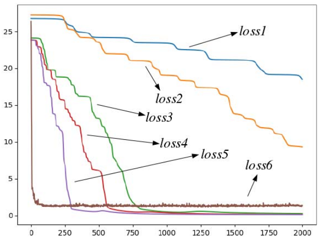

Fig. 4. Loss Function of Learning Rate.

Since the workload of manually calculating the result set is very large, the experiments in this section randomly select the trajectory data under 50 (45 ship unloading instructions and 5 transtacking instructions), 75 (67 ship unloading instructions and 8 transtacking instructions), 100 (90 ship unloading instructions and 10 transtacking instructions), and 125 (112 ship unloading instructions and 13 transtacking instructions) operation instructions in the data set, and after the extraction of trajectory real state feature sequences, the results are compared with the manually acquired result set, and the results are obtained as shown in Fig.5.

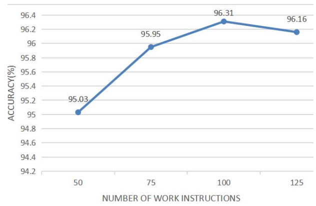

Fig. 5. Accuracy of Algorithms for Data of Different Sizes.

Overall, with the increase of the number of job instructions, the accuracy rate of S-DFFM is roughly at about 96% and the false alarm rate is only about 4%. The algorithms KNN(K-Nearest Neighbor), CNN(Convolutional Neural Networks), SVM(Support Vecor Machine), TSAD-FE(Trajectory Structure Anomaly Detection Based Feature Entropy), TAD-FD(Deep Characteristic) and the S-DFFM proposed in this paper are also selected, and the accuracy rate is averaged by several experiments to get as shown in Fig.6. Through the experimental results, it can be seen that the anomaly detection method based on unsupervised feature fusion proposed in this paper makes the similarity measure between trajectories more accurate and can prove the effectiveness of the algorithm and feasibility of the algorithm.

IV. Conclusion

In order to effectively supervise the cargo transfer process of port operation vehicles, this paper proposes a self-encoder-based feature fusion model (S-DFFM). The existing anomaly detection algorithm does not comprehensively consider the common features of the trajectory and is limited by the spatial

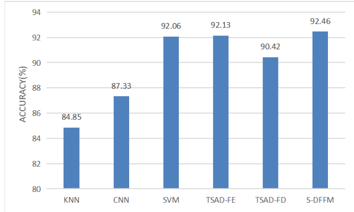

Fig. 6. Comparison of Accuracy Rates of Different Algorithms.

attributes of the trajectory, and this paper will fuse shallow feature sequences with deep feature sequences to represent the fused features of the trajectory, and the trajectory is represented by the feature sequences, which dilutes the spatial attribute of the trajectory, and the limitation of using spatial distance to judge the abnormality is solved. Based on the real operation trajectory data of a port in Chongqing, S-DFFM extracts the fusion features of the trajectory by using the fusion feature model and experiments the data set by using the unsupervised anomaly detection algorithm OPTIC. The experimental results show that the trajectory state extraction method of the fusion feature model in this paper can accurately extract the real state of the trajectory of the operating vehicle and can extract the stationary state of the vehicle during the operation. The experimental results show that the accuracy of S-DFFM in detecting anomalies in the trajectory of operating vehicles reaches 96%, which verifies that the trajectory anomaly detection algorithm proposed in this paper is suitable for detecting anomalies in the trajectory of operating vehicles in this port.

References

[1] Zheng Y. Trajectory data mining: an overview[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2015, 6(3): 29-71.

[2] Knorr E M, NgRT, Tucakov V. Distance-based outliers: algorithms and applications[J]. The VLDB Journal, 2000, 8(3): 237-253.

[3] Lee JG , Han JW, Li XL. Trajectory outlier detection: A partition-and-detect framework[C]. In Proceedings of the 24th IEEE Computer Society International Conference on Data Engineering, 2008: 140-149.

[4] Zhu J, Jiang W, Liu A, et al. Time-dependent popular routes based trajectory outlier detection[C]. In Proceedings of the 2015 International Conference on Web Information Systems Engineering, 2015: 16-30.

[5] Laxhammar R, Falkman G. Online learning and sequential anomaly detection in trajectories[J]. IEEE transactions on pattern analysis and machine intelligence, 2018, 36(6): 1158-1173.

[6] Laxhammar R, Falkman G. Sequential conformal anomaly detection in trajectories based on hausdorff distance[C].In Proceedings of the 14th International Conference on Information Fusion. IEEE, 2017: 1-8.

[7] Amodei D, Ananthanarayanan S, Anubhai R, et al. Deep speech 2: End-to-end speech recognition in english and mandarin[C]. In Proceedings of the 33rd International Conference on Machine Learning. 2016: 173-182.

[8] Ulyanov D, Vedaldi A, Lempitsky V. Deep image prior[C]. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition. 2018: 9446-9454.

[9] AI-Saffar A A M, Tao H, Talab M A. Review of deep convolution neural network in image classification[C]. In Proceedings of the 2017 IEEE International Conference on Radar, Antenna, Microwave, Electronics and Telecommunications. 2018: 26-31.

[10] Dong W , Li J , Yao R , Li C, Yuan T, Wang L. Characterizing driving styles with deep learning[J]. arXiv preprint, 2016: 1607.3611.

[11] Chen C, Liao C, Xie X, Wang Y, Zhao J. Trip2Vec: a deep embedding approach for clustering and profiling taxi trip purposes[J]. Personal and Ubiquitous Computing, 2019, 23(1): 53-66.