This is an Accepted Manuscript of an article published by Taylor & Francis in **CYBERNETICS AND SYSTEMS** on 20.01.2020, available online: https://www.tandfonline.com/doi/full/10.1080/01969722.2019.1705547

Postprint of: Zhang H., Li F., Wang J., Zhou Y., Sanin C., Szczerbicki E.: Experience-Based Cognition for Driving Behavioral Fingerprint Extraction CYBERNETICS AND SYSTEMS Vol. 51, iss. 2 (2020), pp.103-114

Experience-Based Cognition for Driving Behavioral Fingerprint Extraction

Haoxi Zhang [a], Fei Li [a], Juan Wang [a], Yang Zhou[a], Cesar Sanin [b], and Edward Szczerbicki [c]

[a] School of Cybersecurity, Chengdu University of Information Technology, Chengdu, China; [b] School of Mechanical Engineering, The University of Newcastle, Newcastle, NSW, Australia; [c] Faculty of Management and Economics, Gdansk University of Technology, Gdansk, Poland

Abstract. With the rapid progress of information technologies, cars have been made increasingly intelligent. This allows cars to act as cognitive agents, i.e. to acquire knowledge and understanding of the driving habits and behavioral characteristics of drivers (i.e. driving behavioral fingerprint) through experience. Such knowledge can be then reused to facilitate the interaction between a car and its driver, and to develop better and safer car controls. In this paper, we propose a novel approach to extract the driver's driving behavioral fingerprints based on our conceptual framework Experience-Oriented Intelligent Things (EOIT). EOIT is a learning system that has the potential to enable Internet of Cognitive Things (IoCT) where knowledge can be extracted from experience, stored, evolved, shared, and reused aiming for cognition and thus intelligent functionality of things. By catching driving data, this approach helps cars to collect the driver's pedal and steering operations and store them as experience; eventually, it uses obtained experience for the driver's driving behavioral fingerprint extraction. The initial experimental implementation is presented in the paper to demonstrate our idea, and the test results show that it outperforms the Deep Learning approaches (i.e. deep fully connected neural networks and recurrent neural networks/Long Short-Term Memory networks).

Keywords: Vehicle Intrusion Detection, Neural Knowledge DNA, Set of Experience Knowledge Structure, Knowledge Representation, Deep Learning.

Address correspondence to Haoxi Zhang, School of Cybersecurity, Chengdu University of Information Technology, No. 24 Block 1, Xuefu Road, Chengdu, China, 610225. E-mail: haoxi@cuit.edu.cn

Introduction

Recently, together with the rapid progress of information technologies and the advances in vehicle active control and advanced driver assistance systems, there have been significant research efforts in assisting the human driver in vehicle control actions. These advances allow artificial intelligence software to be embedded in cars and to learn the driving habits and behavioral characteristics of drivers (i.e. driving behavioral fingerprint) through their operational data of driving process. More importantly, knowledge of the driver's driving behavioral fingerprint can facilitate the interaction between automobiles and human drivers so that better and safer car controls can be made, for example: predicting a driver's intended actions early and accurately can help mitigate the effects of potentially unsafe driving behaviors and avoid possible accidents (Liu 2008; Olabiyi et al. 2017).

In this paper, we present an experience-based approach to extract driver's driving behavioral fingerprint based on our conceptual framework called Experience-Oriented Intelligent Things (EOIT) that enables Internet of Things (IoT) to extract knowledge from their past experiences. This knowledge can then be stored, shared, and reused enhancing smartness of car's functionality. By catching car driving data, this approach helps to collect experience of drivers' car control operations, to analyse this experience, and eventually to use the obtained experience for the driver's driving behavioral fingerprint extraction. We demonstrate that using the available dataset our proposed approach outperforms in speed and accuracy methodologies based on deep learning models (i.e. LSTM), even though LSTM based results are regarded as very good.

The rest of this paper is organized as follows: section two describes related works; section three introduces the overview, features, and architecture of the Experience-Oriented Intelligent Things; section four presents the experimental implementation of our approach. Finally, in section five, concluding remarks are drawn.

Related Work

The driver behavior modeling is a broad as well as challenging research topic. Largely it involves longitudinal (pedal) and lateral (steering) control modeling. In this modelling pedal positions and steering angles are the output of a sophisticated virtual system, which involves human sensing, inference, decision making, and body movement processes, making driver behavior modeling a very difficult task to accomplish.

In study (Zeng & Wang 2017), an input–output hidden Markov model is used to describe the pedal behavior. The state transition and output distribution functions are designed, and the relation between the input and the key variables of the output distributions is analyzed and modeled using statistical approaches. This model can incorporate the road information, capture individual driver's driving style, and predict the output action probability distribution. Similarly, some other approaches (Berndt et al. 2008; Kuge et al. 2000; Wang 2018) using the hidden Markov model and its modified structures are shown performing well in various real life driving applications. In study (Schnelle et al. 2016), a combined driver model that is demonstrated to be able to not only identify different individual driver behaviors, but also predict a driver's behavior in rare vehicle maneuvers such as collision avoidance (CA) based on his/her daily driving data. The proposed driver model can replicate each driver's steering wheel angle signal for a variety of highway and in-city maneuvers. The advantage of the proposed driver model is its ability to predict a driver's steering wheel angle signal for a CA maneuver just from daily nonemergency driving data. By employing the virtual driver to conduct the speed following task as defined by the standard driving cycle test, a personalized driver model is established in another study (Hu et al. 2017). For speed control purposes this model uses the locally designed neural network and the real-world vehicle test data. The proposed model is able to detect three typical abnormal driving behaviors that are pre-characterized and simulated, namely, the fatigue/drunk, the reckless, and the phone use while driving. In studies (Oliver & Pentland 2000; Liu 2008; Meyer-Delius et al. 2009), Bayesian methods were used to recognize driver intention with complex input feature dimension. Moreover, in studies (Hess et al. 1990; Macadam 2007; McRuer 1980; Donges 1978), driving models were built using optimal control theory for tracking the driving path, estimating the route, or analyzing steering behavior.

Because the driver behavioral modeling is generally a time-series anomaly prediction problem, hand engineered features might not return the best result. Therefore, deep learning (Lecun et al. 2015) based methods, more specifically, the Long

Short-Term Memory networks (LSTM) (Hochreiter & Schmidhuber 1997) models are becoming popular in solving this problem (Olabiyi et al. 2017; Jain et al. 2016; Saleh et al. 2018; Morton et al. 2017; Carvalho et al. 2017). LSTM is capable of connecting previous information to the present task, which allows it to exhibit temporal dynamic behavior for a time sequence and make it well-suited to processing and making predictions based on time series data (Sak et al. 2014).

In comparison, we propose a novel approach that explores the driver behavior modeling from a different perspective: experience-based cognition. We employ the Experience-Oriented Intelligent Things (introduced in Section 3) to collect and manage driving experience, and use the Triangle Mean Algorithm (introduced in Section 4) to reuse experience. Our model can learn both steering and pedal (i.e. braking and acceleration) actions unlike some of the works we mentioned above that focused only on either steering or pedal actions. More importantly, using experience not only enables much faster inference and prediction, which is fundamental for time critical tasks like driving, but also allows our model to evolve in time through adding new experience and up-to-date in real time with high accuracy, which cannot be done in deep learning based models without retraining.

Experience-Oriented Intelligent Things

The Experience-Oriented Intelligent Things (EOIT) framework is proposed to help IoT extract knowledge from their past experiences, as well as store, evolve, share, and reuse such knowledge aiming for cognition and intelligent functions. This section presents the overview, main features, and architecture of the EOIT.

A. Overview

Due to the diversity of IoT applications, the hardware and software implementations could be various. Moreover, different companies, applications, systems could use different devices to perform even exactly the same task. The EOIT concept is designed to handle these issues, providing the means for organizations and individuals to have a common systematic framework within diverse IoT.

Figure 1 shows the overview of the EOIT framework. As illustrated in Figure 1, the framework integrates IoT and different tasks at two major levels: local level, and global level.

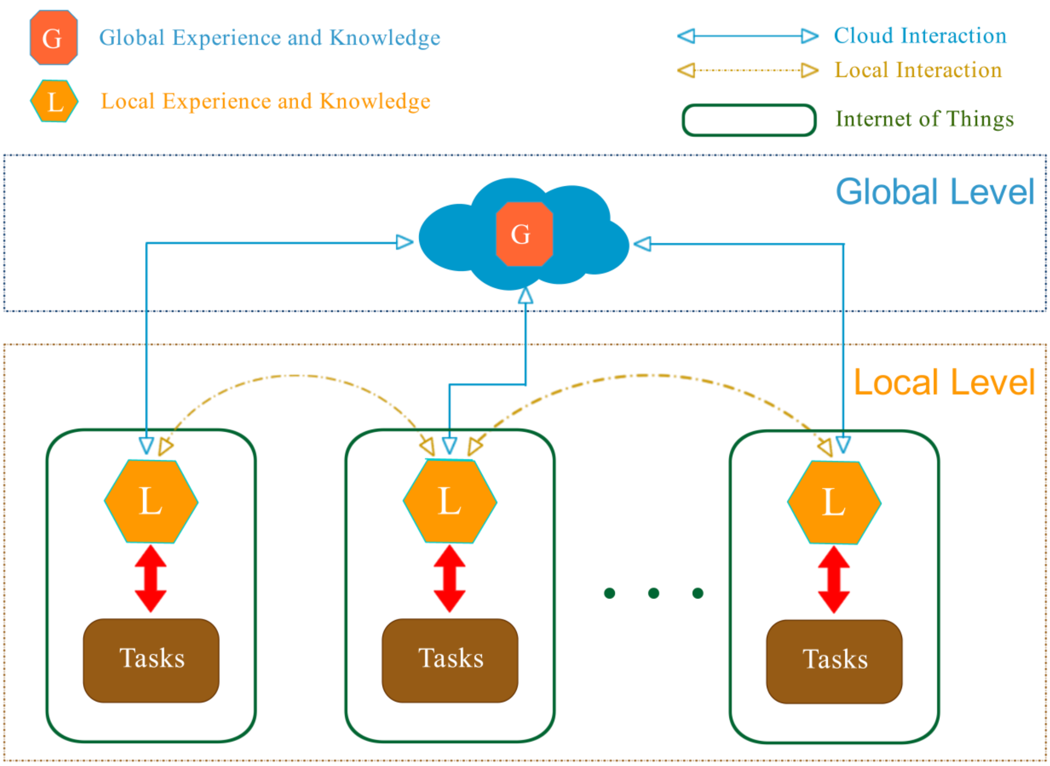

Fig. 1. Overview of the EOIT.

At the bottom of the framework, i.e. local level, there are different IoT hardware and devices computing and generating data for the tasks they are involved in. Each one of IoT at the local level may be assigned to a single or multiple task. In other words, to gain the knowledge of one thing (part of IoT), it may be needed to involve data from one or more tasks. By integrating data captured from tasks, experience of that given thing keeps forming and growing, so that experiential knowledge can be extracted and evolves via its learning engine introduced in (Zhang et al. 2017a; Zhang et al. 2017b). At the top, the global level, technologies Cloud and Fog Computing are applied in order to extract, store, evolve, and share knowledge among IoT at a global scale (Sanin et al. 2018).

B. EOIT Main Features

1) Experience-oriented

Experience, as one kind of knowledge gained from practice, is the ideal source for improving performance of processes, in which a number of practical activities are involved. By reusing experiential knowledge, decision makers can make decisions faster, and more efficiently due to supporting their current decisions on experiences obtained from previous similar situations. Therefore, capturing the experience instead of all available data produced by IoT is the EOIT preference.

2) Adaptable

Currently, most IoT applications are customized; the hardware and software for each of them could be distinctly different. In order to work with customized devices and software, knowledge-related tools are usually specifically designed only for given IoT or even for a particular stage of a product lifecycle. However, adaptability is rarely considered during the design and development of these tools. For solving this problem, EOIT is designed to work with specific designed devices and applications without re-designing or re-building them.

3) Compact and Efficient

Another important element concerning the application of EOIT is that many IoTs are resource-limited, which means our applications have to be designed in a compact but efficient form. As it has been shown in previous case studies (Zhang et al. 2010) using experience as the basic knowledge source is an efficient knowledge representation approach, which can indeed enhance knowledge retrieval and experience reusing applications.

4) Configurable

It is common for IoT applications to run under different scenarios. For example, the same application may work under different power supply conditions: connected to a power line or by using batteries. If the application performs always in the same way regardless of its power supply, it will soon run out of power and shut down when using batteries. Also, an application may work in different modes. For example it can be trained in a training mode, and work in an automatic self-control mode. For developers, it may have debugging mode. Therefore, configurability is a very desirable feature as it allows applications to perform properly in various scenarios according to different settings.

5) Secure and Trustable

It is clear that a secure environment is a key requirement for any distributed system these days, especially when the Internet is used as the primary communication channel. Also, knowledge and knowledge sources must be reliable to make right decisions. Therefore, the concept of decisional trust presented by Sanin and Szczerbicki (2006) is adopted for EOIT and extended to include more features that reflect human-like behaviour.

C. Architecture

For carrying these five main features introduced above, the conceptual four-layer architecture is designed for EOIT. These four layers are: Application Layer, Interface Layer, Processing Layer, and Knowledge Repository Layer. The platform is conceptualized on the top of IoT, to extend the intelligence-related capabilities of the latter by using experience. Figure 2 illustrates the conceptual architecture proposed for the EOIT.

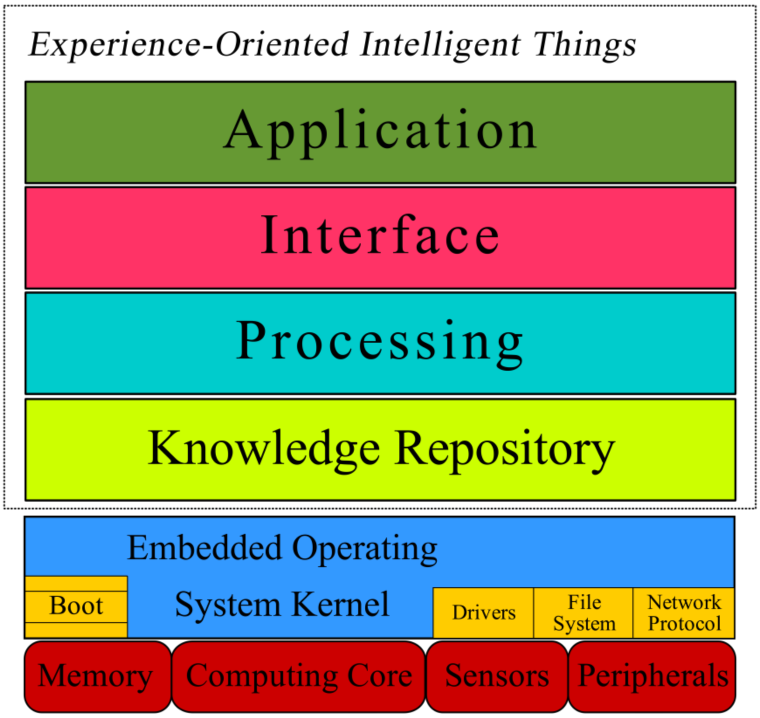

Fig. 2. Conceptual architecture of the EOIT.

Each one of the layers in the conceptual architecture can be briefly characterized as follows:

1) Application Layer

IoT applications run on this layer having access to the framework's whole functionalities. Data flow on this layer through EOIT's APIs (application program interface) to facilitate IoT solving problems, making decisions, and feeding back the EOIT with information based on their daily activities. Additionally, the feedback information can be either knowledge extracted already on the IoT, or raw data from IoT; which depends on the computing capability of certain IoT.

2) Interface Layer

The Interface Layer connects the framework with its outer environment which provides full functionalities of EOIT through a set of APIs.

3) Processing Layer

This layer is the central control of EOIT. In order to achieve knowledge extracting, storing, reusing, evolving, and sharing, a range of functionalities and mechanisms are attached to this layer, such as diagnosis, prognosis, inference, and knowledge management.

4) Knowledge Repository Layer

Experiential knowledge, as the most valuable resource, is stored and managed at this layer. In EOIT, a single decision event is captured and represented as a set of experience, and the knowledge carrying the decisional fingerprint of a particular thing is extracted from a number of such decisional sets of experience, and is organized as Neural Knowledge DNA (NK-DNA). This layer provides the functionalities of access, storage, and administration of knowledge. The central to the Knowledge Repository Layer concept of NKDNA is comprehensively introduced and tested in the study presented in (Zhang et al. 2017b). The NKDNA's four-element combination is able to carry detailed information of reinforcement learning and Markov Decision Processes (Zhang et al. 2017b).

Experimental Implementation

In this initial experimental implementation, we developed a smart phone application that reads the phone's three-axis accelerometer every 5 seconds to collect data that map driver's pedal and steering operations. We collected 560 sets of data for the experiment.

To predict the driver's actions, we used fully connected neural networks, recurrent neural networks (LSTM networks, more specifically), and the EOIT application respectively for comparison. The task is sample: by giving one set of data (i.e. three float numbers from three axes respectively of the accelerometer at time t), to predict the next set of data (i.e. the next three three-axis accelerometer numbers collected at time t+1). As it is a time series forecasting problem, the fully connect neural networks fail to perform. The LSTM networks are trained by 500 samples of the dataset after 30000 epochs with 256 units. Figure 3 shows its cost convergence during the training.

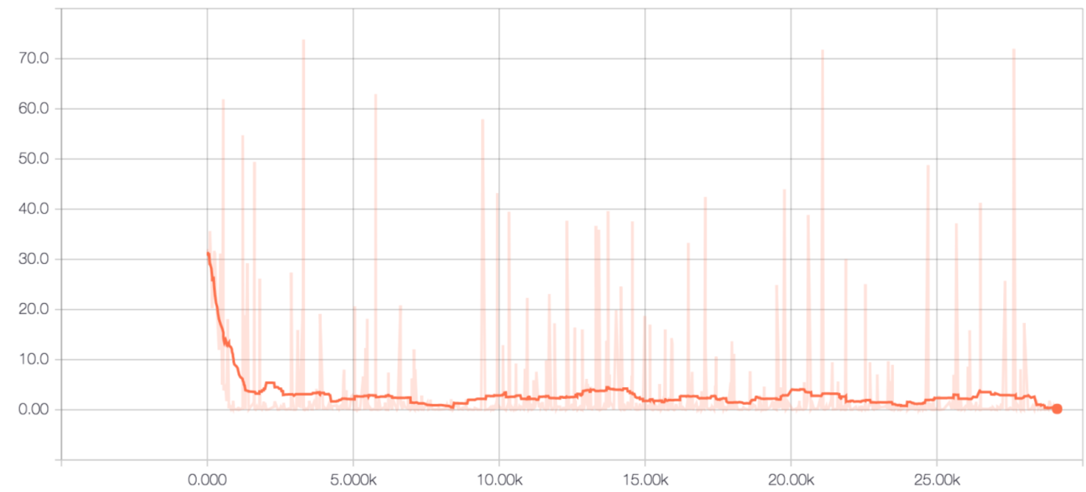

Fig. 3. Cost convergence of training the LSTM.

For our experience-based approach, we introduced a k-Nearest Neighbour (Altman 1992) like algorithm (we call it Triangle Mean Algorithm) to predict the next set of data. The Triangle Mean Algorithm measures the distance $d_E(X_i, X_j)$ between query points X_i and a set of training samples (i.e. the experience) X_j to predict the output points Y_i based on the mean of three nearest points (triangle-points T_i) of training samples. Algorithm 1 describes the details of the developed algorithm.

Algorithm 1 Triangle Mean Algorithm

Input:
(1) training samples { X_j, j = 1,…,M}
(2) query points {X_i, i = 1,…, N}
Output:
prediction points {Y_i, i = 1,…, N}

1: **for** each X_i $i \in 1 .. N$ **do**
2: **for** each X_j $j \in 1 .. M$ **do**
3: $d_E(X_i, X_j) = \sqrt{\sum_{k=1}^{3}(attr_k(X_i) - attr_k(X_j))^2}$
4: $T_i = \{ \ \mathrm{argmin}_{t \in 1..3} \ d_E(X_i, X_j) \mid T_i \in X_j \ \}$
5: prediction = $\frac{1}{3} \sum_{k=1}^{3} attr_k(T_i)$
6: **end for**
7: **end for**

To examine our experience based concept, 500 sets of data are used as the training samples and the rest 60 sets of data are used as testing queries for all three approaches (i.e. fully connected neural networks, LSTM networks, and our EOIT approach). However, as the fully connected neural networks failed in the examination (training cannot converge), the results illustrated below involve only the latter two approaches (see Figure 4).

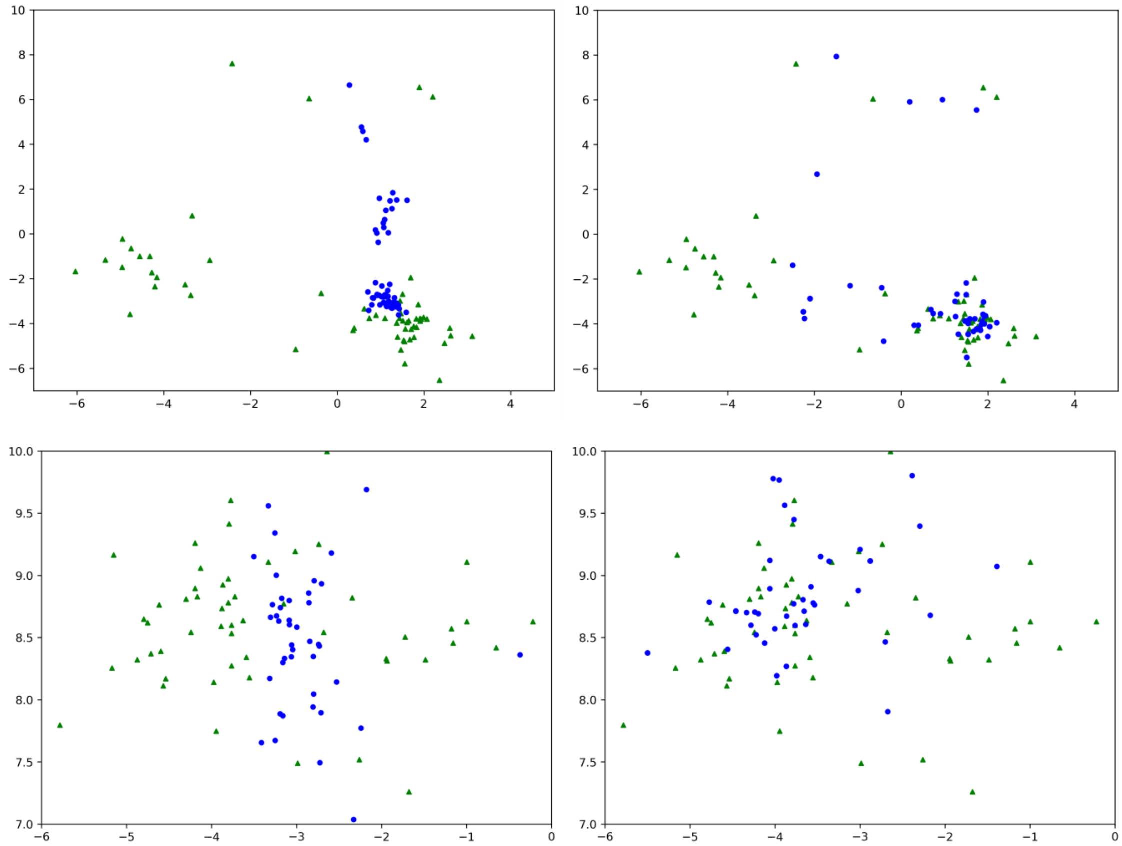

Fig. 4. Comparison of the predictions of LSTM networks (left) and the EOIT approach (right).

The upper row illustrates grand truth (green triangle) and predictions (blue dots) of the x and y axes data, while the bottom row presents the results for y and z axes. The results demonstrate that the predictions of EIOT approach (right) are closer to the grand truth than the ones of LSTM networks (left) are.

Figure 5 shows the sum of errors at three axes respectively. As it can be seen from the figure, the EOIT approach outperforms the LSTM networks, which further reinforces the results presented above in Figure 4.

At the end of this experiment, we measured the efficiency of our EOIT approach and LSTM networks. The results (Figure 6) demonstrate that the EOIT is about three times faster (30ms) than the LSTM networks (151ms) on our test computer for the same test dataset, providing more than two times closer predictions. The total error is measured as the sum of distances for each prediction of the 60 queries of the test dataset.

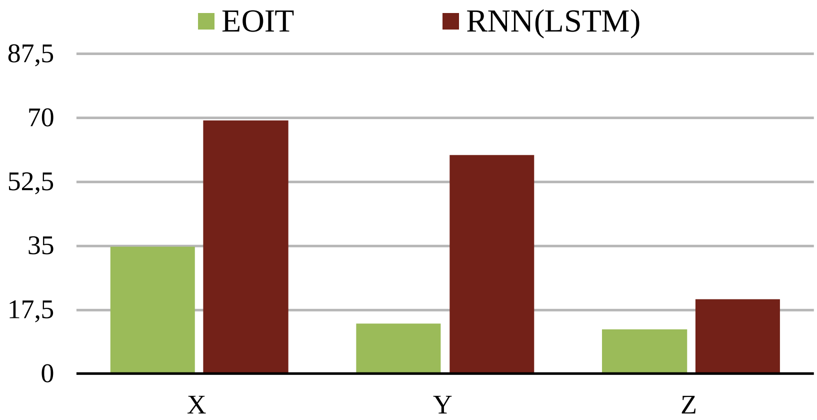

Fig. 5. The prediction errors at three axes.

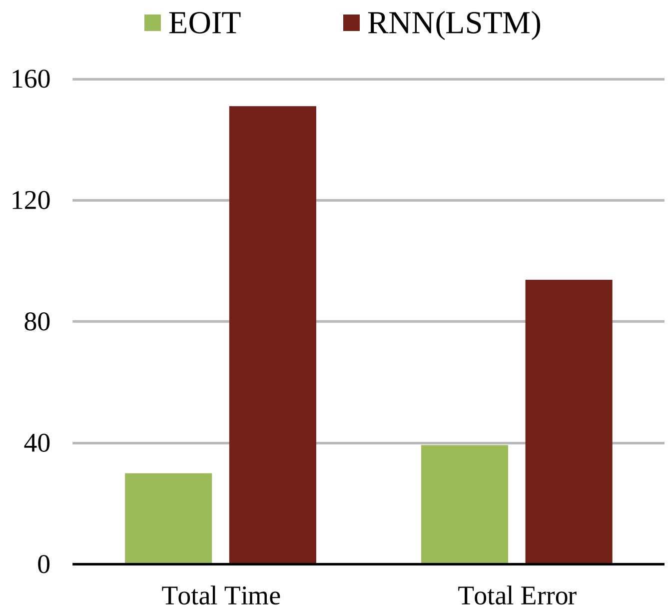

Fig. 6. The total time and total errors comparison.

Conclusions and Future Work

In this paper, we present the concept of an experience-based cognition framework EOIT, which acquires knowledge and understanding of the driving habits and behavioral characteristics of car drivers.

By catching car driving data, this approach helps to collect experience of drivers' car control operations, and eventually, it uses obtained experience for the driver's driving behavioral fingerprint extraction via the Triangle Mean algorithm. Our approach learns both steering and pedal (i.e. braking and acceleration) actions, and is demonstrated to perform faster and more accurately than deep learning based models (i.e. LSTM) on our dataset. In addition, using experience also allows our model to evolve in real time and be up-to-dated with high accuracy, which cannot be done in deep learning based models without time consuming retraining.

In the next step of the EIOT research refinement we will adopt it for mobile applications, expand data access interfaces to manage car control data from the vehicle electronic control unit (ECU), and develop inferential ability to be able to further manipulate the acquired knowledge.

Acknowledgement

This research was supported by Sichuan Science and Technology Program under the grant number 2019YFH0185.

REFERENCES

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician. 46 (3): 175–185. doi:10.1080/00031305.1992.10475879

Berndt, H., Emmert, J., & Dietmayer, K. (2008). Continuous Driver Intention Recognition with Hidden Markov Models. International IEEE Conference on Intelligent Transportation Systems (pp.1189-1194). IEEE.

Carvalho, E., Ferreira, B. V., Ferreira, J., Souza, C. D., Carvalho, H. V., & Suhara, Y., et al. (2017). Exploiting the use of recurrent neural networks for driver behavior profiling. International Joint Conference on Neural Networks (pp.3016-3021). IEEE.

Donges, E. (1978). A two-level model of driver steering behavior. Human Factors the Journal of the Human Factors & Ergonomics Society, 20(6), 691-707.

Hess, R. A., & Modjtahedzadeh, A. (1990). A control theoretic model of driver steering behavior. Control Systems Magazine IEEE, 10(5), 3-8.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780.

Hu, J., Xu, L., He, X., & Meng, W. (2017). Abnormal driving detection based on normalized driving behavior. IEEE Transactions on Vehicular Technology, 66(8), 6645-6652.

Jain, A., Koppula, H. S., Soh, S., Raghavan, B., Singh, A., & Saxena, A. (2016). Brain4cars: car that knows before you do via sensory-fusion deep learning architecture. arXiv preprint arXiv:1601.00740

Kuge, N., Yamamura, T., Shimoyama, O., & Liu, A. (2000). A Driver Behavior Recognition Method Based on a Driver Model Framework. Proceedings of the Society of Automotive Engineers World Congress.

Lecun Y, Bengio Y, Hinton G.(2015). Deep learning. Nature, 2015, 521(7553):436.

Liu, A. (2008). Modeling and prediction of human driver behavior. in Proc. 9th Int. Conf. Human-Comput. Interaction.

Macadam, C. C. (2007). Application of an optimal preview control for simulation of closed-loop automobile driving. IEEE Transactions on Systems Man & Cybernetics, 11(6), 393-399.

McRuer D.. (1980). Paper: human dynamics in man-machine systems. Automatica, 16(3), 237-253.

Meyer-Delius, D., Plagemann, C., & Burgard, W. (2009). Probabilistic situation recognition for vehicular traffic scenarios. IEEE International Conference on Robotics and Automation (pp.459-464). IEEE.

Morton, J., Wheeler, T. A., & Kochenderfer, M. J. (2017). Analysis of recurrent neural networks for probabilistic modeling of driver behavior. IEEE Transactions on Intelligent Transportation Systems, 18(5), 1289-1298.

Olabiyi, O., Martinson, E., Chintalapudi, V., & Guo, R. (2017). Driver action prediction using deep (bidirectional) recurrent neural network.

Oliver, N., & Pentland, A. P. (2000). Graphical models for driver behavior recognition in a SmartCar. Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE (pp.7-12). IEEE.

Sak, H., Senior, A., & Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. Computer Science, 338-342.

Saleh, K., Hossny, M., & Nahavandi, S. (2018). Driving behavior classification based on sensor data fusion using LSTM recurrent neural networks. IEEE, International Conference on Intelligent Transportation Systems (pp.1-6). IEEE.

Sanin C, Szczerbicki E.(2006) Using Set of Experience in the Process of Transforming Information into Knowledge. International Journal of Enterprise Information Systems, 2006, 2(2):45-62.

Sanin, C., Zhang, H., Shafiq, I., Waris, M. M., Oliveira, C. S. D., & Szczerbicki, E. (2018). Experience based knowledge representation for internet of things and cyber physical systems with case studies. Future Generation Computer Systems.

Schnelle, S., Wang, J., Su, H. J., & Jagacinski, R. (2016). A personalizable driver steering model capable of predicting driver behaviors in vehicle collision avoidance maneuvers. IEEE Transactions on Human-Machine Systems, PP(99), 1-11.

Wang, W., Xi, J., & Zhao, D. (2018). Learning and inferring a driver's braking action in car-following scenarios. IEEE Transactions on Vehicular Technology, PP(99), 1-1.

Zeng, X., & Wang, J. (2017). A stochastic driver pedal behavior model incorporating road information. IEEE Transactions on Human-Machine Systems, 47(5), 614-624.

Zhang, H., Li, F., Wang, J., Wang, Z., Sanin, C., & Szczerbicki, E. (2017a). Experience-oriented intelligence for internet of things. Journal of Cybernetics, 48(3), 162-181.

Zhang H, Sanin C, & Szczerbicki E. (2017b). Towards Neural Knowledge DNA. Journal of Intelligent & Fuzzy Systems, 2017, 32(2):1575-1584.

Zhang, H., Sanin, C., & Szczerbicki, E. (2010). Gaining knowledge through experience: developing decisional dna applications in robotics. Cybernetics & Systems, 41(8), 628-637.