MODELING DRIVER TAKEOVER INTENTION IN AUTOMATED VEHICLES WITH ATTENTION-BASED CNN ALGORITHM

Shantanu Gupta, Rohit Mishra, Yu-Hao Chang, Zheng Ma, Fenglong Ma, Yiqi Zhang*
Pennsylvania State University, University Park, PA 16803, USA
* Corresponding author: yuz450@psu.edu

In highly and fully automated vehicles (AV), drivers could divert their attention to non-driving-related activities. Drivers may also take over AVs if they do not trust the way AVs drive in specific driving scenarios. Existing models have been developed to predict drivers' takeover performance in responding to takeover requests initiated by AVs in semi-AVs. However, few models predicted driver-initiated takeover behavior in highly and fully AVs. The present study develops an attention-based multiple-input Convolutional Neural Network (CNN) to predict drivers' takeover intention in fully AVs. The results indicated that the developed model successfully predicted takeover intentions of drivers with a precision of 0.982 and an F1-Score of .989, which were found to be substantially higher than other machine learning algorithms. The developed CNN model could be applied in improving the driving algorithms of the AV by considering drivers' driving styles to reduce drivers' unnecessary takeover behaviors.

INTRODUCTION

With the advances in the automated vehicle technologies, researchers have contributed much effort to investigate ways to establish a framework that provides a reliable and comfortable experience of driving with automated vehicles. The Society of Automotive Engineers (SAE) defines the six levels of automated vehicles (SAE, 2018). Based on the definition, drivers do not need to perform driving tasks in fully automated vehicles. However, previous surveys indicated that most US drivers are afraid of riding with fully AVs (Khosla et al, 2020). Therefore, it is important to design trustworthy AVs to improve drivers' trust in AVs before its deployment on the market.

Previous empirical studies have found the impact of driver's driving styles and AV's driving styles on drivers' trust and behavior in AVs. For example, Ma and Zhang (2021) suggested that drivers showed higher trust in AVs when the AV's driving style is aligned with drivers' driving style, whereas they showed lower trust in AVs and increased takeover behavior frequency when AV's driving style is against the drivers' driving style. Price et al. (2016) studied the drivers' trust in different automated driving algorithms. They found that drivers' trust in the driving algorithms was influenced by drivers' driving styles. Lee et al. (2021) examined the drivers' trust in fully AV in intersection crossing scenarios with the driving styles of AV being manipulated (i.e., aggressive, moderate, and conservative). They found that conservative style increased the frequency and magnitude of accelerator pedal inputs; conversely, the aggressive style increased the frequency and magnitude of brake pedal inputs, which provides an indicator of driver's trust in automated driving styles. Although the behavioral studies brought insights into the development of guidelines for the design of AV's driving styles, driver models that predict drivers AV interaction performance are still lacking to optimize AV's driving styles in different scenarios so that adjustments could be made in a dynamic manner to enhance driver trust and reduce unnecessary takeover behaviors.

To date, models have been developed to predict drivers' takeover performance in response to the takeover requests initiated by AVs using deep learning (DL) algorithms such as Convolutional Neural Network (CNN). Braunagel et al. (2017) and Du et al. (2020) applied machine learning (ML) algorithms (e.g., Linear Discriminant, K-Nearest Neighbors, Support Vector Machine, Naïve Bayes, Random Forest) to classify drivers' takeover quality after receiving takeover requests from AVs when performing the non-driving related tasks (NDRTs). Deo et al. (2018) developed a Long Short-Term Memory (LSTM) model to predict drivers' readiness to take over the vehicle based on observable cues from in-vehicle vision sensors. Most of the reviewed models focused on the prediction of takeover reaction time in responding to takeover requests from AVs. Lotz et al. (2019) adopted support vector machine algorithms to forecast the takeover time from four defined classes of takeover durations. Grese et al. (2021) predicted how long the drivers will take over the AV after the takeover request is initiated with a neural network model.

Fewer models have predicted drivers' takeover intention and decision-making in AVs. One of the exceptions was the DL network developed by Pakdamanian et al. (2020) to predict driver's takeover intention along with takeover reaction time and takeover performance quality. They developed a neural network model using multimodal data including pre-driving questionnaires, driver's physiological measurements, and vehicle data. Their model shows good prediction accuracy on the takeover intention, takeover time and takeover quality after the driver receives the takeover request (TOR) from the computer program in the driving simulator.

In summary, the existing research efforts have been made to predict the drivers' takeover performance in AVs based on the quality, readiness index, time, and intentions of takeover and on the intentions of changing the lanes or braking and the intensity of braking. All these outputs were predicted using inputs ranging from the parameters such as eyes-on-the-road and traffic situation to driver's physiological data, vehicle parameters (velocity, acceleration, positional information), time to collision, and driver's facial images. However, fewer research efforts have been devoted to predicting drivers' takeover intentions. The model that was able to predict takeover intention focused on the passive takeover intention upon the takeover requests issued from automated vehicles. However,

the active takeover behavior in the human-AV interaction initiated by human drivers has not been analyzed. The model discussed in Pakdamanian et al. (2020) predicted driver's takeover intention with the movement data of the vehicle without considering the data of the surrounding vehicles and pedestrians near the AVs, which may affect driver's takeover intention in AVs. Moreover, since driver's driving styles have been found to influence drivers' takeover frequency in the fully AVs, this factor should be quantified to predict driver-initiated takeover behavior in AVs.

The main objective of the present study is to develop a deep learning model employing attention-based Convolutional Neural Network algorithm to predict drivers' takeover intention in fully AVs with data from a driving simulator experiment. Specifically, the developed model will be applied to predict active driver-initiated takeover behavior rather than passive driver takeover responses towards AV's takeover requests. The developed model can be modified and applied in embedded systems of AVs (Kato et al, 2018) to predict drivers' takeover intention in advance so that the AV system can adjust its driving algorithm accordingly to reduce unnecessary takeovers and improve the overall driving experience thus justifying the importance of this study.

Table 1. Features of the CNN Model

	Feature	Description
Input (Independent)	Distance Matrix (For vehicle and pedestrian)	This is a matrix of 8 values as shown in Figure 1. Each of the values will represent the relative position of the closest surrounding vehicles/hazards in the corresponding direction.
	Velocity Matrix (For vehicle and pedestrian)	This is a matrix of 8 values as shown in Figure 1. Each of the values will represent the relative velocity of the closest surrounding vehicle/hazards in the corresponding direction.
	Right of way	This will be a binary variable that indicates if the participant has the right of way at any given point of a scenario. For instance, if the participant wants to make a left turn at a solid green, the right of way value will be 0.
	Signal Light	0 – Signal not visible; 1 – green; 2 – Amber; 3 – Red
	Driving Style	The numeric answers to the Aggressive Driving Scale (ADS) questionnaire for each participant.
	Longitudinal Velocity (ft/s)	Velocity of the subject vehicle along the road
	Lateral Velocity (ft/s)	Velocity of the subject vehicle across the road
	Longitudinal Acceleration (ft/s^2)	Acceleration of the subject vehicle along the road
	Lateral Acceleration (ft/s^2)	Acceleration of the subject vehicle across the road
	Time to Collision	Time remaining for collision with the hazard, computed by the simulator
Output (Dependent)	Autonomous Mode	Binary variable indicating whether the vehicle is in AV mode or manual mode

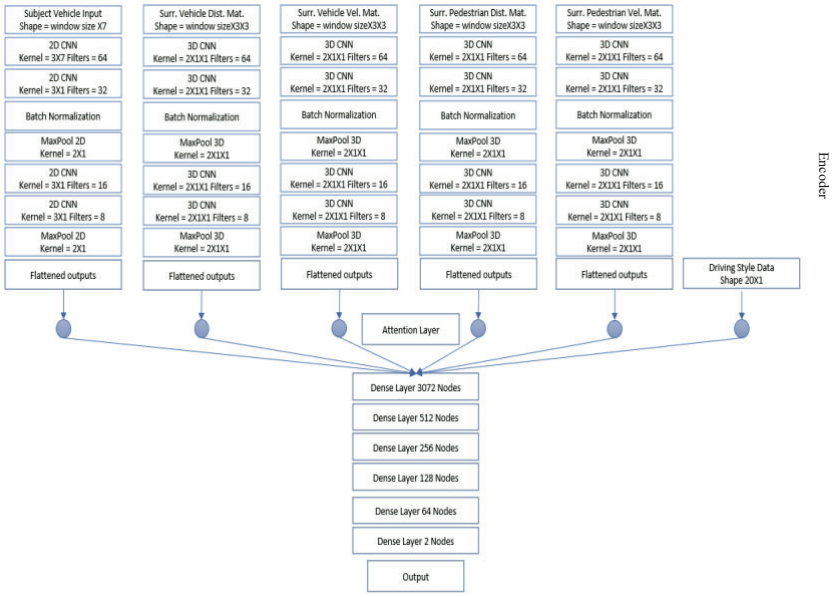

Figure 1. Model with Encoder and Dense Layer Structure

METHOD

Data Collection

The primary research dataset was obtained and reported in Ma and Zhang (2021). Forty-seven people participated in this study. Participants were native English speakers and have held a driver's license for at least 2 years. Their ages ranged from 18 to 39 years with an average age of 23.43 years (SD= 4.56) and an average annual driving mileage of 7547 miles (SD= 5215). A pre-screening procedure was used to classify drivers' driving styles via the Aggressive Driving Scale (ADS) (Krahe and Fenske, 2002). Participants were classified into twenty-two aggressive drivers and twenty-five defensive drivers. Each participant completed 12 scenarios in a level-5 automated vehicle simulated with the STISIM Drive® M300WS-Console System. The 12 scenarios consisted of 8 normal driving

scenarios, including turning corners on a green light to crossing intersections and stopping at stop signs; and 4 hazard scenarios, where a slow lead vehicle breaks down or a pedestrian runs into the road or a vehicle on the adjacent lane cuts in or a traffic jam occurs (Ma and Zhang, 2021). These scenarios were divided into 4 blocks with the sequence of being balanced with a Latin Square design across four experimental conditions. It was observed that more takeovers occurred during the hazard scenarios as compared to the normal ones, but they occurred at random times during the duration of the scenario and hence warranted the need for a ML algorithm.

Feature Engineering

The movement of the subject vehicle (SV) were automatically collected via the driving simulator. The metadata contained within the driving simulation was employed to capture the data of the surrounding traffic including other vehicles, pedestrians, road environment, and traffic signs and signals. The additional information collected provided useful raw features for training the CNN algorithms. As shown in **Table 1**, a set of features were defined to mimic the variables important in the process of maneuvering a vehicle.

The first feature was the surrounding vehicle distance matrix which represents the distance (in feet) of the SV to the closest vehicles in each of the 8 cardinal directions, represented by the left matrix in **Figure 2**. The right matrix in **Figure 2** is the surrounding vehicle velocity matrix which stores the corresponding relative velocities of the vehicles represented in the former matrix. Similar matrices were also generated to capture the movement of the surrounding pedestrians and were called the pedestrian distance matrix (Left matrix **Figure 3**) and pedestrian velocity matrix (Right matrix **Figure 3**), respectively.

80	10.1	30
15	**SV**	3
27	70	999

80	60	40
0	**SV**	10
0	30	10

Figure 2. Surrounding vehicle's distance and velocity matrix

80	16.5	60
20	**SV**	10
25	80	90

5	0	8
0	**SV**	3
2	0	0

Figure 3. Surrounding pedestrian's distance and velocity matrix

Another feature developed was the right of way. This concept is widely used while driving in the USA whereby a driver has the right of way in certain driving scenarios over the surrounding vehicles. This feature becomes important in the context of the current problem, as when driving the AV, the participants might have different levels of trust when the right of way is in their favor than when it is not. The last feature defined was the color of the signal light, defined using the metadata captured from the STISIM. This has four values which represent the different possibilities; no signal present or the light is either green, amber or red. This feature along with the previously defined features completely define the different scenarios during driving and combined with the SV movement data provide comprehensive information to train the model.

Data Preparation

All the ML algorithms were trained with two dimensional matrices whereas the data was engineered into groups of similar nature, each group having a specific dimensional matrix to train the DL algorithms. The SV movement data, right of way, and the signal light were grouped together and stored in a tabular form where each column represented a feature. The four matrices engineered (Two each for surrounding vehicles and pedestrians) were recorded at the same instances as the subject vehicle movement data. Therefore, those were stacks of matrices, where each matrix in a stack represents the position/velocity of the surrounding vehicle/pedestrian at a given time instant. Finally, the driver's driving style data obtained from the questionnaire was represented as a vector for each given participant and was constant for all scenarios involving them.

As the experiments were conducted with different people, they lasted different durations. However, the data was always collected in fixed time periods which resulted in the number of rows for each experiment to be different. To standardize the number of rows the data was picked from a window spanning a pre-decided time interval. For instance, if an experiment was 10 seconds long with each time step being 0.1 seconds, this would result in the experiment containing 100 observations (rows). A window size of 1 second (10 rows per window) running over this experiment would generate 99 datapoints each 1 second long (10 rows). Therefore, this allowed an increase in the total number of data points which could be used to train the model. The subsequent output for the data point would be a binary variable indicating whether the driver took over the automated vehicle within that time window.

The window size was an important parameter as it controls the number of data points generated. A larger window size ended up generating a lower number of datapoints. The length of the window also controlled the amount of information available to the algorithm to make its prediction. Finally, the data generated when the driver takes over the AVs can only be a small portion of the window corresponding to the time when the vehicle is being driven in the manual mode. These data points were recorded as takeover events and for longer windows, influencing the learning of DL algorithms. Therefore, cross validation was employed to optimize the window size.

Model Development

The SV movement data combined with the right of way and signal light features represents a matrix where each column is a feature, and each row is the record of all these features at time intervals of 0.1 seconds. To train the algorithm over this data, a 2D Convolution Network was deployed whereas to train over the surrounding vehicle/pedestrian distance/velocity matrices, a 3D Convolution Network was deployed because of its ability to handle video data for each of the four types of data. Therefore,

there are five distinct CNN blocks in the model. These blocks simplify their respective multidimensional input into a single dimensional output which then combined with the one-dimensional driving style data was further processed with feed forward neural network.

To support window sizes of varying lengths it was decided to use four convolutional layers and two max-pool layers for each block of CNN. The convolution and max-pool layers transform the output into smaller matrices of predefined sizes. A batch normalization layer was added to the model as it was observed that it improved the performance. An attention layer was also appended to the model and once the model is trained, each node of the attention layer gets a score, and these scores can provide information about the importance of each input in predicting the output. Attention here are the nodes between input and output layers which add weights to the input. **Figure 1** displays the five CNN blocks and the driving style vector in a network to train the model.

Four baseline machine learning models (**Table 3**) were also developed to compare and justify the development of CNN algorithm. They were trained on the dataset without considering the window size and using the same features as the CNN model. They use the 2D form of data where each point of the time-series was treated as a row of data and used to train these models. Each algorithm was trained to optimization by cross validating the different parameters associated with each of them. It was observed that logistic regression optimizes at 220 iterations and support vector machine gives the highest classification accuracy using rbf as the kernel function. A comparison of these models with the CNN model is discussed in the next section.

Model Training and Testing

The window size for cross validation was varied between 1 and 10 seconds with increments of 1 second each. Therefore, for the data obtained from each of the ten windows, separate instances of the model were trained using 5-fold cross validation. Each instance was trained for 100 epochs with each training batch containing 50 data points. To efficiently train the algorithm two loss functions were used, one was sparse categorical cross entropy loss function, and the other was contrastive loss function (Jianhao et al, 2019). Using the second function required breaking up the data into two parts as explained in Khosla et al. (2020). To update the weights of the model, Adam optimizer was used with a learning rate which was varied between 0.000001 and 0.00001 and was determined to be optimum at 0.00008. Hence, we trained 20 CNN models from T10 to T100 for both the loss functions (Table 2).

RESULTS

Model Performance Comparison

The performance of all the trained models were measured using the four metrics of a classification algorithm based on the confusion matrix of the test data. Specificity evaluates the model's ability to predict true negatives while sensitivity evaluates model's ability to predict the true positives. Precision is the ratio of true positives to the total predicted positives and F1-Score is the harmonic mean of precision and sensitivity. Since we used two loss functions, two models for each were trained and **Table 2** summarizes the performance of each model.

Table 2. Comparison of model performances

Model	Specificity	Sensitivity	Precision	F1 Score
T10-C	0.989	0.988	0.969	0.978
T10-NC	0.994	0.944	0.925	0.934
T20-C	0.992	0.987	0.977	0.982
T20-NC	0.970	0.962	0.914	0.937
T30-C	0.992	0.995	0.976	0.986
T30-NC	0.981	0.817	0.882	0.848
T40-C	0.994	0.996	0.982	0.989
T40-NC	0.982	0.973	0.946	0.959
T50-C	0.862	1.000	0.707	0.828
T50-NC	0.858	1.000	0.701	0.824
T60-C	0.882	1.000	0.738	0.849
T60-NC	0.878	1.000	0.732	0.845
T70-C	0.890	1.000	0.753	0.859
T70-NC	0.900	1.000	0.769	0.869
T80-C	0.996	0.990	0.988	0.989
T80-NC	0.993	1.000	0.980	0.990
T90-C	0.921	0.960	0.960	0.960
T90-NC	0.904	0.896	0.895	0.895
T100-C	0.930	0.998	0.827	0.905
T100-NC	0.918	0.998	0.802	0.889

Note. C denotes training with contrastive loss function; NC denotes training with other loss function; T10 denotes model with a time window of 1 seconds with the time step of 0.1 second.

While the sparse categorical loss function tries to find a hyperplane to separate the two classes, the contrastive loss function tries to increase the contrast in the classes by increasing the distance between them. The benefits of the latter are more pronounced for models with smaller time windows. The results manifest that the three models (i.e., T80-NC, T80-C and T40-C) performed better since their performance is consistent across all four metrics. The T40-C model was identified as the optimal model among the three best models, with its training time being the lowest followed by T80-NC and T80-C. Therefore, only T40-C was used for further analysis.

A comparative analysis of the T40-C CNN model with other baseline machine learning algorithms (without incorporating the window size feature) identifies the T40-C model as an efficient predictor of the takeover intention of the driver and justifies developing a deep learning algorithm. **Table 3** compares the classification metrics of these models.

Table 3. Comparison of CNN model to Baseline ML models

Model	Specificity	Sensitivity	Precision	F1 Score
CNN (T40-C)	0.994	0.996	0.982	0.989
Logistic Regression	0.853	0.821	0.841	0.831
Random Forest	0.960	0.998	0.959	0.978
SVM	0.951	0.595	0.920	0.723
Decision Tree	0.974	0.980	0.973	0.977

DISCUSSION

This literature talks about the process of developing a robust framework with a multiple-input Convolutional Neural

Network (CNN) model to predict a driver's takeover intention in fully AVs with drivers' driving style, the movement data of AVs, and four matrices of movement data of surrounding vehicles and pedestrians. It is one of the first DL models that could predict active takeover intention initiated by human drivers in AVs. The developed CNN model was able to learn the features of time series data to predict the takeover action in different time windows. Since the larger time windows can anticipate a longer future in the time series scenario, multiple window sizes were used to ensure the appropriate configuration of the CNN model. Several other baseline models were also trained to justify the need to build a highly complex CNN model. The T40-C model built using a contrastive loss function with a window size of 4 seconds was identified as the best CNN model with a prediction accuracy of 98.2%, a F1-Score of 0.989, and a specificity of 0.994 all of which are the highest in comparison to the performance of baseline models. Such high accuracy can be attributed to the fact that most takeovers occur in hazard scenarios where the features are significantly different from the other scenarios.

The built CNN model finds its application in improving the driving algorithms of the AVs in real time by matching them with the driving style of drivers. It can use the information captured in the event of an active takeover by the driver and adjust the driving algorithm of the AV to reduce the probability of takeover by the driver if similar circumstances arise again. This combination manifests a self-learning algorithm which will overtime improve the trust of drivers in AVs.

Effect of Window Size

The window size controlled the size of each data point which in turn affected how much information was given to the model to make an individual prediction. Moreover, processing data from a longer window would also require additional time thereby limiting the usefulness of the prediction. Therefore, the most effective window size would be a balance between model performance and the time required to make the prediction. However, in the current study, models obtained from window sizes of 4 and 8 seconds had very similar performance despite the later containing almost twice the amount of data. This suggests that the loss of trust in the AV is not dependent on the information far out in the past. Moreover, the drop in F1 score is not gradual. It has two peaks, at windows of lengths 4 and 8 seconds. It is not clear why this behavior is exhibited and would require further analysis.

Limitation and Future Work

Although the current work highlights the importance of understanding the driving style of the drivers to instill trust in AVs, there are certain limitations that need to be addressed. The data being used for the study is derived from a driving simulator study. The model needs to be validated in the future with naturalistic driver data in AV. Secondly, the prediction of whether a takeover will take place or not is of limited use in the real world. A more actionable prediction would be the probability with which a takeover will happen in each time window. This would allow researchers to focus on instances where there is a higher probability of loss of trust.

In conclusion, an attention-based CNN model was built to study the take-over behavior of drivers in a simulated automated vehicle. Data on the vehicle movement, driving style behavior and surrounding environment was used along with some feature engineering to develop a convolution neural network model. The task of this model was to predict whether a takeover by the driver happened within a time frame. The time window was varied to obtain the optimal window for which the prediction performance was maximum. Four machine learning models were also trained to compare the performance with the attention-based CNN model as baselines. It was observed that the CNN model showed a better performance in predicting driver takeover intentions than all baseline models.

ACKNOWLEDGEMENT

This material is based upon work supported by the National Science Foundation under Grant No. (1850002).

REFERENCES

Braunagel, C., Rosenstiel, W., & Kasneci, E. (2017). Ready for take-over? A new driver assistance system for an automated classification of driver take-over readiness. IEEE Intelligent Transportation Systems Magazine,9(4), 10–22.

Deo, N., & Trivedi, M. M. (2018). Looking at the driver/rider in autonomous vehicles to predict take-over readiness. Retrieved from http://arxiv.org/abs/1811.06047.

Du, N., Zhou, F., Pulver, E. M., Tilbury, D. M., Robert, L. P., Pradhan, A. K., & Yang, X. J. (2020). Predicting driver takeover performance in conditionally automated driving. Accident; Analysis and Prevention, 148(105748), 105748

Grese, J. M., Pasareanu, C., & Pakdamanian, E. (2021). Formal analysis of a neural network predictor inShared-control autonomous driving. AIAA Scitech 2021 Forum. Reston, Virginia: American Institute of Aeronautics and Astronautics.

SAE, (2018) Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles -SAE international. (J3016B). Retrieved March 16, 2021, from Sae.org website: https://www.sae.org/standards/content/j3016_201806/

Jianhao, Jing, Longqiang, Yi, Hanzhang, & Wanzhong. (2019). Control oriented prediction of driver brake intention and intensity using a composite machine learning approach. Energies, 12(13), 2483.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., … Krishnan, D. (2020). Supervised Contrastive Learning. Retrieved from http://arxiv.org/abs/2004.11362.

Krahé, B. and Fenske, I. (2002), Predicting aggressive driving behavior: The role of macho personality, age, and power of car. Aggr. Behav., 28: 21-29.

Lee, J. D., Liu, S.-Y., Domeyer, J., & DinparastDjadid, A. (2021). Assessing drivers' trust of automated vehicle driving styles with a two-part mixed model of intervention tendency and magnitude. *Human Factors*, 63(2), 197–209.

Lotz, A., & Weissenberger, S. (2019). Predicting take-over times of truck drivers in conditional autonomous driving. *In Advances in Intelligent Systems and Computing* (pp. 329–338). Cham: Springer International Publishing.

Ma, Z., & Zhang, Y. (2021). Drivers trust, acceptance, and takeover behaviors in fully automated vehicles: Effects of automated driving styles and driver's driving styles. *Accident Analysis & Prevention*, 159, 106238.

Pakdamanian, E., Sheng, S., Baee, S., Heo, S., Kraus, S., & Feng, L. (2020). DeepTake: Prediction of driver takeover behavior using multimodal data. Retrieved from http://arxiv.org/abs/2012.15441

Price, M. A., Venkatraman, V., Gibson, M., Lee, J., & Mutlu, B. (2016). Psychophysics of trust in vehicle control algorithms (No. 2016-01-0144). SAE Technical paper.

S. Kato et al., "Autoware on Board: Enabling Autonomous Vehicles with Embedded Systems," 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS), 2018, pp. 287-296.