Human Driver Behavior Classification from Partial Trajectory Observation

Angelos Mavrogiannis
MS Student
Intelligent Control Lab
Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
Email: angelosm@andrew.cmu.edu

Changliu Liu
Assistant Professor
Intelligent Control Lab
The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
Email: cliu6@andrew.cmu.edu

As autonomous vehicles are being tested on public roads, they must be able to share the road safely with human-driven vehicles. To ensure safety, autonomous vehicles must be capable of accurately estimating human drivers' intentions and their future trajectories. While there has been extensive research in this area, most of the existing approaches do not take into account the individual drivers' personalities and the patterns these personalities reflect on the trajectories of the vehicles. We aim to tackle this issue by proposing a novel method of extracting high-level features from raw vehicle trajectory data and classifying drivers into behavioral classes based on their level of aggressiveness. We demonstrate how the identification of a driver's behavior improves the accuracy of the short-term trajectory prediction problem by introducing a prior knowledge on their behavior.

Nomenclature

q_t^i State of agent i at time t
Q_t State of all agents at time t
$\xi_{0\to t}$ Historical trajectory information for agent i
$\Xi_{0\to t}$ Multi-agent historical trajectory information
$\mathcal{B}$ Behavioral space
b^i Behavioral class of agent i
B Behavioral classes of all agents

1 Introduction

Autonomous driving research has drawn a lot of attention and resources recently, both in the academia and in the industry. A major challenge impeding the massive deployment of autonomous vehicles in public roads lies in the uncertainty introduced by the presence of human drivers in mixed traffic scenarios. A human driver can be flexible to the surroundings and accurately infer the future behaviors of road agents given past experience, but a self-driving car is not yet capable of emulating this on-the-spot decision-making process. Furthermore, while autonomous vehicles might be able to successfully communicate with each other and plan their motion in a centralized manner in the future, they will still need to be able to anticipate the future actions of their human counterparts in order to ensure safety and become valuable members of the society. This is a very challenging problem due to the stochasticity and heterogeneity of human drivers, who can often act unexpectedly, leading to dangerous situations and traffic accidents. Research in vehicle trajectory prediction has been extensive, constantly advancing the state-of-the-art with the rise of deep learning methods. However, there has not been significant focus on the human personality perspective and its impact on road behaviors. While many of the accidents caused in roads can originate from spontaneous mistakes made by human drivers, another source is the reckless behavior exhibited by some drivers in a repetitive manner. For example, some drivers might often move from one lane to the other without a priori using a turn signal light or by taking steep turns instead of smoothly transitioning to the adjacent lane. As a result, behaviors like this might not always lead to accidents, however they increase the probability of a crash. Therefore, the identification of potentially risky drivers is crucial in order to accurately estimate the future actions of human drivers and plan the motion of autonomous vehicles ensuring safety.

Our research hypothesis is that leveraging inherent cues representative of drivers' personalities can lead to more accurate predictions for their future trajectories. The main contributions of this work can be described as following:

1. We propose a novel data-driven framework of identifying different behavioral classes of drivers and classifying them into these categories.
2. We do not fix the number of driving behaviors a priori, but we define it based on the available data.
3. The behavior classification framework is computation-

ally efficient and scales better with data compared to similar methods.

2 Related Work

We propose a framework for classifying human driver behaviors and we apply it to the problem of trajectory prediction. In this section, we discuss about related work both in the area of human driver classification and in the area of motion prediction.

2.1 Motion Prediction

Adapting the work of Rudenko et al. [1] in human motion prediction to vehicle trajectory prediction, we divide the problem of trajectory prediction into the following three components.

1. **Stimuli**, which include the agent's motion intent as well as other directly or indirectly observable features encoded in the environment. Information related to the agent's motion intent usually consists of historical trajectory information about its state, which can include positional coordinates, velocities or accelerations acquired via on-board sensors or measured from a global point of view (e.g. a camera-equipped drone from above). On the other hand, static and dynamic obstacles and constraints, including pedestrians and other road agents can be considered as environmental features that implicitly impact an agent's motion. We can also consider the different driving scenarios (urban driving, highway driving) as environmental features which also affect the behavior of road agents. Both the agent's motion intent and the environmental features can be semantic: expressed in a high-level context, like raw vehicle trajectory data, or auxiliary: constitute of encodings of various important information with fixed dimensionality and are often hard to interpret, such as data derived directly from a LiDAR sensor.
2. **Prediction**, which corresponds to the two major categories in vehicle motion prediction:
   (a) Maneuver prediction, where the output of the prediction consists of discrete states, such as driving straight, turning right or turning left at an intersection [2].
   (b) Trajectory prediction, where the exact future state of an agent is predicted. For example, in [3], the predicted values are the future lateral coordinate and the longitudinal velocity of a vehicle. Some other methods [4] propose a multi-modal solution using Gaussian mixture models over predicted states or sample from generative models [5].
3. **Modeling approach**, which addresses the way different methods represent and solve the motion prediction problem.
   (a) *Physics-based methods* predict future trajectories by developing handcrafted dynamical models, as in [6] and [7]. These methods can be accurate and reliable for a short prediction horizon, however they are known to underperform in long-term prediction problems and therefore they are gradually replaced by better-performing learning-based methods. A survey on physics-based methods can be found in [8].
   (b) *Data-driven methods* learn the motion models of road agents from observed trajectory data. Tran et al. [9] construct three-dimensional Gaussian process regression models from two-dimensional trajectory patterns and compare the likelihoods of the observation data for each individual regression model. Schlechtriemen et al. [10] use a Naive Bayes approach followed by a Hidden Markov Model (HMM) where the states of the model correspond to the maneuvers extracted from raw trajectory data and use this framework to detect lane changes. Liu et al. [11] address the decision making problem in highway driving as an optimal control problem, which is solved through a Hidden Markov Model formulation, defining the intention for a set of discrete maneuvers as the states and predicting future trajectories using an empirical model which contains adjustable parameters to accommodate the driver's time-varying behavior. However, most of these approaches do not model the interaction between different road agents in the future trajectories.
   (c) *Deep learning methods* also constitute data-driven methods, however they are mentioned separately, given their wide use and remarkable success in tackling trajectory prediction problems. More specifically, recurrent neural networks (RNN) have been used in vehicle trajectory prediction problems [12], [2], [4] and especially Long Short-Term Memory (LSTM) networks, given their ability to learn long-term dependencies between the inputs. Altché et al. [3] use an LSTM network, which receives information regarding an ego vehicle and a fixed number of surrounding vehicles and outputs future sequences of coordinates for the ego vehicle. Messaoud et al. [13] use an Encoder-Decoder RNN architecture with an Attention mechanism [14] in order to model the spatio-temporal interactions between a road agent and its surrounding vehicles and predict its future trajectory. In this work, all agents in the vicinity of the ego vehicle are considered for the prediction task. Other approaches [15], [16] combine Imitation Learning with Generative Adversarial Networks (GAN) under a framework [17] to predict future vehicle trajectories. Furthermore, methods combining RNN and Convolutional Neural Networks (CNN) emerged. Deo et al. [18] combined LSTM with CNN to predict future vehicle trajectories in highway driving scenarios. However, these methods are not considering the human personality factor and fail to capture underlying traits in the human

driver behavior which are responsible for repeating patterns, such as reckless driving. As a result, some of these methods only perform well on specific datasets or even in specific demografic regions. This motivates the study of human driver behavior classification, which is discussed in the next subsection.

2.2 Driving Behavior

The term vehicle behavior prediction in literature has been used in the same context as vehicle motion prediction, indicating that the predicted values are either future trajectories or discrete maneuvers. However, in this work, we use the term behavior to indicate the unique personality traits of human drivers, which influence their actions on the road.

There is a variety of factors that affect human driving behavior according to studies. Some factors include the driver's age, gender, personality, potential disabilities and other personal characteristics [19], while others include psychological aspects such as driving under the influence, drowsy driving and so on [20]. There have been some methods to classify drivers into categories based on vehicular factors such as positions, acceleration, speed, throttle responses, steering wheel measurements, lane changes, and brake pressure. More specifically, Chandra et al. [21] proposed a new metric to classify driver behaviors based on computational graph theory concepts and social traffic psychology. In [22], the inter-agent interactions are represented using unweighted and undirected traffic graphs, again for the purpose of classifying driver behaviors with an application to trajectory prediction. Similarly, Chandra et al. [23] used a combination of spectral graph analysis and deep learning to predict both future trajectories and road-agent behaviors. Finally, Cheung et al. [24] identified driver behaviors from vehicle trajectories via a user study and applied their method on motion planning. However, some limitations of these methods are that they are defining a priori the number of different driving behaviors, and they can be computationally intractable as the amount of data increases.

3 Problem Statement

Let q_t^i be the state of an agent i, with $q^i \in \mathbb{R}^2$. The state includes the lateral and longitudinal coordinates (x,y) of an agent i at time t, measured as shown in Fig. 1.

$$q_t^i = (x_t^i, y_t^i) \tag{1}$$

Let Q_t be the state of all n agents of interest at time t:

$$Q_t = \{(x_t^i, y_t^i), ..., (x_t^n, y_t^n)\} \tag{2}$$

Then, let $\Xi_{t_0 \to t-1}$ be the historical trajectory information for all n vehicles of interest from time t_0 to time $t-1$. For simplicity we assume $t_0 = 0$.

$$\Xi_{0 \to t-1} = (Q_0, Q_1, \ldots, Q_t - 1) \tag{3}$$

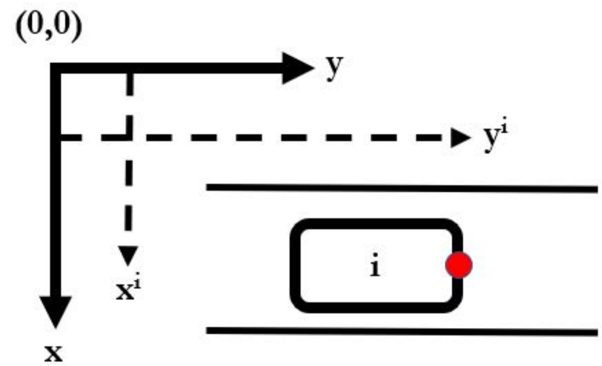

Fig. 1. The coordinates of an agent i are measured from the leftmost edge of the road laterally and from the beginning of the road segment under consideration, longitudinally, until the front center of the vehicle (represented by a red dot in the figure).

The solution of the trajectory prediction problem in a multi-agent setting lies in the estimation of the probability of observing a future sequence $\Xi_{t \to T}$ given a partial observation in the present $\Xi_{0 \to t-1}$:

$$P(\Xi_{t \to T} | \Xi_{0 \to t-1}) \tag{4}$$

According to our research hypothesis, an agent's behavior heavily influences its future trajectory, therefore we can rewrite the previous probability in the following way:

$$P(\Xi_{t \to T} | \Xi_{0 \to t-1}) = \sum_B P(\Xi_{t \to T} | \Xi_{0 \to t-1}, B) P(B | \Xi_{0 \to t-1}) \tag{5}$$

where B is a tuple consisting of all the behavioral classes of all n agents and $\mathcal{B}$ is a discrete behavioral space that we are going to define.

$$\begin{aligned} &B = (b^1, b^2, \ldots, b^n) \\ &b^i \in \mathcal{B} \end{aligned} \tag{6}$$

Expanding the second probability from the right hand size of equation (5) we get:

$$P(B | \Xi_{0 \to t-1}) = P(b^1, b^2, \ldots, b^n | \Xi_{0 \to t-1}) = \prod_{i \in n} P(b^i | \Xi_{0 \to t-1}) \tag{7}$$

Therefore, we need to estimate the probability of an agent i belonging to a specific behavioral class $b \in \mathcal{B}$ given a multi-agent trajectory sequence $\Xi_{0 \to t}$ in the past and expand this computation for all n agents.

Now we focus on the first probability of the right hand side of equation (5), and by decomposing the historical multi-agent information, we can write it as following:

$$P(\Xi_{t \to T} | \Xi_{0 \to t-1}, B) = P(Q_t, Q_{t+1}, \ldots, Q_T | \Xi_{0 \to t-1}, B) \tag{8}$$

The basis of our research hypothesis lies on the fact that an agent's behavior heavily influences its trajectory, therefore we can assume that given the knowledge of the behavioral classes B of all agents, the future trajectories only depend on the last element of $\Xi_{0\to t-1}$. This assumption is then compatible with the following equation:

$$P(Q_t, Q_{t+1}, \ldots, Q_T | \Xi_{0\to t-1}, B) = P(Q_t, Q_{t+1}, \ldots, Q_T | B) \tag{9}$$

Using the chain rule, we get the following equation:

$$P(Q_t, Q_{t+1}, \ldots, Q_T | B) = P(Q_T | Q_{T-1}, Q_{T-2}, \ldots, Q_t, B) \cdot P(Q_{T-1} | Q_{T-2}, Q_{T-3}, ..., Q_t, B)(Q_1 | Q_t, B) \tag{10}$$

We now use a Markov assumption, according to which the state of all agents Q_{t+1} at every timestep only depends on the value of the state during the previous timestep Q_t, but not on the values of the state before the previous timestep $Q_{t-1}, Q_{t-2} \ldots, Q_0$.

$$Q_{t+1} \perp\!\!\!\perp Q_{t-1} | Q_t \tag{11}$$

Based on the Markov assumption we can write the final equation of our model:

$$P(\Xi_{t\to T} | \Xi_{0\to t-1}) = \prod_T P(Q_{t+1} | Q_t, B) = \prod_T \prod_{i\in n} P(q_{t+1} | Q_t, B) \tag{12}$$

4 Methodology

4.1 Overall Method

The proposed method can be decomposed into the following components:

1. **Preprocessing** where we prepare the data.
2. **Feature Extraction** where we compute high-level features available from the data.
3. **Low-dimensional representation**, which is achieved by applying the Principal Component Analysis (PCA) [25] algorithm.
4. **Behavioral Clustering**, where we form the behavioral clusters by applying the K-means [26] algorithm on the low-dimensional data distribution.
5. **Online Behavior Classification**, which is achieved by comparing the performance of K-Nearest Neighbors (KNN), Logistic Regression (LR) and a Deep Neural Network (DNN) on classifying a driver's behavior given their descriptive statistics.
6. **Trajectory Prediction**, where we use a Long Short-Term Memory (LSTM) [27] neural network in order to infer the future position of an agent given information about nearby agents as well as their behavioral labels.

The pipeline is illustrated in Fig. 3.

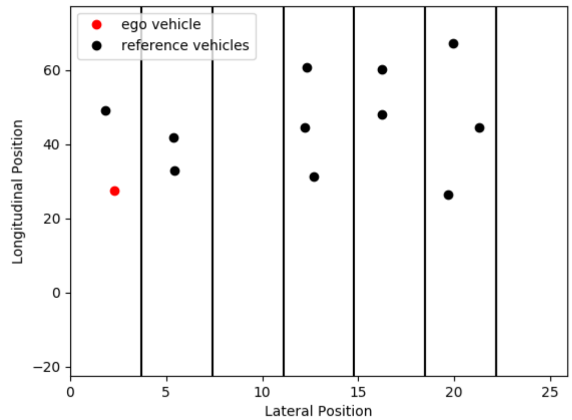

Fig. 2. Screenshot taken from the simulator we developed on Python. The red dot represents the ego vehicle and the black dots represent the reference vehicles which are nearby the ego vehicle for each timestep.

4.2 Preprocessing

We begin by identifying the active vehicles in our data for every timestep, and we save their vehicle ID as well as their trajectory information. Additionally, we develop a simulator on Python that receives as input the number of lanes and the coordinates of every vehicle for all timesteps and illustrates the trajectories of the vehicles, so that we can compare the different driving behaviors that emerge from our model. As seen in Fig. 2, the red dot signifies the ego agent, while the black dots represent the nearby agents. We preprocess the data in order to facilitate the performance of the Principal Component Analysis. For every feature f_i of the data, if μ_i is its mean and σ_i is its standard deviation throughout the entire dataset, we subtract the mean from it and divide by the standard deviation.

$$f_i' = \frac{f_i - \mu_i}{\sigma_i} \tag{13}$$

In this way, every feature in the data has a mean value $\mu_i' = 0$ and standard deviation $\sigma_i' = 1$.

4.3 Feature Extraction

To define the behavioral space $\mathcal{B}$, we will attempt to uncover underlying patterns in the trajectories demonstrated by drivers, which indicate their aggressiveness on the road. We believe that there are some descriptive statistics that capture the entire trajectory sequence of a certain driver and therefore accurately describe their behavior on the road. These are shown in Table 1.

For the lateral direction of motion, we are interested in the average values of an agent's velocity and acceleration as an indication of the deviation of an agent from its current lane. As the arithmetic mean does not capture all the important characteristics that we believe are indicative of aggression, we also take into account the maximum values of

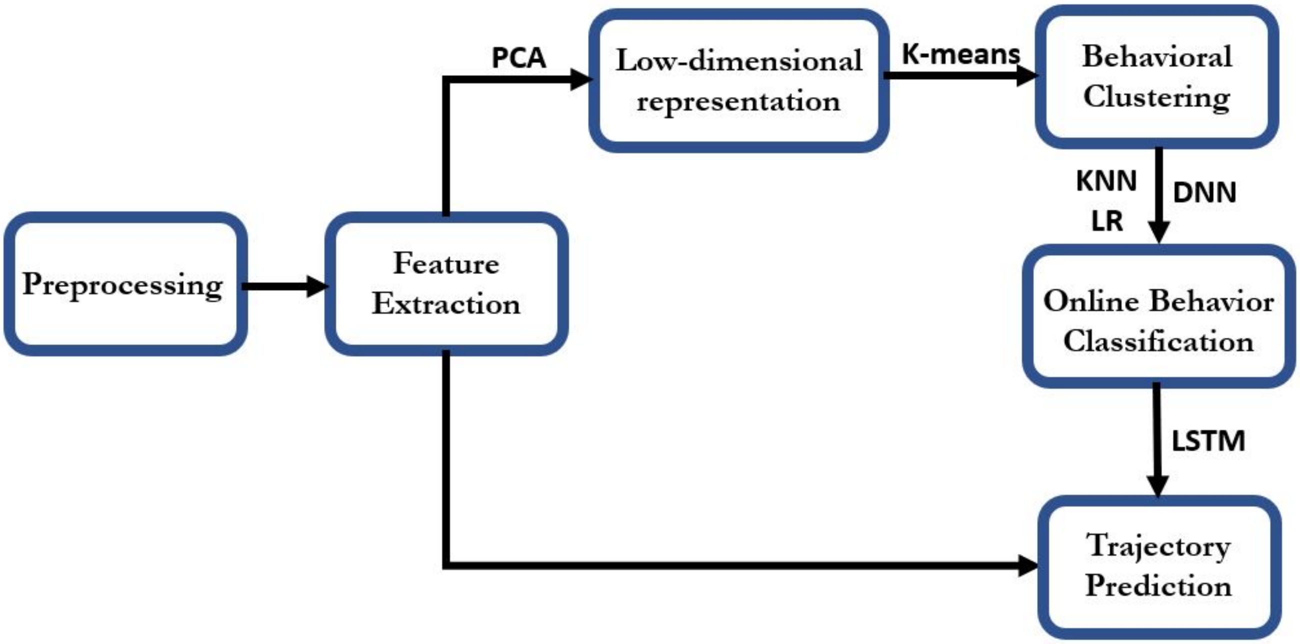

Fig. 3. The pipeline of the proposed method starts with preprocessing the data and extracting features from it, applying PCA to reach a low-dimensional representation and forming behavioral clusters with K-means. Finally, K-nearest neighbors, Logistic Regression and a Deep Neural Network are used for online inference of a behavioral label, which is fed to a Deep Neural Network which predicts future trajectories.

Table 1. Descriptive statistics for every driver used in the Principal Component Analysis

Statistical Metric	Symbol	Units
Average lateral velocity	$\mid \bar{\upsilon}_x \mid$	$\frac{m}{s}$
Maximum lateral velocity	$\mid \upsilon_x^{max} \mid$	$\frac{m}{s}$
Standard deviation of lateral velocity	$\sigma(\upsilon_x)$	$\frac{m}{s}$
Average longitudinal velocity	$\bar{\upsilon}_y$	$\frac{m}{s}$
Maximum longitudinal velocity	υ_y^{max}	$\frac{m}{s}$
Standard deviation of longitudinal velocity	$\sigma(\upsilon_y)$	$\frac{m}{s}$
Average lateral acceleration	$\mid \bar{a}_x \mid$	$\frac{m}{s^2}$
Average longitudinal acceleration	$\bar{a}_y$	$\frac{m}{s^2}$
Average speed relative to the traffic flow	υ_y^{flow}	$\frac{m}{s}$
Average speed relative to the preceding vehicle	υ_y^{prec}	$\frac{m}{s}$

an agent's lateral velocity which correspond to potential aggressive lane changes as well as the standard deviation of lateral velocity in order to capture the fluctuation of an agent's speed as a metric of irregular behavior.

For the longitudinal direction of motion, we focus on the average values of an agent's velocity and acceleration, as well as the maximum and standard deviation of its velocity. The intuition behind this lies on the fact that aggression does not only depend on large values of speed, but also on other factors, such as patterns of continuous deceleration and acceleration. We believe that an irregular behavior can be also identified by comparing an agent's road behavior with the behavior of its surrounding agents. As such, we define a bounded box surrounding an agent for every timestep, and we measure the average longitudinal velocity for all agents in the box. Then, we subtract this quantity from our agent's velocity and we use this difference to quantify the speed difference between its velocity and the velocity of its neighboring vehicles. Finally, as an extension to the previous metric, we also take into account the longitudinal velocity of an agent relative to the longitudinal velocity of the leading agent (the preceding vehicle). A large value of an agent's velocity and a small value of its leading agent's velocity corresponds to an aggressive behavior.

4.4 Low-dimensional representation

After acquiring a set of 10 descriptive statistics for every driver in the dataset, we project these statistics to a low-dimensional space using Principal Component Analysis (PCA) in order to gain some further insight in determining the most dominant features that correspond to aggressive behaviors. We focus on the two most dominant components, which capture the largest percentage of the total variance of the data.

4.5 Behavioral Clustering

Having acquired a low-dimensional space consisted of the two principal components of the data, we apply the K-means clustering algorithm in order to define an optimal way of identifying the different behavioral classes according to the distribution of the data in the dimensional space defined by the two dominant principal components. To determine the optimal number of clusters k for the K-means algorithm we use two different methods:

1. The elbow method [28]
2. The silhouette method [29]

According to the elbow method, we calculate the Within-Cluster-Sum of Squared errors (WCSS) [30] for different values of k and we choose the value of k that minimizes the errors. However, the errors keep decreasing as we increase k, so we need to bound k so that we do not have a large number of clusters but at the same time we need to maintain a low value of errors. We believe that a low value of k (e.g. $k = 2$) is not enough to capture the diversity of the driving behaviors, as it is possible that the aggressive drivers constitute only a small part of the total number of drivers in the dataset, however they get classified as outliers and eventually get merged with other drivers who are not equally aggressive due to the low value of k. Therefore, we set the following constraint to the selection of k:

$$k > 2 \tag{14}$$

As an additional method to validate the optimal number of clusters k, we use the Silhouette method. The silhouette method measures how similar a point is to its own cluster compared to other clusters. A high value is desirable and indicates that the point is placed in the correct cluster. Again, we calculate the silhouette value for different numbers of k and we choose the optimal number of clusters.

4.6 Online Behavior Classification

At this point we have defined the behavioral space $\mathcal{B}$, we know the behavioral labels B of all agents, and therefore we can estimate the probability:

$$P(b^i|\Xi_{0\rightarrow t-1}) \tag{15}$$

To this end, we are going to use a set of supervised learning algorithms to infer the behavioral class of a driver given his descriptive statistics which were established during the feature extraction stage. More specifically we train a K-Nearest Neighbors model, a Logistic Regression model, and a Deep Neural Network in a multiclass classification setting. The neural network consists of two hidden layers with 256 hidden units on each layer, followed by *ReLU* activation functions and a $softmax$ layer in the output layer. The models receive the descriptive statistics of a training set of drivers based on partial trajectory observation, as well as their behavioral class labels from the K-means algorithm and predict the behavioral class labels of a number of drivers in a test set. Finally, we compare these algorithms and discuss the trade-off between computational complexity and model accuracy.

4.7 Trajectory Prediction

To test our research hypothesis, we are going to apply the Vehicle Behavior Classification framework we developed to a trajectory prediction problem, estimating the probability from equation (12):

$$P(q_{t+1}|Q_t, B) \tag{16}$$

To this end, we are going to use a Long Short-Term Memory (LSTM) neural network that receives information about the states of the ego vehicle q_t^{ego}, which represents the autonomous vehicle making the prediction, the states of a set of reference vehicles in the vicinity of the ego vehicle Q_t, as well as the behavioral labels of these vehicles B and predicts the future state of the ego vehicle q_{t+1}^{ego} at a predefined short-term time horizon. Following the work in [3], we consider a set of nine reference vehicles, which form the following tuple:

$$(f, r, l, b, ff, fr, fl, br, bl) \tag{17}$$

Table 2. The set of reference vehicles in the vicinity of the ego vehicle for the task of trajectory prediction.

Description	Symbol
Preceding vehicle on current lane	f
Nearest vehicle on the right lane	r
Nearest vehicle on the left lane	l
Following vehicle	b
Preceding vehicle of f	ff
Preceding vehicle of r	fr
Preceding vehicle of l	fl
Following vehicle of r	br
Following vehicle of l	bl

Table 2 describes each of the reference vehicles used. For every timestep, we extract the states of these reference vehicles and save them during the preprocessing stage, and when one of these vehicles does not exist at a certain timestep, we consider the states as zero. A spatial example of an ego vehicle and the reference vehicles is shown in Fig. 4. The motivation behind the selection of these vehicles is based on the realistic assumption that the ego vehicle making the prediction can only acquire information regarding the vehicles in its proximity, due to the limitations of its sensors. The addition of the vehicle ff in front of the leading vehicle f of the ego vehicle is an attempt of estimating the traffic flow for the current lane, as, for example, in a traffic jam situation, f could be moving very slowly but at the moment ff starts to move, this indicates f will also start to move and as a result the ego vehicle will follow up shortly.

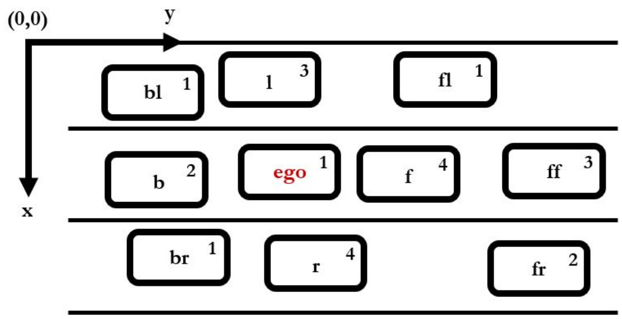

Fig. 4. An example illustrating the ego vehicle making predictions and the reference vehicles considered in its vicinity. Each vehicle has a number on the upper right part, corresponding to its behavioral cluster.

The neural network consists of a layer of 256 LSTM cells, followed by a fully connected layer of 128 units and *ReLU* activation and the output layer which results in two outputs, the lateral and the longitudinal position of the predicted vehicle in the next timestep.

The neural network predicts the state of the ego vehicle q_{t+1}^{ego} at the next timestep, but it can also predict its state further in the future in the following way: We acquire the predicted coordinates of the ego vehicle for timestep $t+1$, and by inspecting our data, we observe the active reference vehicles in the vicinity of the ego vehicle at timestep t. Therefore, we use the neural network to predict their coordinates at the next timestep $t+1$ and by doing so we have predicted Q_{t+1}. If we repeat this procedure by feeding the predicted state information to the network, we can predict for more timesteps, however, it is evident that if we attempt to predict for too many timesteps, the error will propagate and the accuracy will drop significantly as time flows.

5 Experiments and Results

The dataset we are using to evaluate the proposed method is the Interstate 80 (I-80) freeway dataset from the Next Generation SIMulation (NGSIM) [31] program. The dataset consists of 2355 trajectories corresponding to unique drivers, which were obtained from video analysis. The video was recorded on eastbound I-80 in the San Francisco Bay area in Emeryville, CA, on April 13, 2005. The road segment under consideration is approximately 500 meters in length and consists of six freeway lanes including a high-occupancy vehicle (HOV) lane. The dataset provides a total of 45 minutes of raw vehicle trajectory data, segmented into three 15-minute periods: 4:00 p.m. to 4:15 p.m., 5:00 p.m. to 5:15 p.m., and 5:15 p.m. to 5:30 p.m. Some of the advantages of this dataset are that it has been widely studied in literature, and, except for the coordinates of the vehicles, it also provides information about the preceding and following vehicles of each vehicle, as well as lane identifiers. Finally, it offers the coordinates of the vehicles in the format we described during the problem statement, therefore it is suitable for evaluating our method.

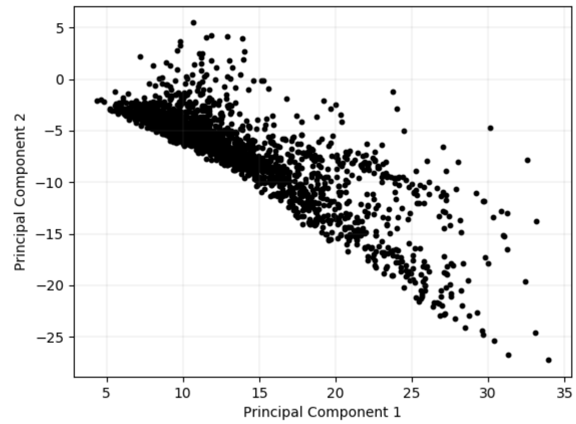

Fig. 5. First and second principal components of the PCA algorithm, capturing 54% of the total variance of the data.

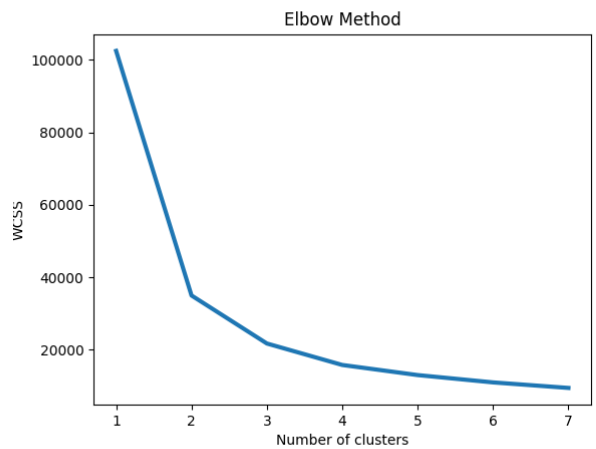

Fig. 6. Within-Cluster Sum of Squared errors for different values of k according to the elbow method.

After preprocessing the data and performing feature extraction, we apply the PCA algorithm on the extracted descriptive statistics. We retain the two first principal components, as they capture 54% of the total variance of the data. The low-level representation we get from the PCA algorithm is shown in Fig. 5, where we observe a pattern of linearity between Principal Components 1 and 2. As we can see in Fig. 6 the recommended optimal number of clusters is $k=2$, since an elbow at the curve usually represents the point at which we start to have diminishing returns. However, according to our constraint, the number of clusters should satisfy the condition:

$$k > 2 \tag{18}$$

Therefore, we use the Silhouette method as an additional tool to define the optimal number of clusters. Respecting the constraint we set on k and ignoring the first peak at $k=2$, we can see in Fig. 7 that the optimal number of clusters is the second largest silhouette value:

Fig. 7. Silhouette values for different number of clusters k.

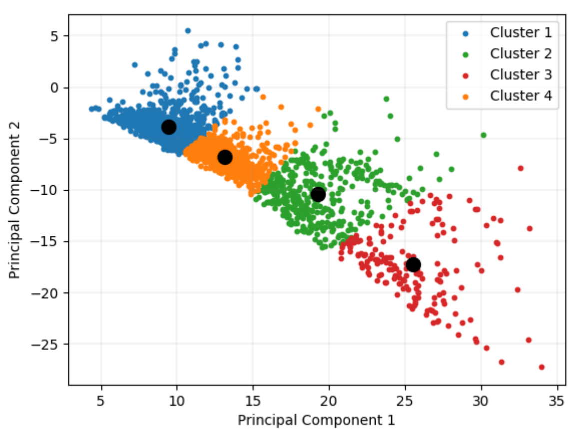

Fig. 8. The four behavioral clusters formed using K-means. The black dots represent the center of each cluster, respectively.

$$k = 4 \tag{19}$$

Proceeding with $k = 4$ as the number of clusters, we apply the K-means clustering algorithm on the low-dimensional space and we get the four clusters shown in Fig. 8. To evaluate the defined behavioral classes, we compare the ten statistics used in the Principal Component Analysis and examine how these values differ throughout the four clusters. We show the results from the comparison for three representative statistics (υ_y^{max}, $|\upsilon_x^{max}|$, υ_y^{flow}) which we believe are the most indicative of the differences between the four behaviors. Observing Fig. 9, we can see that the maximum longitudinal velocity as well as the relative velocity to the traffic flow increases from cluster 1 to cluster 4, however the maximum lateral velocity is showing a decreasing trend. We can correlate these results with the distribution we saw in the dimensional space defined by principal components 1 and 2. It is clear that the most dominant feature is the longitudinal velocity, which corresponds to Principal Component 1 and increases throughout the cluster, whereas the lateral velocity corresponds to Principal Component 2, and shows a slightly decreasing tendency. This can be explained rationally, considering that drivers who move very fast on the road in the longitudinal direction of motion tend to stay in the current lanes and avoid lane changes more than the drivers who move in a slower speed. Overall, the driving behavior from cluster 1 to cluster 4 ranges from timid to aggressive.

At this point we know the behavioral classes of all drivers in the dataset based on our previous analysis, and we want to train a model that learns to map the descriptive statistics of an agent to its behavioral class. We split the descriptive statistical data in a training and test set, keeping 80% of the drivers for training and 20% of them for testing. We train a Logistic Regression model, a K-Nearest Neighbors model with k-fold cross validation, and a neural network consisting of two hidden layers with *ReLU* activation functions and a $softmax$ layer in the output layer. The learning rate is $\eta = 0.001$, the objective function is the Cross Entropy Loss and the network is trained for 200 epochs with the Adam optimizer. Simulating a realistic scenario, an ego vehicle navigating in public roads will need to obtain information through its sensors and perform the behavior classification in an online setting, therefore except for the accuracy of the model, we also care about the computational complexity. Therefore, after conducting several experiments training all models and testing multiple times, we present the results in Table 3. As we can see in the table, as well as in Fig. 10 and Fig. 11, KNN and Logistic Regression are very fast to train and demonstrate good results in the classification problem (Accuracy: 87% and 86% respectively), while the DNN has a larger runtime ($1.4s$) but achieves almost perfect accuracy (97%). The final choice of the classification algorithm depends on the trade-off between the accuracy and the computational complexity, since the runtime will increase as the amount of data increases. For the purpose of our study, we will continue with the DNN and prioritize accuracy over time complexity.

Table 3. The set of reference vehicles in the vicinity of the ego vehicle for the task of trajectory prediction.

Algorithm	KNN	Logistic Regression	DNN
Runtime (s)	0.025	0.06	1.4
Accuracy (%)	87	86	97

Now that we have defined the behavioral space and developed a mechanism for human driver behavior classification, we can use the behavioral classes of the agents to predict their future trajectories. We train our LSTM network on 80% of the data for 30 epochs using the Adam optimizer and a learning rate of $\eta = 0.0005$. The duration of the sequences considered is 20 seconds and the prediction horizon is 1 second in the future. During the first experiment, the network receives as input the states of the ego vehicle and its 9 surrounding vehicles, while during the second experiment, the network additionally receives the behavioral

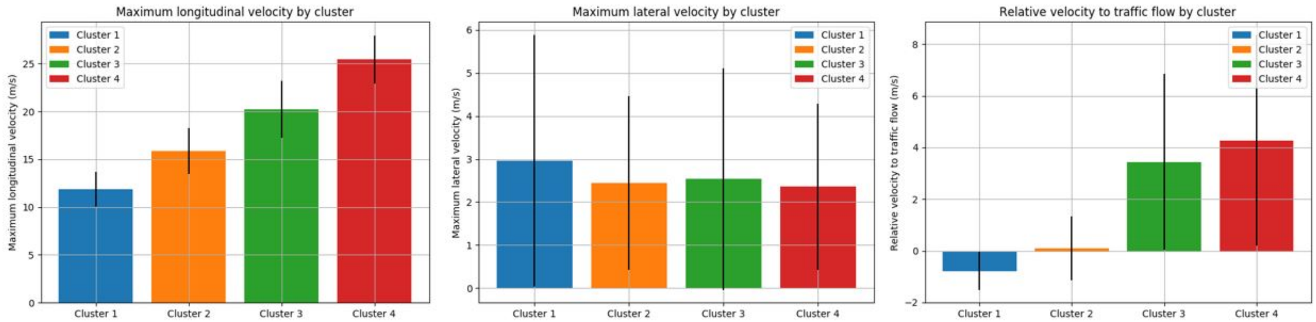

Fig. 9. Bar charts illustrating the distribution of three important descriptive statistics of drivers throughout the four different behavioral clusters.

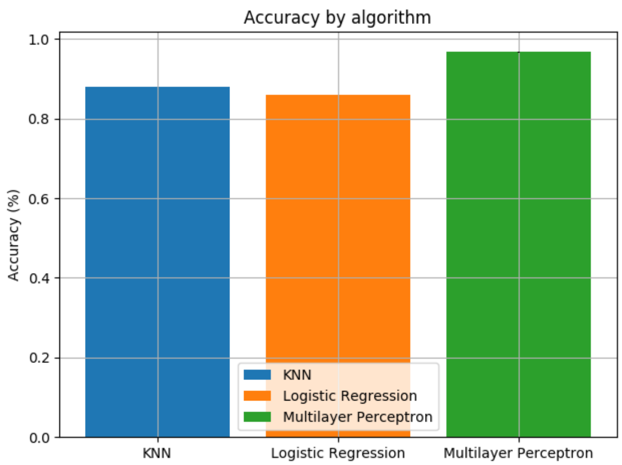

Fig. 10. Accuracy of the algorithms used for the online behavior classification.

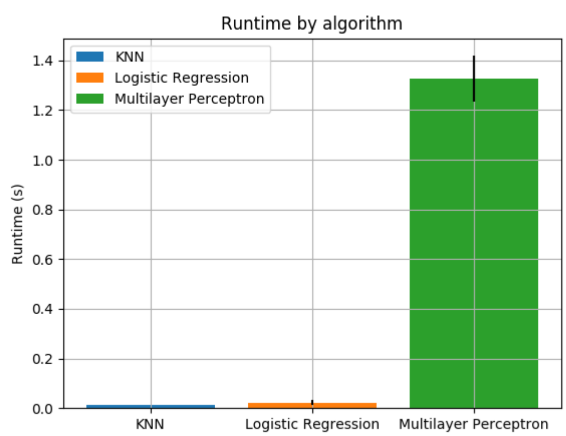

Fig. 11. Runtime of the algorithms used for the online behavior classification.

classes of the respective agents. Testing on the remaining 20% of the data, we observe that the addition of the behavioral classes increases the accuracy of the prediction by 56% longitudinally and 22% laterally. The Root Mean Squared Error (RMSE) per vehicle is used as the comparison metric, as shown in Table 4.

Table 4. Root mean squared error values per vehicle for 1 second prediction horizon

	Without Behavioral Classes	With Behavioral Classes
Longitudinal $RMSE$ (m)	5.60	2.44
Lateral $RMSE$ (m)	0.92	0.71

6 Conclusion and Future Work

We proposed a framework for identifying human driver behaviors in behavioral classes, classifying drivers in them, and using them to predict their future trajectories. We demonstrated that the addition of the behavioral classes to our LSTM predictor improves the accuracy of the short-term future predictions both in the longitudinal and in the lateral direction of motion. However, our method has some limitations. There is an implied linearity assumption behind the use of PCA for projecting the descriptive statistics to a low-dimensional space. In the future, we could try to use a Variational Autoencoder (VAE) to project the descriptive statistics to a latent space and consider non-linearities as well. Furthermore, our model has only been evaluated on the NGSIM I-80 dataset, so we need to apply it on other publicly available datasets as well in order to generalize its validity. Finally, as part of future work, we could improve the prediction part of our framework, using a more advanced neural architecture, such as an Encoder-Decoder model, and extend the prediction horizon.

References

[1] Rudenko, A., Palmieri, L., Herman, S., Kitani, K. M., Gavrila, D. M., and Arras, K. O. "Human motion trajectory prediction: A survey". *arXiv preprint arXiv: 1905.06113*.

[2] Phillips, D. J., Wheeler, T. A., and Kochenderfer, M. J., 2017. "Generalizable intention prediction of human drivers at intersections". In 2017 IEEE Intelligent Vehicles Symposium (IV), pp. 1665–1670.

[3] Altché, F., and de La Fortelle, A., 2017. "An lstm network for highway trajectory prediction". In 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), pp. 353–359.

[4] Deo, N., and Trivedi, M. M., 2018. "Multi-modal tra-

jectory prediction of surrounding vehicles with maneuver based lstms". *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1179–1184.

[5] Gupta, A., Johnson, J. E., Fei-Fei, L., Savarese, S., and Alahi, A., 2018. "Social gan: Socially acceptable trajectories with generative adversarial networks". *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2255–2264.

[6] Xie, G., Gao, H., Qian, L., Huang, B., Li, K., and Wang, J., 2018. "Vehicle trajectory prediction by integrating physics- and maneuver-based approaches using interactive multiple models". *IEEE Transactions on Industrial Electronics,* **65**, pp. 5999–6008.

[7] Veeraraghavan, H., Papanikolopoulos, N., and Schrater, P., 2006. "Deterministic sampling-based switching kalman filtering for vehicle tracking". In 2006 IEEE Intelligent Transportation Systems Conference, pp. 1340–1345.

[8] Schubert, R., Richter, E., and Wanielik, G., 2008. "Comparison and evaluation of advanced motion models for vehicle tracking". *2008 11th International Conference on Information Fusion*, pp. 1–6.

[9] Tran, Q., and Firl, J., 2014. "Online maneuver recognition and multimodal trajectory prediction for intersection assistance using non-parametric regression". In 2014 IEEE Intelligent Vehicles Symposium Proceedings, pp. 918–923.

[10] Schlechtriemen, J., Wedel, A., Hillenbrand, J., Breuel, G., and Kuhnert, K., 2014. "A lane change detection approach using feature ranking with maximized predictive power". In 2014 IEEE Intelligent Vehicles Symposium Proceedings, pp. 108–114.

[11] Liu, C., and Tomizuka, M., 2016. "Enabling safe freeway driving for automated vehicles". In 2016 American Control Conference (ACC), pp. 3461–3467.

[12] Zyner, A., Worrall, S., Ward, J., and Nebot, E., 2017. "Long short term memory for driver intent prediction". In 2017 IEEE Intelligent Vehicles Symposium (IV), pp. 1484–1489.

[13] Messaoud, K., Yahiaoui, I., Verroust-Blondet, A., and Nashashibi, F., 2020. "Attention Based Vehicle Trajectory Prediction". *IEEE Transactions on Intelligent Vehicles*, Apr.

[14] Bahdanau, D., Cho, K., and Bengio, Y., 2015. "Neural machine translation by jointly learning to align and translate". *CoRR*.

[15] Bhattacharyya, R. P., Phillips, D. J., Wulfe, B., Morton, J., Kuefler, A., and Kochenderfer, M. J., 2018. "Multi-agent imitation learning for driving simulation". *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1534–1539.

[16] Si, W., Wei, T., and Liu, C., 2019. "Agen: Adaptable generative prediction networks for autonomous driving". pp. 281–286.

[17] Ho, J., and Ermon, S., 2016. "Generative adversarial imitation learning". In NIPS.

[18] Deo, N., and Trivedi, M. M., 2018. "Convolutional social pooling for vehicle trajectory prediction". *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1549–15498.

[19] Wei-hua, Z., 2012. "Selected model and sensitivity analysis of aggressive driving behavior".

[20] Dai, J., Teng, J., Bai, X., Shen, Z., and Xuan, D., 2010. "Mobile phone based drunk driving detection". In 2010 4th International Conference on Pervasive Computing Technologies for Healthcare, pp. 1–8.

[21] Chandra, R., Bhattacharya, U., Mittal, T., Bera, A., and Manocha, D., 2020. "Cmetric: A driving behavior measure using centrality functions". *arXiv preprint arXiv:2003.04424*.

[22] Chandra, R., Bhattacharya, U., Mittal, T., Li, X., Bera, A., and Manocha, D., 2019. "Graphrqi: Classifying driver behaviors using graph spectrums". *arXiv preprint arXiv:1910.00049*.

[23] Chandra, R., Guan, T., Panuganti, S., Mittal, T., Bhattacharya, U., Bera, A., and Manocha, D., 2019. "Forecasting trajectory and behavior of road-agents using spectral clustering in graph-lstms". *arXiv preprint arXiv:1912.01118*.

[24] Cheung, E., Bera, A., Kubin, E., Gray, K., and Manocha, D., 2018. "Identifying driver behaviors using trajectory features for vehicle navigation". *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3445–3452.

[25] F.R.S., K. P., 1901. "Liii. on lines and planes of closest fit to systems of points in space". *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science,* **2**(11), pp. 559–572.

[26] Macqueen, J., 1967. "Some methods for classification and analysis of multivariate observations". In In 5-th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297.

[27] Hochreiter, S., and Schmidhuber, J., 1997. "Long short-term memory". *Neural computation,* **9**, 12, pp. 1735–80.

[28] Thorndike, R. L., 1953. "Who belongs in the family". *Psychometrika*, pp. 267–276.

[29] Rousseeuw, P., 1987. "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis". In Journal of Computational and Applied Mathematics.

[30] Muça, M., Kutrolli, G., and Kutrolli, M., 2015. "A proposed algorithm for determining the optimal number of clusters". *European Scientific Journal, ESJ,* **11**.

[31] Halkias, J., and Colyar, J., 2006. *Interstate 80 Freeway Dataset*.