2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Cluster Naturalistic Driving Encounters Using Deep Unsupervised Learning

Sisi Li, Wenshuo Wang, Zhaobin Mo, and Ding Zhao

Abstract— **Learning knowledge from driving encounters could help self-driving cars make appropriate decisions when driving in complex settings with nearby vehicles engaged. This paper develops an unsupervised classifier to group naturalistic driving encounters into distinguishable clusters by combining an auto-encoder with k-means clustering (AE-kMC). The effectiveness of AE-kMC was validated using the data of 10,000 naturalistic driving encounters which were collected by the University of Michigan, Ann Arbor in the past five years. We compare our developed method with the k-means clustering methods and experimental results demonstrate that the AE-kMC method outperforms the original k-means clustering method.**

Index Terms— **Driving encounter classification, unsupervised learning, auto-encoder**

I. Introduction

Driving encounter in this paper is referred to as the scenario where two or multiple vehicles are spatially close to and interact with each other when driving. Autonomous vehicle has been becoming a hot topic in both industry and/or academia over past years and making human-like decisions with other encountering human drivers in complex traffic scenarios brings big challenges for autonomous vehicles. Currently, one of the most popular approaches to deal with the complex driving encounters is to manually and empirically classify it into several simple driving scenarios according to their specific application. For example, lane change behavior for autonomous driving applications were empirically decomposed into different categories [1], [2] based on human driver's active-passive decision behavior [3]. Then machine learning techniques, such as Markov decision process (MDP) and partially observable MDP (POMDP) [4], [5], were introduced to learn specific decision-making models. A well-trained model usually requires sufficient amounts of naturalistic driving data [6]. The authors in [7] combined reinforcement learning with game theory to develop a decision-making controller for autonomous vehicles using the vehicle interactions data generated from a traffic simulator. However, the trained controller in [7] may become invalid when deployed in real world traffic environment. In addition, directly feeding the training data into model without any classification could not make full use of underlying data resource and thereby could miss some rare but important driving scenarios in real traffic settings. One of the most significant challenges in solving this issue is to obtain massive high-quality labeled and categorized traffic data from naturalistic traffic settings, which is essential for the robustness and accuracy of learning decision-making models.

Modern sensing technologies such as cameras, Lidar, and radar give great advantages to gather large-scale traffic data with multiple vehicles encountering; for instance, the University of Michigan Transportation Research Institute (UMTRI) equipped multiple buses with GPS tracker in Ann Arbor to gather the vehicle interactions data for autonomous vehicle research and more released database listed in [8], [9]. The gathered vehicles' interaction data could be explored to develop autonomous vehicle controller and generate testing motions to evaluate self-driving algorithms [10]–[12]. However, how to label such a massive dataset efficiently and group driving encounters in a reasonable way still remain as challenges. Labeling a huge amount of data manually is a time-consuming task, which requires the data analysts with rich prior knowledge covering the fields of traffic, intelligent vehicles as well as human factors. Ackerman claimed that for one-hour well-labeled training data, it approximately takes 800 human hours. Ohn-Bar [13] took excessive time to manually annotate the objects in recorded driving videos to investigate the importance rank of interesting objects when driving. In order to meet the data-hungry learning-based methodologies, the industry and academia both need a tool that could automate the labeling process, thereby effectively eliminating labeling costs [14]. Except for classifying driving encounters, research has been conducted in recent years to automate the data labeling process, for example, ranging from labeling the ambiguous tweets automatically using supervised learning methods [15] to classifying driving styles using unsupervised learning methods (e.g., k-means) [16] and semi-supervised learning methods (e.g., semi-supervised support vector machine) [17]. Although these aforementioned unsupervised and semi-supervised approaches could reduce labeling efforts, but they are not suitable to deal with huge amounts of high-dimensional time-series data. Toward this end, the deep learning approach is becoming popular to gain insight into such data. For example, the auto-encoder and its extensions have been used to analyze driving styles [18] and driver behavior [19], which empirically demonstrates its effectiveness. For autonomous vehicles, learning models from these manually predefined categories could

S. Li is with the Robotics Institute, University of Michigan, Ann Arbor, MI, 48109

W. Wang is with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, 48109 wenshuow@umich.edu

Z. Mo is with the Automotive Engineering at the Tsinghua University, Beijing, China, 100084

D. Zhao is with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, 48109 zhaoding@umich.edu

This work was funded by the Toyota Research Institute with grant No. N021936.

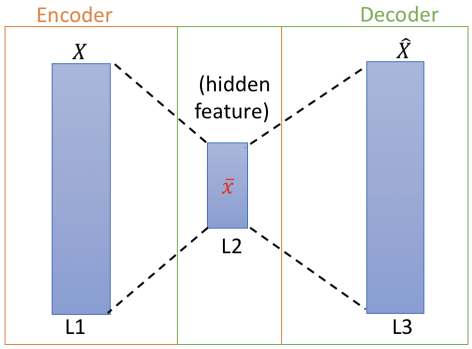


Fig. 1: Structure of a typical auto-encoder.

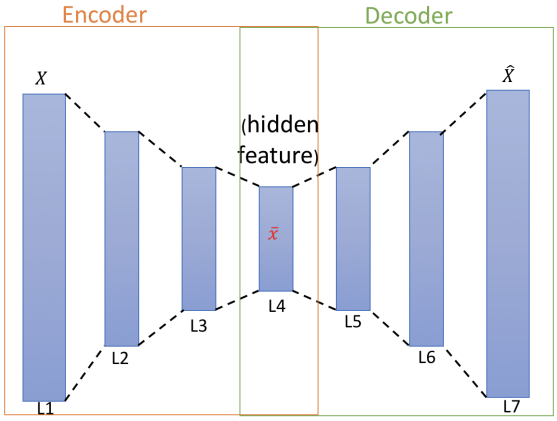


Fig. 2: Structure of the auto-encoder used in this paper.

suffer the following limitations:

1) Manually labeling huge amounts of high-dimensional data requires excessive cost of time and resources and could generate large biases due to the diverse prior knowledge of data analysts.
2) The most suitable classified categories of driving encounters for human understanding may not generate the most suitable results to learn a decision-maker for self-driving cars.

This paper presents an unsupervised framework, i.e., combining an autoencoder with k-means clustering (AE-kMC), to automatically cluster driving encounters with less subjective interference. The auto-encoder is employed as a component of extracting hidden features in a driving encounter classifier. The data were collected by the M-City with more than 2,800 cars, including commercial trucks and transit vehicles, throughout five years. The original contribution of this work is that an autoencoder-based framework was developed and implemented to automatically label the driving encounters according to the identified features, which has not been previously proposed elsewhere to the best of our knowledge. Finally, we make a comprehensive analysis for experiment results. The source code can be found at `https://github.com/zhao-lab/li-18-deep-unsupervised-ivs`.

The rest of the paper is organized as follows. The developed AE-kMC approach is detailed in Section II. The vehicle encounter data collection and model training procedure are shown in Section III. Section IV shows the experiment results. Section V presents discussion and conclusion.

II. Unsupervised Learning Methods

In this section, we will present two different unsupervised schemes to automatically cluster vehicle encounters using k-means clustering and AE-kMC. In what follows, we will introduce the theoretical basis of the traditional auto-encoder, the architecture of auto-encoder neural networks, and the k-means clustering method.

A. Auto-Encoder Framework

The auto-encoder implemented in this work is a typical auto-encoder that contains a encoder and a decoder, as shown in Fig. 1.

1) Encoder: The encoder, labeled by an orange box in Fig. 1, maps an input data, $\mathbf{X} \in \mathbb{R}^n$, into a hidden representation $\mathbf{x} \in \mathbb{R}^m$ [20]:

$$f_\theta(\mathbf{X}) = \mathbf{W}\mathbf{X} + \mathbf{b}, \tag{1a}$$

where $\mathbf{W} \in \mathbb{R}^{m \times n}$ is a weight matrix and $\mathbf{b} \in \mathbb{R}^m$ is a bias term. Unlike the normal encoder that only has one input layer and one output layer, the encoder we used is a four-layer neural network that contains one input layer (L1), two hidden layers (L2 and L3) and one output layer (L4), as shown in Fig. 2. Two hidden layers are added because each of those two extra layers represents an approximation of any function according to the universal approximator theorem [21]. The modified encoder benefits for the new depth since the training cost, both computational cost and training data size, can be reduced [21]. The representation of the data at each layer is calculated as:

$$\mathbf{X_{i+1}} = \mathbf{W_{i,i+1}}\mathbf{X_i} + \mathbf{b_{i,i+1}}, \tag{2}$$

where $\mathbf{X_i}$ is the input data representation at layer L_i, and $\mathbf{W_{i,i+1}}$ and $\mathbf{b_{i,i+1}}$ are the weight vector and the bias vector, respectively, capable of mapping $\mathbf{X_i}$ to $\mathbf{X_{i+1}}$. Note that the data representation at the layer 4, $\bar{\mathbf{x}}$, is the code or the hidden feature of the input data.

2) Decoder: The coded data, or the hidden representation $\bar{\mathbf{x}} \in \mathbb{R}^n$ from the encoder will be mapped back to $\hat{X}$ through layers L5 and L7, which is the reconstructed representation of the original input data $\mathbf{X}$. Note that the decoder employed in this work is a deep neural network that contains one input layer (L4), two hidden layers (L5 and L6) and one output layer.

B. Optimization

In order to reconstruct the input data from the hidden feature, $\bar{\mathbf{x}}$, the error between the reconstructed representation and the original input should be minimized. Hence the cost function can be defined as

$$E(\mathbf{\Omega}) = (\mathbf{X} - \hat{\mathbf{X}})^T(\mathbf{X} - \hat{\mathbf{X}}), \tag{3}$$

where $\mathbf{\Omega}$ is the parameter set that contains the weight vector and the bias vector of each layer, $\hat{\mathbf{X}} = g_\Omega(\bar{\mathbf{x}})$, g_Ω is the mapping that map the hidden representation $\mathbf{x}$ to $\hat{\mathbf{X}}$. Then, the optimization problem is defined as

$$\Omega^* = \arg\min_{\Omega} E(\Omega). \tag{4}$$

C. k-Means Clustering

The k-means clustering is one of popular unsupervised machine learning techniques for classification. Applying the trained auto-encoder on the raw vehicle encounter data, the hidden feature can be extracted and then fed into the k-means clustering. Given n observations $(x_1, x_2, \ldots, x_n)$ and define k classes, the k-means clustering method clusters the observations into k groups by solving the optimization problem

$$\mathbf{v}^* = \underset{\mathbf{v}}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{j=1}^{n} |x_i - v_j|^2, \tag{5}$$

where $|x_i - v_j|$ is the Eucledian distance between the centroid and the observation [22]. The objective function tries to pick centroids that minimize the distances to all points belonging to its respective cluster so that the centroids are more representative of the surrounding cluster of data points.

III. Data Collection and Experiment

A. Data Collection

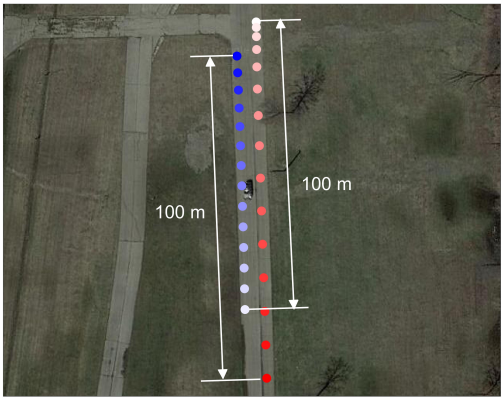


Fig. 3: Example of driving encounters. Dark dot is start point and light dot is end point.

We used the Safety Pilot Model Deployment (SPMD) database, which provides sufficient naturalistic data [6]. This database was conducted by the University of Michigan Transportation Research Institute (UMTRI) and provides driving data logged in the last five years in Ann Arbor area [23]. It includes approximately 3,500 equipped vehicles and 6 million trips in total. Latitude and altitude information for clustering was collected by the on-board GPS. The collection process starts at the ignition for each equipped vehicle. The data was collected with a sampling frequency of 10 Hz.

We used the dataset of 100,000 trips, collected from 1900 vehicles with 12-day runs. The trajectory information we extracted includes latitude, longitude, speed and heading angle of the vehicles. The selection range was constricted to an urban area with the range of latitude and longitude to be (-83.82,-83.64) and (42.22,42.34), respectively. The vehicle encounter was defined as the scenario where the vehicle distance was small than 100 m, as shown in Fig. 3. The dots indicate the position of the vehicle at every sample time. After querying from the SPMD database, we got 49,998 vehicle encounters.

The distribution of these encounters is shown in Fig. 4. The central points of the minimal rectangular that can encompass the trajectory is used to include the massive trajectories in one image. Note that to reduce the computational load, 10,000 encounters were randomly selected to test the unsupervised clustering methods.

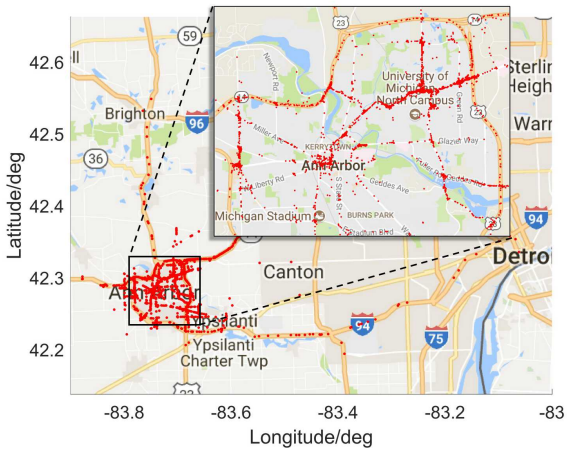


Fig. 4: Distribution of driving encounters.

B. Model Training Procedure

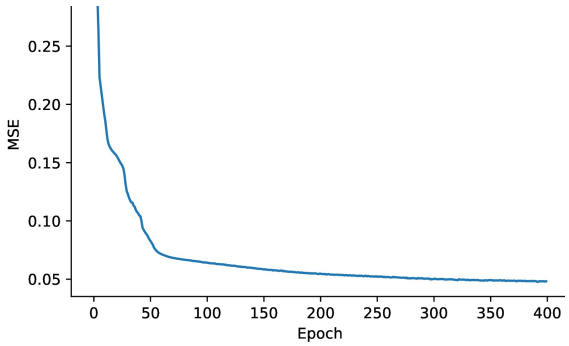
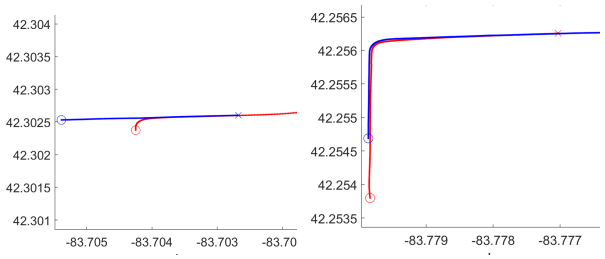


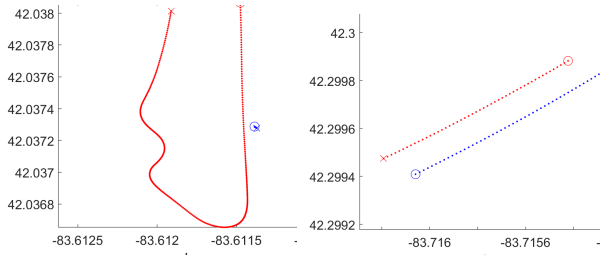
Fig. 5: Cost during the training of the model.

The GPS data of an encounter was used as the input of the auto-encoder. Before doing this, the data was normalized to put all the training data in the same scale. The weights for each layers were initialized using random numbers between 1 and -1 following uniform distribution. More specifically, we set $\mathbf{W_{1,2}} \in \mathbb{R}^{100\times200}$, $\mathbf{W_{2,3}} \in \mathbb{R}^{50\times100}$, $\mathbf{W_{3,4}} \in \mathbb{R}^{25\times50}$,



(a) Two vehicles merge

(b) One vehicle follows another vehicle



(c) One vehicle bypasses another vehicle

(d) Two vehicles encounter in the same road from opposite directions

Fig. 6: Original vehicle interaction trajectories. Horizontal and vertical axises are the longitude and latitude of vehicles, respectively.

$\mathbf{W_{4,5}} \in \mathbb{R}^{25\times50}$, $\mathbf{W_{5,6}} \in \mathbb{R}^{50\times100}$, $\mathbf{W_{6,7}} \in \mathbb{R}^{100\times200}$. The classic stochastic gradient decent method was employed to learn Ω. The cost function value during the training process is shown in Fig. 5. The training error converges to 0.041, which shows the trained Auto-encoder could reconstruct the input data from the hidden feature. Fig. 6 shows an example of the raw input of auto-encoders, i.e, raw vehicle GPS trajectory.

IV. Result Analysis and Discussion

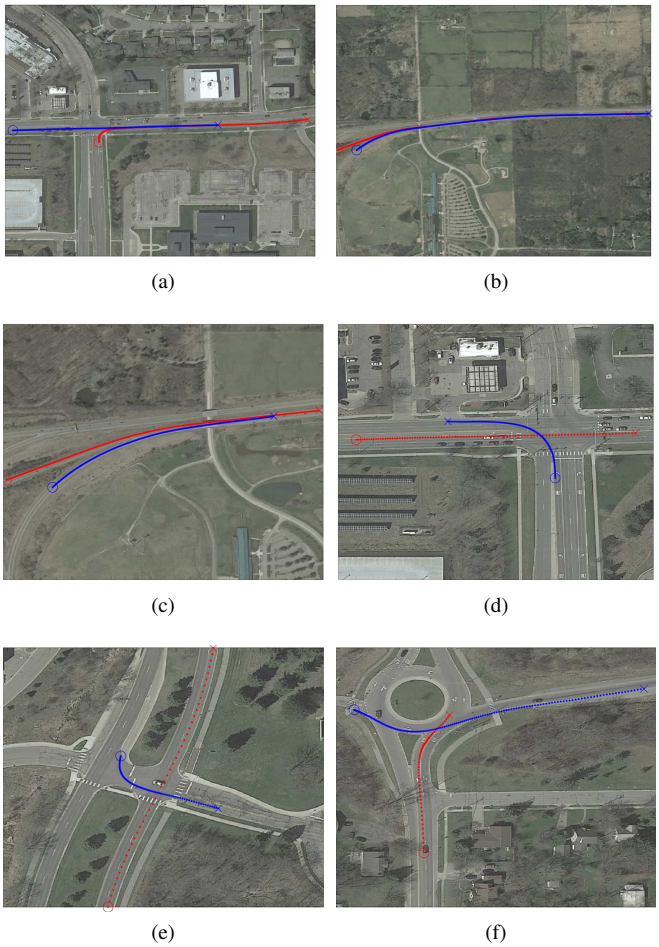
In this section, we will present and compare the cluster results obtained from the two unsupervised learning methods. In real traffic environment, the typical driving encounters mainly consist of four cases:

1) Category A: Two vehicles encounter with each other in an intersection.
2) Category B: Two vehicles encounter in the opposite direction of a road.
3) Category C: One vehicle bypasses another vehicle;
4) Category D: Two vehicles interact in a same road (with and without lane changing).

Fig. 7 shows a sample of cluster results that contains the interaction behavior in an intersection. The developed AE-kMC successful clusters the vehicle interactions that fall into the above four categories.

The scenario where two vehicles encounter on the same road from opposite directions are shown in Fig. 8. Fig. 9 demonstrates two vehicles interact on the same road. Note that the lane changing actions are contained in this cluster but are difficult to be identified from the results.

The traffic scenario where one vehicle passes another vehicle is shown in Fig. 10. Note that in this case, one vehicle



(a) (b) (c) (d) (e) (f)

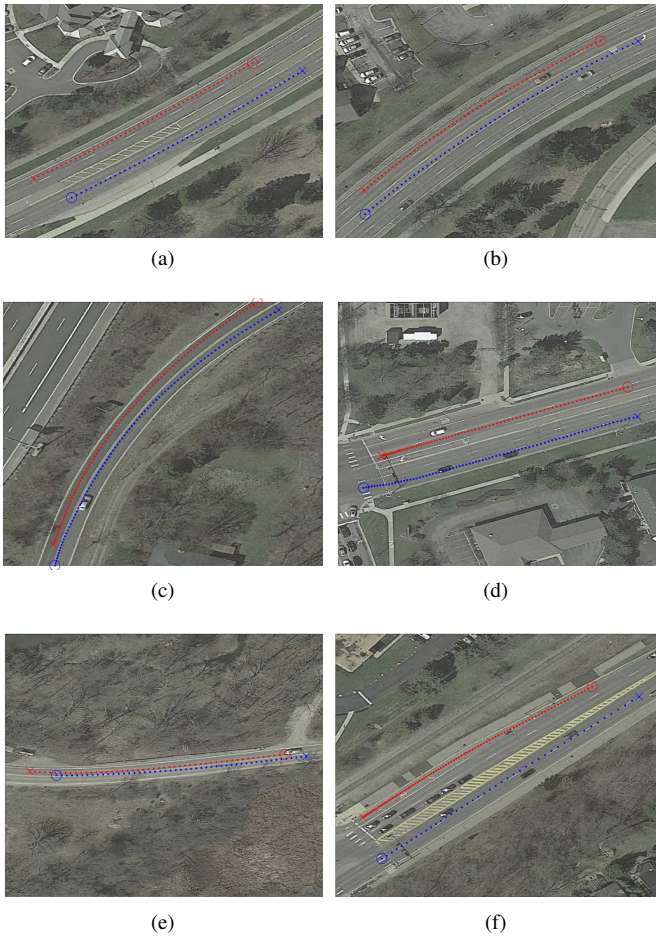
Fig. 7: Cluster of two vehicles encountering at intersections. Circle represents the start positions and cross represents the end position.

is stationary (Fig. 10a, Fig. 10b, Fig. 10c and Fig. 10d) or moving but located at far distance from the other vehicle (Fig. 10e and Fig. 10f).

To evaluate the accuracy, we randomly shuffled and re-sampled 100 samples for each cluster obtained from the proposed method, then we manually identified the ones that do not fit into the patterns that most of the samples in the cluster shown. Thus the performance of each cluster is calculated by

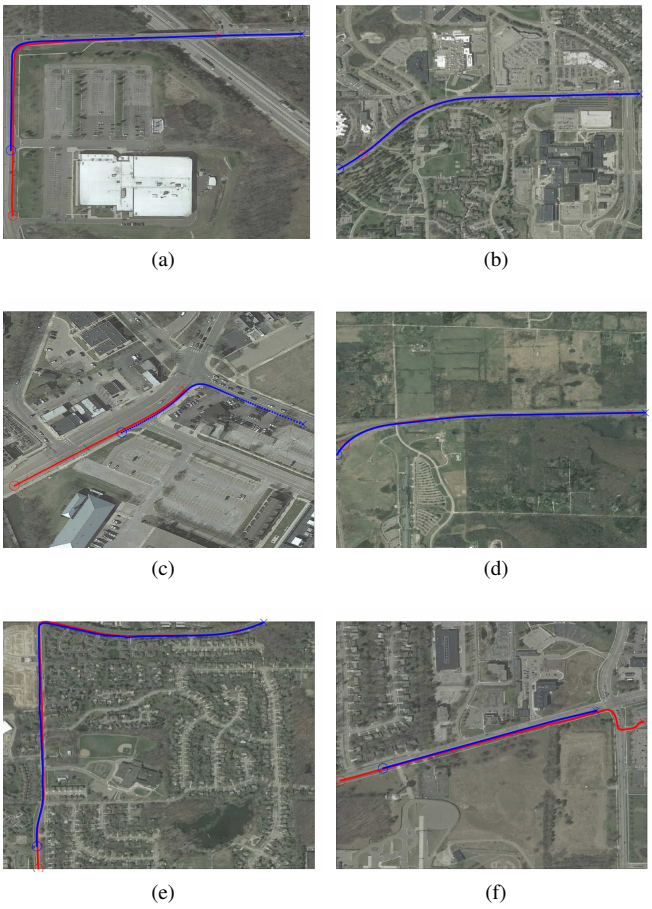
$$\eta = 1 - \frac{n_{abnormal}}{N_{sample}}, \tag{6}$$

where $n_{abnormal}$ is the number of the abnormal data and $N_{sample} = 100$. The method we proposed can cluster vehicle encounters that have similar geographic features but each cluster still contains abnormal driving encounters. Table I shows the performance of the clusters. The method of only utilizing k-means clustering is also evaluated and the results indicate that it can also cluster the above vehicle interaction categories with a lower performance compared with the method uses AE-kMC.



(a) (b) (c) (d) (e) (f)

Fig. 8: Clusters of two vehicle encounter on the same road with opposite directions.



(a) (b) (c) (d) (e) (f)

Fig. 9: Cluster of two vehicles encountering on the same road.

TABLE I: Clustering performance analysis

Cluster	AE-kMC	k-means
Category A	73%	41.6%
Category B	74.4%	37.9%
Category C	68.2%	47.6%
Category D	83.8%	78.4%

V. Conclusion and Future Work

In this paper, we proposed an unsupervised learning method to cluster vehicle encounter data. More specifically, the auto-encoder was introduced to extract the hidden feature of driving encounters and the extracted features were then grouped using the k-means clustering method. The proposed AE-kMC method finally obtains five main typical types of vehicle encounters, including 1) two vehicles intersect with each other; two vehicles merge; 3) two vehicles encounter in the opposite direction of a road; 4) one vehicle bypasses another vehicle; 5) two vehicles interact in a same road (with and without lane changing). We also compared the developed AE-kMC method with the k-means clustering method and the result shows that our developed AE-kMC method outperforms the k-means clustering method.

In future work, we will focus on improving the accuracy of the unsupervised clustering approaches, including evaluating different types of the auto-encoders and clustering approaches. Moreover, target vehicle encounter extraction, for example, vehicle interactions in roundabout traffic environment, will be investigated.

Acknowledgment

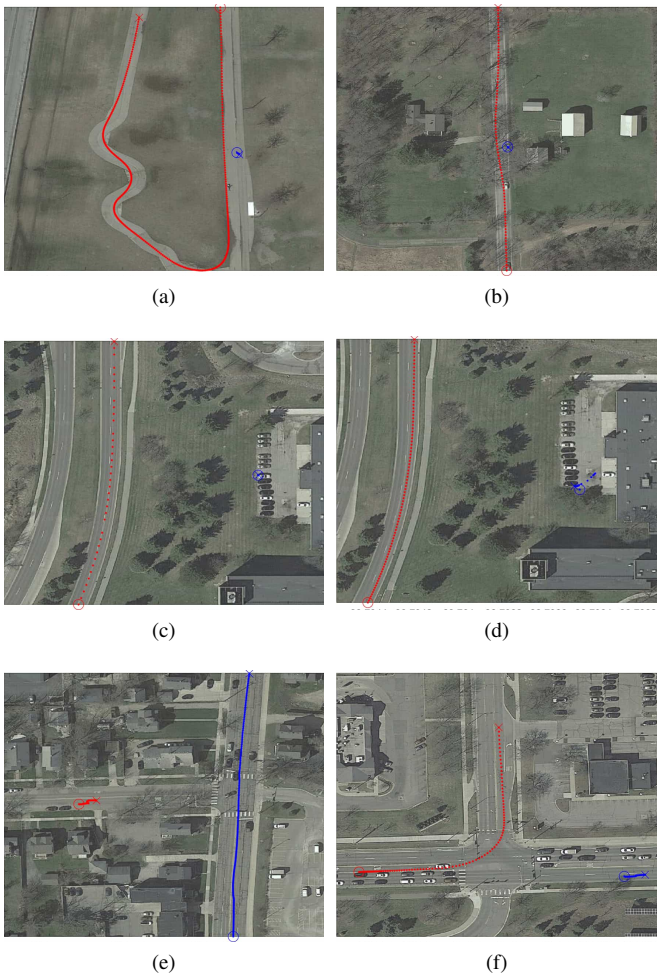
Toyota Research Institute ("TRI") provided funds to assist the authors with their research but this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

References

[1] J. Nilsson, J. Silvlin, M. Brannstrom, E. Coelingh, and J. Fredriksson, "If, when, and how to perform lane change maneuvers on highways," *IEEE Intelligent Transportation Systems Magazine*, vol. 8, no. 4, pp. 68–78, 2016.

[2] J. Nilsson, M. Brännström, E. Coelingh, and J. Fredriksson, "Lane change maneuvers for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1087–1096, 2017.

[3] Q. H. Do, H. Tehrani, S. Mita, M. Egawa, K. Muto, and K. Yoneda, "Human drivers based active-passive model for automated lane change," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 42–56, 2017.



(a) (b) (c) (d) (e) (f)

Fig. 10: Cluster of one vehicle passing another vehicle.

[4] C. Hubmann, M. Becker, D. Althoff, D. Lenz, and C. Stiller, "Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles," in *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 1671–1678.

[5] M. Kuderer, S. Gulati, and W. Burgard, "Learning driving styles for autonomous vehicles from demonstration," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2641–2646.

[6] W. Wang, C. Liu, and D. Zhao, "How much data are enough? a statistical approach with case study on longitudinal driving behavior," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 2, pp. 85–98, 2017.

[7] N. Li, D. W. Oyler, M. Zhang, Y. Yildiz, I. Kolmanovsky, and A. R. Girard, "Game theoretic modeling of driver and vehicle interactions for verification and validation of autonomous vehicle control systems," *IEEE Transactions on Control Systems Technology*, vol. PP, no. 99, pp. 1–16, 2017.

[8] W. Wang and D. Zhao, "Extracting traffic primitives directly from naturalistically logged data for self-driving applications," *IEEE Robotics and Automation Letters*, 2018, DOI: 10.1109/LRA.2018.2794604.

[9] D. Zhao, Y. Guo, and Y. J. Jia, "Trafficnet: An open naturalistic driving scenario library," *arXiv preprint arXiv:1708.01872*, 2017.

[10] D. Zhao, H. Lam, H. Peng, S. Bao, D. J. LeBlanc, K. Nobukawa, and C. S. Pan, "Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques," *IEEE transactions on intelligent transportation systems*, vol. 18, no. 3, pp. 595–607, 2017.

[11] D. Zhao, X. Huang, H. Peng, H. Lam, and D. J. LeBlanc, "Accelerated evaluation of automated vehicles in car-following maneuvers," *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[12] Z. Huang, H. Lam, D. J. LeBlanc, and D. Zhao, "Accelerated evaluation of automated vehicles using piecewise mixture models," *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[13] E. Ohn-Bar and M. M. Trivedi, "Are all objects equal? deep spatio-temporal importance prediction in driving videos," *Pattern Recognition*, vol. 64, pp. 425–436, 2017.

[14] T. Appenzeller, "The scientists' apprentice," *Science*, vol. 357, no. 6346, pp. 16–17, 2017.

[15] M. Erdmann, E. Ward, K. Ikeda, G. Hattori, C. Ono, and Y. Takishima, "Automatic labeling of training data for collecting tweets for ambiguous tv program titles," in *2013 International Conference on Social Computing*, Sept 2013, pp. 796–802.

[16] C. M. Martinez, M. Heucke, F.-Y. Wang, B. Gao, and D. Cao, "Driving style recognition for intelligent vehicle control and advanced driver assistance: A survey," *IEEE Transactions on Intelligent Transportation Systems*, 2017, DOI: 10.1109/TITS.2017.2706978.

[17] W. Wang, J. Xi, A. Chong, and L. Li, "Driving style classification using a semisupervised support vector machine," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 5, pp. 650–660, 2017.

[18] W. Dong, T. Yuan, K. Yang, C. Li, and S. Zhang, "Autoencoder regularized network for driving style representation learning," *arXiv preprint arXiv:1701.01272*, 2017.

[19] H. Liu, T. Taniguchi, Y. Tanaka, K. Takenaka, and T. Bando, "Visualization of driving behavior based on hidden feature extraction by using deep learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2477–2489, 2017.

[20] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010.

[21] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[22] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[23] X. Huang, D. Zhao, and H. Peng, "Empirical study of dsrc performance based on safety pilot model deployment data," *IEEE Transactions on Intelligent Transportation Systems*, 2017.