



PBE: Driver Behavior Assessment Beyond Trajectory Profiling

Bing He², Xiaolin Chen¹, Dian Zhang¹(✉), Siyuan Liu³, Dawei Han⁴,
and Lionel M. Ni²

¹ College of Computer Science and Software Engineering, Shenzhen University,
Shenzhen, China

zhangd@szu.edu.cn

² Department of Computer and Information Science, University of Macau,
Macau SAR, China

³ Smeal College of Business, Pennsylvania State University,
State College, PA, USA

⁴ Auto Insurance Department, China Pacific Insurance Company, Shenzhen, China

Abstract. Nowadays, the increasing car accidents ask for the better driver behavior analysis and risk assessment for travel safety, auto insurance pricing and smart city applications. Traditional approaches largely use GPS data to assess drivers. However, it is difficult to fine-grained assess the time-varying driving behaviors. In this paper, we employ the increasingly popular On-Board Diagnostic (OBD) equipment, which measures semantic-rich vehicle information, to extract detailed trajectory and behavior data for analysis. We propose PBE system, which consists of Trajectory Profiling Model (PM), Driver Behavior Model (BM) and Risk Evaluation Model (EM). PM profiles trajectories for reminding drivers of danger in real-time. The labeled trajectories can be utilized to boost the training of BM and EM for driver risk assessment when data is incomplete. BM evaluates the driving risk using fine-grained driving behaviors on a trajectory level. Its output incorporated with the time-varying pattern, is combined with the driver-level demographic information for the final driver risk assessment in EM. Meanwhile, the whole PBE system also considers the real-world cost-sensitive application scenarios. Extensive experiments on the real-world dataset demonstrate that the performance of PBE in risk assessment outperforms the traditional systems by at least 21%.

Keywords: Driver behavior analysis · On-Board Diagnostic (OBD)

1 Introduction

Nowadays, the number of traffic accidents increases rapidly every year [6, 16]. Meanwhile, researchers have found that the driver behavioral errors caused more than 90% of the crash accidents [13], served as the most critical factor leading to the crash accidents. Therefore, how to effectively analyze the driver behavior

and assess the driver risk plays a significant role in travel safety, auto insurance pricing and smart city applications.

In the last decades, the significance of this task has led to numerous research efforts [15, 16]. Most of the previous work used GPS from vehicle [27], various sensors (e.g., magnetic and accelerometer sensors) from smartphone [5] and cameras [21] to collect data for analysis. Generally, when dealing with high-dimension and heterogeneous data, these work usually fails to take the fine-grained driver actions into consideration. Therefore, the prediction and evaluation of the driver behavior is limited. Besides, most traditional work does not consider the time-varying driving behaviors, making the driver risk assessment not sufficient.

To overcome the above drawbacks, we develop PBE system, which is able to fine-grained analyze the driving behavior based on the increasingly popular On-Board Diagnostic (OBD) equipments¹. Each vehicle in our experiment is integrated with such an OBD device. So we have not only GPS-related information, but also semantic-rich vehicle information including engine speed and so on. Some recent work [6, 20] also explores OBD. But, they usually focus on each OBD data tuple not from the trajectory perspective, which can consider the relationship among tuples and analyze from a global view for a better assessment.

Our PBE system aims to build a 3-tier model: Trajectory Profiling Model (PM), Driver Behavior Model (BM) and Risk Evaluation Model (EM). PM utilizes our insight from the data (the alarm information of OBD) to predict the trajectory class for profiling. It is able to remind drivers of danger in real-time. Besides, the labeled trajectories can be utilized to boost the training of BM and EM, when partial data is missing. BM evaluates the driving risk by fine-grained behavioral information from the trajectory perspective. EM combines the driver-level demographic information and BM's trajectory-level evaluation, to provide a comprehensive assessment for each driver to denote his/her risk. Besides, the time-varying driving pattern is also incorporated in EM. Meanwhile, PBE fully employs a cost-sensitive setting to satisfy the real-world application requirements, e.g., to lower the cost of misclassifying high risk as low risk in the real-time alarming system and auto insurance pricing scenario.

Overall, the main contributions are listed as follows. (1) PBE builds a real-time system via OBD device to remind drivers of danger. (2) Beyond fine-grained trajectory profiling results, PBE integrates the time-varying patterns and driver-level demographic information, to provide comprehensive evaluation scores for drivers. (3) We deploy the cost-sensitive setting to provide the practical analysis of drivers in the real-world application scenarios. (4) We perform extensive experiments using real-world OBD data. The performance of PBE system in risk assessment much better outperforms the traditional systems, by at least 21%.

¹ https://en.wikipedia.org/wiki/On-board_diagnostics.

2 Related Work

The existing work usually used GPS [27] records of a vehicle to generate the trajectory and mobility pattern for driver behavior analysis, due to the easy accessibility of GPS [27]. However, it is hard for these work to capture fine-grained driving actions. Besides, other work utilized smartphones (embedded with GPS and inertial sensors) [14, 23] and camera image information [21]. But, some require the installation of external cameras in vehicles, which brings concerns on cost and privacy. Alternatively, Chen et al. [6] used OBD data tuples to judge the driving state. Furthermore, incorporated with OBD, Ruta et al. [20] also used other kinds of data like map and weather information to infer the potential risk factors. However, they mainly only emphasize on each data tuple. Different from the previous work, via OBD, we extract the fine-grained driving-action-related features to analyze drivers on a trajectory level.

Concerning the driver behavior analysis techniques, fuzzy logic [5, 20] and statistical approaches [18] were explored. But, they need to manually set the rules. Besides, Bayesian Network [26] and its variants (Hidden Markov Model (HMM) [8, 21], Dynamical Bayesian Network [1]) were used to find the inner relationship between the driving style and sensor data for the driving behavior inference. However, they have practical challenges due to the model complexity and the required large amount of data. Additionally, some work used AdaBoost [6] and Support Vector Machine [26] classifiers to determine the driving state. Although they can achieve high precision sometimes, these work fails to consider the cost-sensitive setting with the real-world application requirement. Meanwhile, traditional trajectory classification methods [2] mainly utilized HMM-based model, which are difficult to capture the driver behaviors when encountering the fine-grained multi-dimension data. On the other hand, time-series classification can also be used to classify driving trajectories for the behavior pattern analysis, e.g., the 1-nearest neighbor classifier [24, 25]. But, the trajectories in our applications are quite different from time series with each point having multi-dimension points rather than only real values. Unlike the mentioned approaches, PBE considers the cost-sensitive setting and time-varying pattern, and analyzes comprehensively from multiple perspectives of the trajectory and driver level.

3 Preliminary

3.1 Data Description

OBD is an advanced on-board equipment in vehicles to record data. Each OBD data tuple x is defined as $\langle u_x, t_x, lon_x, lat_x, \phi_x, \psi_x \rangle$ where: (1) u_x is the driver identification; (2) t_x is the data recording timestamp (in second); (3) lon_x, lat_x are the longitude and latitude location record where x is created; (4) ϕ_x ($\phi_x = [v_x, a_x, \omega_x, \Omega_x]$) is a four-dimensional vector representing the real-time *physical driving state* of *speed, acceleration, engine speed* (Round Per Minutes (RPM))

Table 1. Data description (the semantic driving state is set by domain experts).

Physical driving state	Semantic driving state	Description
Speed	Vehicle speeding (<i>vs</i>)	Speed higher than the road speed limit after matching with road types by GPS
Acceleration	Abnormal acceleration (<i>aa</i>)	Acceleration value $\geq 1.8 \text{ m/s}^2$
	Abnormal deceleration (<i>ad</i>)	Acceleration value $\leq -1.8 \text{ m/s}^2$
Engine speed	Engine high RPM warning (<i>ehrw</i>)	Engine speed higher than the default upper engine speed of a vehicle
	Abnormal engine speed increase (<i>aesi</i>)	Engine speed increases sharply in a short time
Vehicle angular velocity	Sharp turn (<i>st</i>)	Vehicle angular velocity $\geq 30 \text{ Rad/s}$
	Lane change (<i>lc</i>)	$10 \text{ Rad/s} < \text{vehicle angular velocity} < 30 \text{ Rad/s}$

and *vehicle angular velocity* (Radian per second (Rad/s)); (5) ψ_x is a seven-dimensional vector representing the **semantic driving state** to denote the real-time *warning message* about the vehicle. It is derived from physical driving state, where $\psi_x = [vs_x, aa_x, ad_x, ehrow_x, aesi_x, st_x, lc_x]$ (More details are in Table 1 and the value type is binary (i.e., 1 means driver u is in this driving state at time t , and vice versa.). Besides, OBD can offer the **crash alarm** message to denote whether the car is highly likely to have a crash accident or not². Each data tuple z is defined as $\langle u_z, t_z, c_z \rangle$ where: (1) u_z, t_z are similar to aforementioned identification u_x and timestamp t_x ; (2) c_z is the crash alarm. Like mentioned ψ_x , c_z also uses the binary value to denote the state.

Then, given a driver’s massive OBD driving data, we analyze a driver’s behavior by *trajectory* [27] with the following definition:

Definition 1 (TRAJECTORY). Given a driver’s physical driving state record sequence $S = x_1x_2 \dots x_n$ and a time gap³ Δt , a subsequence $S' = x_i x_{i+1} \dots x_{i+k}$ is a trajectory of S if S' satisfies: (1) $v_{x_{i-1}} = 0, v_{x_i} > 0, v_{x_{i+k}} > 0, v_{x_{i+k+1}} = 0$; (2) if there exists a subsequence $S'' = x_j x_{j+1} \dots x_{j+g} \in S'$, where for $\forall 0 \leq q \leq g, v_{x_{j+q}} = 0$, then, $t_{x_{j+g}} - t_{x_j} \leq \Delta t$; (3) there is no longer subsequences in S that contain S' , and satisfy condition (1)(2).

A trajectory essentially leverages the speed and time constraints to extract reliable record sequences in the huge-volume driving records for effective studies.

² The OBD equipment made by JUWU IoT Technology Company (www.szjuwu.com) is incorporated with multiple processing units and domain experts’ knowledge to decide the crash alarm state upon receiving sensor data.

³ Four minutes is usually selected by domain experts for practical applications.

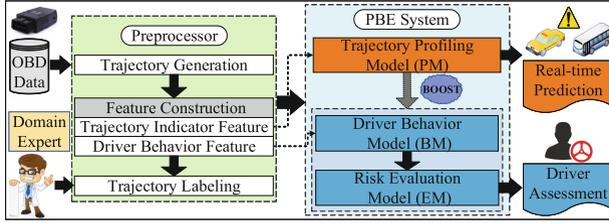


Fig. 1. Architecture of PBE system.

3.2 Problem Formulation

Given a driver set $U = \{u_i\}$ and their historical OBD data $X = \{x_i\}$, $Z = \{z_i\}$:

- (1) How to profile driving trajectories?
- (2) Based on trajectories, how to model drivers' driving risk?
- (3) How to assess drivers as safe, risky or dangerous?

3.3 System Overview

In this subsection, we present PBE system as our solution to the problem. As shown in Fig. 1, it consists of four major components: **Preprocessor** performs *preprocessing* of OBD data, including generating trajectories, features and labels. The labels come from the claimed insurance data and domain experts. Concerning the *correlation* and *causality* to crash accidents, the generated features are divided into *trajectory indicator features* and *driver behavior features*. Trajectory indicator features are those trajectory variables (e.g., crash alarms) which indicate a vehicle is apt to crash accidents. They have no interpretation for the driving behavior. While, driver behavior features (e.g., abnormal accelerations/decelerations) denote driving actions during a trajectory, served as the possible reasons for crash accidents. **Trajectory Profiling Model (PM)** leverages trajectory indicator features to predict the trajectory class quickly for real-time alarming systems. When data is incomplete, PM's predicted label is able to give a boost for the latter training of BM and EM as a trajectory's pseudo label. **Driver Behavior Model (BM)** utilizes driver behavior features to model driving risk for behavior analysis from the trajectory level. Finally, **Risk Evaluation Model (EM)** computes drivers' risk evaluation scores considering both the time-varying trajectory-level pattern and demographic information.

4 Preprocessor

Preprocessor takes OBD data as the input and performs the following tasks to prepare the data for future processing:

Trajectory Generation reads a driver's OBD records to generate trajectories according to Definition 1. The filtering of noisy data tuple is conducted with a heuristic-based outlier detection method through speed information [27].

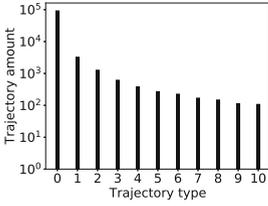


Fig. 2. Trajectory amount distribution.

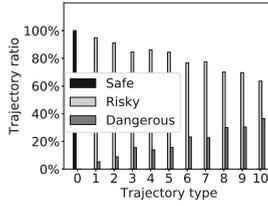


Fig. 3. Trajectory ratio distribution.

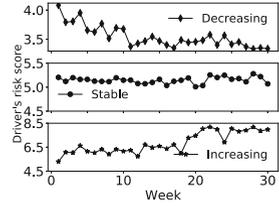


Fig. 4. Time-varying pattern of driver's risk score.

Feature Construction computes features with a trajectory S ($S = x_1x_2 \dots x_n$):

(i) Trajectory Indicator Features: First, we utilize trajectory beginning time t_{x_1} and ending time t_{x_n} to query which crash alarm records exist during the trajectory period, and construct the crash alarm record sequence $\mathcal{Z} = z_1z_2 \dots z_m$. Then, we compute features: (1) *trajectory's running time* ($t_{x_n} - t_{x_1}$); (2) *trajectory's distance* ($\sum_{1 \leq i \leq n-1} \frac{1}{2}(v_{x_i} + v_{x_{i+1}})(t_{x_{i+1}} - t_{x_i})$); (3) *Crash Alarm Counts per trajectory (cac)* $cac = \sum_{1 \leq i \leq m} 1(c_{z_i} = 1)$ where $1(\cdot)$ is an indicator function.

(ii) Driver Behavior Features: Driver behavior features (π_S , a eleven-dimensional vector) is defined as $\langle \bar{k}, dsc_q \rangle$ where:

$$(1) \bar{k} = \frac{\sum_{1 \leq i \leq n-1} \frac{1}{2}(k_{x_i} + k_{x_{i+1}})(t_{x_{i+1}} - t_{x_i})}{t_{x_n} - t_{x_1}}, k \in \{v, a, \omega, \Omega\},$$

represents the *average speed/acceleration/engine speed/vehicle angular velocity*;

$$(2) dsc_q = \frac{\sum_{1 \leq i \leq n} 1(q_{x_i} = 1)}{t_{x_n} - t_{x_1}}, q \in \{vs, aa, ad, erhw, aesi, st, lc\},$$

is the *Driving State Count per unit of time* for different semantic driving state q .

Trajectory Labeling sets the real-world ground truth label y_S of trajectory S by domain experts from insurance and transportation companies. There are three-class labels: **Safe Class (SC)**, **Risky Class (RC)** and **Dangerous Class (DC)**. Dangerous Class means the vehicle has crash accidents during the trajectory period according to auto insurance accident records. For the remaining trajectories with no accidents, domain experts judge them into Safe and Risky Class according to the driving smoothness of each trajectory.

5 Trajectory Profiling Model (PM)

In this section, we first conduct data statistics about generated trajectories to find data insights. Then based on the discovered insights, we develop two PMs of decision stump and decision tree to predict a trajectory label for profiling. Finally, we explain the PM boosting when data is incomplete.

5.1 Data Insight

After preprocessing, we count the trajectory amount distribution and the safe, risky and dangerous trajectory ratio with different trajectory types (i.e., different

Table 2. Cost matrix C .

	Predict SC	Predict RC	Predict DC
Actual SC	0	$C(SC, RC)$	$C(SC, DC)$
Actual RC	$C(RC, SC)$	0	$C(RC, DC)$
Actual DC	$C(DC, SC)$	$C(DC, RC)$	0

crash alarm counts per trajectory). As shown in Figs. 2 and 3, when crash alarm counts increase, the amount of the corresponding trajectory type decreases and the dangerous ratio increases. Interestingly, we find that the trajectories in the zero-crash-alarm-count type are all Safe Class. Furthermore, if a trajectory has more than one crash alarm, it can only be Risky or Dangerous Class. The reason may be that during the trajectory period, zero crash alarm means that the driver is driving smoothly without any risk or danger, leading to Safe Class. While, the generated crash alarm indicates the driver's aggressive driving, which results in a high probability of having crash accidents, lying in Risky or Dangerous Class. Thus, based on these observations, we develop an insight that crash alarm can be a critical factor to predict a trajectory's label, so that it can be utilized for profiling trajectories in real-time.

5.2 Decision-Stump-Based Model

We first profile trajectories with only focusing on crash alarms (i.e., crash alarm counts per trajectory feature cac). To this end, we develop *Decision-stump-based* model to predict a trajectory S 's label. Detailedly, we set two thresholds θ_1, θ_2 to generate a trajectory S 's predicted label \hat{y}_S as: $\hat{y}_S = \{SC : \text{if } cac \leq \theta_1; RC : \text{if } \theta_1 < cac \leq \theta_2; DC : \text{Otherwise}\}$.

To learn the parameters, we minimize a cost-sensitive objective function $C(\theta_1, \theta_2)$ with predicted label \hat{y}_S and actual label y_S (ground truth label) by:

$$C(\theta_1, \theta_2) = \sum_{S \in \text{all S}} 1(y_S \neq \hat{y}_S) \cdot C(y_S, \hat{y}_S), \quad (1)$$

where $C(i, j)$ is a cost matrix C (designed in Table 2). It means the cost that class i is mislabeled as class j and $i, j \in \{SC, RC, DC\}$. The value of the cost matrix is discussed in Sect. 8.6.

5.3 Decision-Tree-Based Model

Besides crash alarms, we also consider other trajectory indicator features (i.e., a trajectory’s running time and distance) for profiling. Due to multiple features, we utilize decision tree rather than decision stump. The *Classification and Regression Tree* with *Gini index* is selected [4]. To achieve the cost-sensitive setting, we do not prune the decision tree with the max depth [9].

5.4 PM Boosting

Considering the real-world scenarios, the collected data sometimes is incomplete, e.g., labeled data missing in transmission or non-access to the private insurance claim data. Under such condition of lacking ground truth label, we use PM’s predicted label instead as a pseudo label to boost the training of BM and PM.

6 Driver Behavior Model (BM)

6.1 Problem Formulation

In this part, we start to model each driver’s driving behaviors from trajectories. This problem aims to predict the probability (P_S^k) of a trajectory lying in safe, risky or dangerous class given a trajectory S ’s driver behavior feature π_S as:

$$P_S^k = P(y_S = k | \pi_S), k \in \{SC, RC, DC\}. \quad (2)$$

This formulates the problem to a typical multi-class classification problem. Then, we employ the popular Gradient Boost Machine (GBM) with tree classifier [7] as the multi-class classifier (Open for other classifiers to plug in). However, different from traditional GBM, we include the cost-sensitive setting for the practical applications.

6.2 Cost-Sensitive Setting Design

Given the whole trajectory set $S_{all} = \{S_1, S_2, \dots, S_N\}$, we first build $N * 3$ basic regression tree classifiers. Each N classifiers classify Safe/Risky/Dangerous Class respectively through *One-vs-All* strategy and output J_S^k to denote the score of trajectory S belonging to class k . Then, with *Softmax Function*, we have trajectory S ’s risk probability $P_S^k = e^{J_S^k} / (e^{J_S^{SC}} + e^{J_S^{RC}} + e^{J_S^{DC}})$, $k \in \{SC, RC, DC\}$.

Importantly, during the process, to learn the parameters and achieve a cost-sensitive purpose, we design and minimize the following objective function,

$$\Psi = \lambda - \sum_{S \in S_{all}} \sum_{k \in \{SC, RC, DC\}} l_S^k w_k \ln P_S^k, \quad (3)$$

where λ is the regularized parameter for all tree classifiers, l_S^k is a binary value for selecting which P_S^k to compute. Detailedly, if ground truth label $y_S = k$, $l_S^k = 1$. Otherwise, $l_S^k = 0$. w_k is class k 's weight for the cost-sensitive setting, achieved by multiplying different w_k values (different priorities) with the corresponding class k 's cross entropy. By default, we set weights by ratio as $w_{SC} : w_{RC} : w_{DC} = C(SC, RC) : C(RC, SC) : C(DC, SC)$ from cost matrix C .

Then, for the iterative gradient tree boosting processing [7], the first and second order approximations (i.e., gradient $grad_s$ and Hessian matrix $hess_s$) are used to quickly optimize Ψ :

$$\begin{aligned} grad_s &= \sum_{k \in \{SC, RC, DC\}} \frac{\partial \Psi}{\partial J_S^k} = \sum_{k \in \{SC, RC, DC\}} (P_S^k - l_S^k) w_k, \\ hess_s &= \sum_{k \in \{SC, RC, DC\}} \frac{\partial^2 \Psi}{\partial J_S^{k^2}} = \sum_{k \in \{SC, RC, DC\}} 2(1 - P_S^k) P_S^k w_k. \end{aligned} \quad (4)$$

7 Risk Evaluation Model (EM)

In this part, we first evaluate drivers from two perspectives: *Mobility-aware* from trajectories and *Demographic-aware* from driving habits (driver level). Then, we comprehensively consider the two evaluation scores and deploy the percentile ranking for the driver assessment.

7.1 Mobility-Aware Evaluation

For a trajectory S , after BM processing, we have the probability $P_S^{SC}, P_S^{RC}, P_S^{DC}$. Then, we compute a trajectory S 's risk score, $risk_S$ by:

$$risk_S = P_S^{SC} d_{SC} + P_S^{RC} d_{RC} + P_S^{DC} d_{DC}, \quad (5)$$

where $d_{SC/RC/DC}$ is the *risk level* of probability $P_S^{SC/RC/DC}$. By default for the cost-sensitive goal, we set $d_{SC} : d_{RC} : d_{DC} = C(SC, RC) : C(RC, SC) : C(DC, SC)$ by ratio. Next, we generate driver u 's m -th week's risk score $risk_u^m$ with this week's whole trajectory set ($\{S_1, S_2, \dots, S_N\}$) by:

$$risk_u^m = \frac{1}{N} \sum_{S \in \{S_1, S_2, \dots, S_N\}} risk_S. \quad (6)$$

Thus, with driver u 's M -week OBD data, \mathbf{r}_u ($\mathbf{r}_u = [risk_u^1, risk_u^2, \dots, risk_u^M]$) denotes the risk score sequence. By generating and plotting the whole drivers' risk score sequences in Fig. 4, we find three typical time-varying patterns over time (i.e., increasing/stable/decreasing). Therefore, when rating drivers, it is

Table 3. Demographic-aware variable (habit) description. (Set by domain experts)

Group	Variable (per month)	Variable description
Time of day	Daytime hours ratio	Fraction between 8 a.m. to 8 p.m.
	Nighttime hours ratio	Fraction between 8 p.m. to 8 a.m.
Day of week	Weekday hours ratio	Fraction between Monday to Friday
	Weekend hours ratio	Fraction between Saturday to Sunday
Road type	Urban roads hours ratio	Fraction on urban roads
	Highway hours ratio	Fraction on highways
	Extra-urban hours ratio	Fraction on extra-urbans (countryside)
Mileage	Mileage per month	Overall exposure 30-day average

necessary to pay more attention to the present than the past. Then, we employ a *Linear Weight Vector* \mathbf{w} ($|\mathbf{w}| = M, w_i = i$) to compute driver u 's time-varying Mobility-aware evaluation score $Eval_u^{Mob}$ by: (Note that, concerning different time-varying patterns, open for other *weight vectors* to plug in)

$$Eval_u^{Mob} = \frac{1}{|\mathbf{w}|} \mathbf{w}^T \mathbf{r}_u. \quad (7)$$

7.2 Demographic-Aware Evaluation

We can also evaluate drivers by driving habits like the nighttime/daytime driving hours fraction per month (More driving habits in Table 3). On the other hand, viewing drivers' past τ -month trajectory data, according to domain experts ($\tau = 6$ for half of a year), there are three types of drivers: **Accident-Involved (AI)** (having more than two dangerous trajectories), **Accident-Related (AR)** (having less than one dangerous trajectory but more than fifteen risky trajectories), **Accident-Free (AF)** (the remaining). Based on these, our Demographic-aware evaluation problem aims to predict driver u 's probability ($P_u^k, k \in \{AI, AR, AF\}$) of lying in AI, AR and AF by utilizing the driving habit variables.

Similar to BM, this problem also leads to a cost-sensitive multi-class classification task. After employing similar cost-sensitive solutions in BM (Due to space limit, we omit to present again), for driver u , we generate the probability of $P_u^{AI}, P_u^{AR}, P_u^{AF}$. Then, we have Demographic-aware evaluation score $Eval_u^{Dem}$ as:

$$Eval_u^{Dem} = P_u^{AI} m_{AI} + P_u^{AR} m_{AR} + P_u^{AF} m_{AF}, \quad (8)$$

where $m_{AI/AR/AF}$ is the *risk level* of probability $P_u^{AI/AR/AF}$ and we set $m_{AF} : m_{AR} : m_{AI} = C(SC, RC) : C(RC, SC) : C(DC, SC)$, similar to Eq. 5.

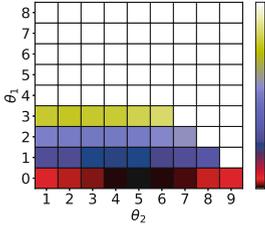


Fig. 5. PM grid search cost.

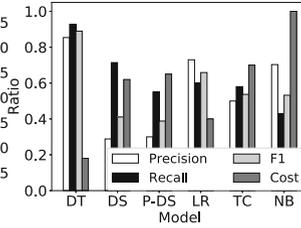


Fig. 6. PM comparison.

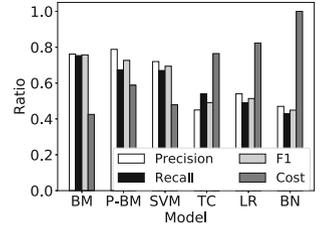


Fig. 7. BM comparison.

7.3 Driver Evaluation

Finally, for driver u , we sum the two scores as a driver evaluation score $Eval_u$:

$$Eval_u = \alpha Eval_u^{Mob} + \beta Eval_u^{Dem}, \quad (9)$$

where $Eval_u^{Mob}$, $Eval_u^{Dem}$ are normalized due to their different value ranges and α, β are weight parameters to indicate the significance/priority of corresponding evaluation score. The higher value leads to the higher importance. Generally, we set $\alpha = \beta = \frac{1}{2}$ to denote the equal significance in evaluating a driver (Flexible for other preferences to plug in for different user requirements). Finally, $Eval_u$ suggests a comprehensive risk score of driver u . The higher the score is, the more risky the driver is.

After generating all drivers' evaluation scores, we deploy the percentile ranking⁴ to assess the drivers. According to domain experts' knowledge, 20% drivers can cause 80% crash accidents. Then, we set the percentile 80% drivers as the Dangerous Drivers. Among the rest drivers, usually 20% drivers are risky. Motivated by this, we set 80% percentile as the Risky Drivers and the final remaining as Safe Drivers (Available for other percentiles to plug in). Finally, by this setting, we can obtain two evaluation scores as thresholds to quickly assess a driver as safe, risky or dangerous.

8 Experiment

8.1 Setting and Dataset

During the experiments, we generated the equal number of trajectories in each class by resampling to balance the data. Besides, the 10-fold cross validation was conducted to present robust results. The real-world dataset collected drivers' OBD data from *August 22, 2016* to *March 27, 2017* for nearly 30 weeks, provided by a major OBD product company in China. After preprocessing, we have basic data statistics: 198 drivers, 98,218 trajectories (*Safe* 91,687, *Risky* 5,853 and *Dangerous* 678), average trajectory time of 25.22 min and distance of 12.95 km.

⁴ https://en.wikipedia.org/wiki/Percentile_rank.

Table 4. BM feature performance study.

Column ID (COL. ID)		Experiment setting							
		1	2	3	4	5	6	7	8
Feature*	Speed+	✓	✓	✓	✓	✓	✓	✓	✓
	Acceleration+		✓			✓	✓		✓
	Engine Speed+			✓		✓		✓	✓
	Vehicle Angular Velocity+				✓		✓	✓	✓
Metrics	Precision (%)	49	61	53	60	62	64	62	69
	Recall (%)	48	58	53	57	61	62	62	70
	F1 score (%)	49	57	53	56	61	61	62	69
	Cost ratio (%)	100	91	69	99	73	86	75	63

*Feature+ means the physical driving state feature plus its corresponding semantic driving state features (E.g., Speed+ means Speed plus Vehicle Speeding in Table 1).

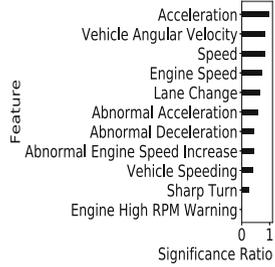


Fig. 8. BM feature significance.

8.2 Trajectory Profiling Model (PM) Evaluation

In this part, we first employ the grid search to find the optimal θ_1^*, θ_2^* in Decision-Stump-based model (DS) with the minimal cost-sensitive objective in Sect. 5.2. Then, we utilize the optimal DS and Decision-Tree-based model (DT) to profile trajectories, by predicting a trajectory’s label through trajectory indicator features (i.e., crash alarm counts, distance and running time information).

After the grid search, as shown in Fig. 5 (Where cost is divided by its maximal cost as a ratio), we have the optimal thresholds of $\theta_1^* = 0, \theta_2^* = 5$. Explicitly, $\theta_1^* = 0$ means if a trajectory has no crash alarm records, it is Safe. It is consistent with the trajectory data statistics (Sect. 5.1). Besides, $\theta_2^* = 5$ denotes that if OBD generates crash alarm records during a trajectory, the crash alarm count of five is used to quickly judge whether the trajectory is Risky or Dangerous, for sending timely messages to drivers and reminding them of danger.

Figure 6 shows results for profiling trajectories in metrics of Precision, Recall, F1 score, as well as the cost (i.e., the misclassification cost in Eq. 1). We see that DT gets Precision, Recall and F1 score, close to 0.9, with the lowest cost. It outperforms the compared methods of multi-class Logistic Regression (LR) [10], Trajectory Clustering (TC) [11,12] and robust cost-sensitive Naive Bayes (NB) [9]. Furthermore, we also compare our two profiling models of DT and DS. As shown, DT is much better than DS. The reason may be that DT uses more rules (i.e., the higher depth in tree) and more features to sufficiently judge a trajectory’s label, even in complex conditions. However, DS can quickly judge by only using one feature. According to the positive feedbacks from domain experts, DS is much easier to be implemented in current OBD device and has a great potential in real-time driving alarming systems with the easiest portability. Therefore, both DT and DS have their own advantages and suitable application scenarios. When users prefer the better performance, they may choose DT. For example, in PM boosting, by default, the predicted label is generated from DT.

Otherwise, if they want to quickly obtain results, DS is a good choice. ***The impact of the cost-sensitive setting.*** By examining the results of DS and Pure Decision-Stump-based model (P-DS), which removes the cost-sensitive setting when learning parameters, we find DS performs better with higher Recall and F1 score than P-DS. It validates the effectiveness of the cost-sensitive setting in the real-world application scenario to retrieve more risky/dangerous trajectories for higher Recall and F1 score.

8.3 Driver Behavior Model (BM) Evaluation

To evaluate BM, alternatively, we perform the formulated classification task (Sect. 6.1). We set the following compared methods: Logistic Regression (LR), Trajectory Clustering (TC), Support Vector Machine (SVM) [17], Bayesian Network (BN) [22] and Pure-BM (P-BM) where we remove BM's cost-sensitive setting.

The experiment result is shown in Fig. 7. It is observed that, BM has high Precision, Recall and F1 score close to 80% and the lowest cost. It beats SVM, LR, BN and TC in all metrics. This means that in our application, BM is more suitable to process the high-dimension trajectory feature data for the multi-class classification task. But, as mentioned before in Sect. 6.1, BM is open for other classifiers to plug in. ***The impact of the cost-sensitive setting.*** Compared to P-BM, BM outperforms in Recall and F1 score about 8% improvement with 27.70% lower cost. The reason may be that BM's cost-sensitive setting guides BM to give more priority to more risky classes like Risky and Dangerous class. This leads to the final prediction of Risky and Dangerous class more accurate than Safe class with higher Recall/F1 and lower cost. ***The impact of feature (feature performance study)***. We also test the effects of the driving behavior features under various feature combinations. The result is shown in Table 4.

It is observed that: (1) Compared to traditional GPS-related Speed+ features, adding any unique OBD-related feature like Acceleration+, Engine Speed+ and Vehicle Angular Velocity+ can improve the performance with higher Precision, Recall, F1 score and lower cost (See COL. 1 vs. COL. 2–8). It is natural to understand it because with more driving features, we can get larger feature space to describe the trajectory, leading to better predictions. Then, it effectively suggests the advantage of OBD for its involvement of more fine-grained driving features. (2) Seeing two comparing pairs: (i) (COL. 1 vs. COL. 2) vs. (COL. 1 vs. COL. 4) and (ii) (COL. 3 vs. COL. 5) vs. (COL. 3 vs. COL. 7), Acceleration+ and Vehicle Angular Velocity+ seem to have similar improving effects. The reason may be that both Acceleration+ and Vehicle Angular Velocity+ directly manifest the driving actions. Then, the improvement of adding either one feature is almost the same. However, adding Engine Speed+ leads to lower improvement in Precision, Recall, F1 score (see COL. 1 vs. COL. 2–4) probably because Engine Speed+ indicates the putting state of the drivers' feet on the oil pedal. It may not directly reflect the driving behaviors in the road like Acceleration+ and Vehicle Angular Velocity+. But viewing COL. 8, all the features together lead to the best performance. (3) Furthermore, in experiment setting 8, we investigate the significance of the whole features by measuring how many times a feature

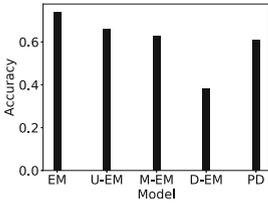


Fig. 9. EM comparison.

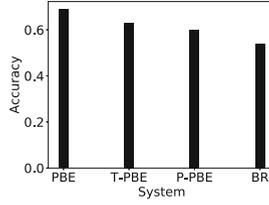


Fig. 10. System comparison.

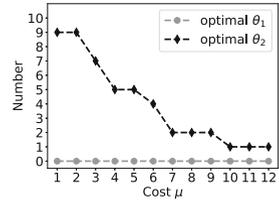


Fig. 11. Effect of parameter.

is used in BM’s tree classifiers to split a subtree. As shown in Fig. 8, the top two features of Acceleration and Vehicle Angular Velocity validate our previous analysis of their similar high improvements in performance compared to Speed.

8.4 Risk Evaluation Model (EM) Evaluation

In this subsection, we evaluate EM under the following enterprise scenario: with the whole driver set and the first 20-week data, after EM, we get a dangerous driver set with 80% percentile. Then, in the following 10-week dataset, we check whether these dangerous drivers have crash accident records or not. If Yes, our predictions are accurate and vice versa. We choose *Accuracy* as the metric. The compared methods are: (1) Pay as you Drive model (PD) is a state-of-the-art technique to evaluate drivers by conducting the vehicle classification task [19]. The generated classification probability is used for the evaluation and the parameters are carefully tuned to give the best performance. (2) M-EM/D-EM only contains the Mobility-aware/Demographic-aware evaluation score. (3) Unified-weight EM (U-EM) ignores the time-dependent pattern (Fig. 4) and utilizes the unified weight week vector to evaluate drivers (i.e., $w_i = 1$ in Eq. 7).

As shown in Fig. 9, EM has the highest accuracy. EM outperforms PD with 21% improvement. It may be caused from PD’s vehicle classification part, which fails to consider the mobility-aware perspective. ***The impact of the time-dependent pattern.*** By examining the performance of EM and U-EM, we find that EM is more effective and improves the accuracy by about 13%. As aforementioned, we should give more priority to the latest driver behaviors rather than the very early, due to the changes of driving proficiency over time. ***The impact of the Mobility-aware and Demographic-aware evaluation.*** Compared with M-EM and D-EM, the accuracy of EM is about 47% better on average. It suggests that multiple perspectives lead to a more comprehensive evaluation for better performance. If only focusing on Mobility-aware or Demographic-aware information, we lose something for the evaluation. By viewing M-EM and D-EM only, one interesting finding is that M-EM is better. The reason is that M-EM evaluates drivers by the fine-grained driving behaviors from the trajectory perspective. It tells the dynamic mobility pattern so that it better describes and distinguishes drivers for the risk assessment. But, D-EM’s rating is mainly based on general, less-distinctive and static variables like the per-month traveled mileage.

8.5 PBE System Evaluation

We investigate PBE system’s performance by the same task described in EM evaluation in Sect. 8.4. To fully test the boosting effect, we consider the extreme condition with no ground truth labels, by setting PM’s predicted labels as the whole trajectories’ ground truth labels. The compared methods are: (1) Two-class PBE (T-PBE), which considers the two-class setting rather than PBE’s three-class setting. Specially, it regards Dangerous class as one class while Risky and Safe class as another class. (2) Pure-PBE (P-PBE), one removes the cost-sensitive setting in the whole system. (3) Behavior-centric Risk-level model (BR) is a state-of-the-art method to evaluate drivers [3]. It is incorporated in an insurance pricing model to rate drivers’ risk for evaluation.

The result is shown in Fig. 10. It can be found that: (1) PBE outperforms T-PBE with 9% improvement. The reason may be that PBE additionally utilizes the Risky Class trajectory to develop more semantic-rich descriptions of a trajectory and a driver (i.e., more probabilities to describe). This result suggests the advantage of the multi-class fine-grained analysis. Furthermore, PBE system is open for other multi-class settings to plug in not limited to current three classes. (2) By viewing the result of PBE and P-PBE, we observe that PBE has higher accuracy than P-PBE by 15%. It suggests that the cost-sensitive setting is effective in the whole system. The reason is aforementioned that through the cost-sensitive guidance in the system, Risky and Dangerous Classes get more priority to retrieve more risky trajectories/drivers in the real-world enterprise scenario like auto insurance. (3) Examining PBE and BR’s performances, we note that PBE beats BR by 28%. Different from PBE, BR’s evaluation from classification fails to consider trajectories’ fine-grained driving behaviors, which leads to the lower performance. (4) Comparing the results of PBE’s predicted label and the ground truth label (in Sect. 8.4), current PBE is only slightly worse by 7%. Such slight difference is acceptable in the real-world applications when the ground truth label is hard to assess and data is incomplete. It justifies the effectiveness of PM for boosting the training of BM and EM.

8.6 Parameter Tuning

In PBE, the major parameter is the cost matrix C in Table 2. Considering the trade-off between the cost-sensitive requirement and the scalable training, we set $C(SC, RC) : C(SC, DC) : C(RC, DC) : C(RC, SC) : C(DC, RC) : C(DC, SC) = 1 : 1 : 1 : \mu : \mu : \mu^2$. Through studying cost μ by Decision-Stump-based model’s cost-sensitive objective $C(\theta_1, \theta_2)$ (in Sect. 5.2), where C is used for the first time in PBE, we can examine the impact of C . As shown in Fig. 11, μ greatly influences θ_2^* without affecting θ_1^* . Specifically, μ increases with θ_2^* decreasing. When μ is too low ($\mu < 4$)/high ($\mu > 6$), the optimal θ_2^* is just around θ_2 ’s max/min value, leading to improper results. Thus, we select the middle value $\mu = 5$, resulting in a middle threshold value $\theta_2^* = 5$ in the experiment.

Table 5. The running time of PBE system.

Week #	1	5	10	15	20	25	30
Time (in second)	7	43	88	138	189	227	285

8.7 Efficiency Study

We report the running time of PBE when the week number increases (i.e., when more data is collected). As shown in Table 5, the time cost increases with more data. The reason is that with more data, more trajectories are generated and more tree classifiers are built, which result in more running time.

9 Conclusion

In this paper, we proposed PBE system, including PM, BM and EM, to assess the driver behaviors. PM utilizes the insight from the collected data for real-time alarming. BM assesses the driver behavior risk by fine-grained analyzing the trajectory data. EM evaluates drivers from multiple perspectives and gives comprehensive scores to reflect different risky scores. PBE is evaluated via extensive experiments and outperforms the traditional systems by at least 21%. In the future, we will consider more spatial factors like location/road type for analysis.

Acknowledgements. This research was supported by Shenzhen Peacock Talent Grant 827-000175, Guangdong Pre-national Project 2014GKXM054, the University of Macau Start-up Research Grant (SRG2015-00050-FST) and Research & Development Grant for Chair Professor (CPG2015-00017-FST), and Natural Science Foundation of China: 61572488 and 61673241.

References

1. Al-Sultan, S., Al-Bayatti, A.H., Zedan, H.: Context-aware driver behavior detection system in intelligent transportation systems. *VT* **62**(9), 4264–4275 (2013)
2. Bashir, F.I., Khokhar, A.A., Schonfeld, D.: Object trajectory-based activity classification and recognition using hidden Markov models. *TIP* **16**(7), 1912–1919 (2007)
3. Bian, Y., Yang, C., Zhao, J.L., Liang, L.: Good drivers pay less: a study of usage-based vehicle insurance models. *Transp. Res. Part A: Policy Pract.* **107**, 20–34 (2018)
4. Breiman, L.: *Classification and Regression Trees*. Routledge, New York (2017)
5. Castignani, G., Derrmann, T., Frank, R., Engel, T.: Driver behavior profiling using smartphones: a low-cost platform for driver monitoring. *ITSM* **7**(1), 91–102 (2015)
6. Chen, S.H., Pan, J.S., Lu, K.: Driving behavior analysis based on vehicle OBD information and AdaBoost algorithms. In: *IMECS*, vol. 1, pp. 18–20 (2015)
7. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: *SIGKDD*, pp. 785–794. ACM (2016)

8. Daptardar, S., Lakshminarayanan, V., Reddy, S., Nair, S., Sahoo, S., Sinha, P.: Hidden Markov model based driving event detection and driver profiling from mobile inertial sensor data. In: SENSORS, pp. 1–4. IEEE (2015)
9. Elkan, C.: The foundations of cost-sensitive learning. In: IJCAI, pp. 973–978. Lawrence Erlbaum Associates Ltd. (2001)
10. Friedman, J., Hastie, T., Tibshirani, R.: The Elements of Statistical Learning. Springer Series in Statistics, vol. 1. Springer, New York (2001). <https://doi.org/10.1007/978-0-387-21606-5>
11. Gaffney, S., Smyth, P.: Trajectory clustering with mixtures of regression models. In: SIGKDD, pp. 63–72. ACM (1999)
12. Hartigan, J.A., Wong, M.A.: Algorithm as 136: a k-means clustering algorithm. *J. R. Stat. Society. Ser. C (Appl. Stat.)* **28**(1), 100–108 (1979)
13. Hendricks, D.L., Fell, J.C., Freedman, M.: The relative frequency of unsafe driving acts in serious traffic crashes. In: AAAM (2000)
14. Hong, J.H., Margines, B., Dey, A.K.: A smartphone-based sensing platform to model aggressive driving behaviors. In: SIGCHI, pp. 4047–4056. ACM (2014)
15. Kaplan, S., Guvensan, M.A., Yavuz, A.G., Karalurt, Y.: Driver behavior analysis for safe driving: a survey. *ITS* **16**(6), 3017–3032 (2015)
16. Kumar, M.K., Prasad, V.K.: Driver behavior analysis and prediction models: a survey. *IJCSIT* **6**(4), 3328–3333 (2015)
17. Nilsson, R., Peña, J.M., Björkegren, J., Tegnér, J.: Evaluating feature selection for SVMs in high dimensions. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) ECML 2006. LNCS (LNAI), vol. 4212, pp. 719–726. Springer, Heidelberg (2006). <https://doi.org/10.1007/11871842.72>
18. Osafune, T., Takahashi, T., Kiyama, N., Sobue, T., Yamaguchi, H., Higashino, T.: Analysis of accident risks from driving behaviors. *Int. J. ITS Res.* **15**(3), 192–202 (2017)
19. Paefgen, J., Staake, T., Fleisch, E.: Multivariate exposure modeling of accident risk: insights from pay-as-you-drive insurance data. *Transp. Res. Part A: Policy Pract.* **61**, 27–40 (2014)
20. Ruta, M., Scioscia, F., Gramegna, F., Di Sciascio, E.: A mobile knowledge-based system for on-board diagnostics and car driving assistance. In: UBIComm, pp. 91–96 (2010)
21. Tran, C., Doshi, A., Trivedi, M.M.: Modeling and prediction of driver behavior by foot gesture analysis. *CVIU* **116**(3), 435–445 (2012)
22. Tschiatsek, S., Reinprecht, P., Mücke, M., Pernkopf, F.: Bayesian network classifiers with reduced precision parameters. In: Flach, P.A., De Bie, T., Cristianini, N. (eds.) ECML PKDD 2012. LNCS (LNAI), vol. 7523, pp. 74–89. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33460-3_10
23. Van Ly, M., Martin, S., Trivedi, M.M.: Driver classification and driving style recognition using inertial sensors. In: IEEE Intelligent Vehicles Symposium IV, pp. 1040–1045. IEEE (2013)
24. Wei, L., Keogh, E.: Semi-supervised time series classification. In: SIGKDD, pp. 748–753. ACM (2006)
25. Xi, X., Keogh, E., Shelton, C., Wei, L., Ratanamahatana, C.A.: Fast time series classification using numerosity reduction. In: ICML, pp. 1033–1040. ACM (2006)
26. Xu, G., Liu, L., Song, Z.: Driver behavior analysis based on Bayesian network and multiple classifiers. In: ICIS, vol. 3, pp. 663–668. IEEE (2010)
27. Zheng, Y.: Trajectory data mining: an overview. *ACM TIST* **6**(3), 29 (2015)