Sentio: Driver-in-the-Loop Forward Collision Warning Using Multisample Reinforcement Learning

Salma Elmalaki
University of California, Los Angeles
selmalaki@ucla.edu

Huey-Ru Tsai
University of California, Los Angeles
tsai.debra@cs.ucla.edu

Mani Srivastava
University of California, Los Angeles
mbs@ucla.edu

ABSTRACT

Thanks to the adoption of more sensors in the automotive industry, context-aware Advanced Driver Assistance Systems (ADAS) become possible. On one side, a common thread in ADAS applications is to focus entirely on the context of the vehicle and its surrounding vehicles leaving the human (driver) context out of consideration. On the other side, and due to the increasing sensing capabilities in mobile phones and wearable technologies, monitoring complex human context becomes feasible which paves the way to develop driver-in-the-loop context-aware ADAS that provide personalized driving experience. In this paper, we propose Sentio[1]; a Reinforcement Learning based algorithm to enhance the Forward Collision Warning (FCW) system leading to Driver-in-the-Loop FCW system. Since the human driving preference is unknown a priori, varies between different drivers, and moreover, varies across time for the same driver, the proposed Sentio algorithm needs to take into account all these variabilities which are not handled by the standard reinforcement learning algorithms. We verified the proposed algorithm against several human drivers. Our evaluation, across distracted human drivers, shows a significant enhancement in driver experience—compared to standard FCW systems—reflected by an increase in the driver safety by 94.28%, an improvement in the driving experience by 20.97%, a decrease in the false negatives from 55.90% down to 3.26%, while adding less than 130 ms runtime execution overhead.

CCS CONCEPTS

• **Human-centered computing → Human computer interaction (HCI)**; Ubiquitous computing; • **Computing methodologies → Reinforcement learning**;

KEYWORDS

Reinforcement Learning, Human-in-the-Loop, Personalized ADAS, Human Vehicular Interaction, Autonomous Systems.

ACM Reference Format:

Salma Elmalaki, Huey-Ru Tsai, and Mani Srivastava. 2018. Sentio: Driver-in-the-Loop Forward Collision Warning Using Multisample Reinforcement Learning. In *The 16th ACM Conference on Embedded Networked Sensor Systems (SenSys '18), November 4–7, 2018, Shenzhen, China.* https://doi.org/10.1145/3274783.3274843

[1]Sentio is a Latin word which means "be aware".

SenSys '18, November 4–7, 2018, Shenzhen, China

ACM ISBN 978-1-4503-5952-8/18/11…$15.00
https://doi.org/10.1145/3274783.3274843

1 INTRODUCTION

The dramatic increase in sensing capabilities has opened the door for a multitude of new applications in the context of Advanced Driver Assistance Systems (ADAS). ADAS are developed to adapt the vehicle systems according to the vehicle and the human context in an attempt to enhance both the vehicle safety and the driving experience. In these context-aware systems, sensor information collected from various sensors are fused together to infer the current context (or state) of the system and adapt the system functionality to match the system's and the user's state. Lane departure warning, front collision warning, and driver drowsiness detection are just examples to name a few [5, 18, 19, 46].

While ADAS is one of the fastest-growing segments in automotive electronics [40], most ADAS systems—except for driver drowsiness detection—focus on adapting to the state of the environment surrounding the vehicle without taking the driver and passenger state into consideration. Automatic lighting, adaptive cruise control, automatic braking, lane departure warning, and front collision warning are all among the most utilized ADAS systems that fall into this category. However, due to the increase in sensing capabilities and sensing frameworks, mobile systems and wearables have shown substantial success in inferring different human contexts [11, 21, 26, 35]. Given the state of the environment surrounding the vehicle (inferred using the vehicle sensors) along with the state of the driver (inferred using mobile/wearables systems), we ask the question of how to design algorithms that can use this information and provide a personalized driving experience.

In this paper, we focus on the Forward Collision Warning (FCW) system. Standard FCW system uses radars to detect vehicles or obstacles in front of the car. The system measures the time-to-crash based on the distance and the relative velocity of the front object and, if the time-to-crash is below a certain threshold (signaling a possible risk of collision), it sounds an alarm and displays a visual alert, prompting the driver to apply the brakes.

Unfortunately, to design FCW systems, automotive makers fix a certain threshold based on multiple experiments capturing the average human behavior and response. However, as studied by the US Department of Motor Vehicles, emotions (both negative and positive) can cause distraction and delay driver's response by a few seconds [9]. Moreover, arguing with a passenger could be even more distracting than talking on a cellphone while driving [17]. Significant findings showed that contentious conversations to be more emotionally taxing, and the ability to drive was significantly compromised [17]. These observations motivate the need to design a driver-in-the-loop FCW system which adapts to individual drivers' responses based on their cognition context. We argue in this paper that by carefully designing the FCW system, a driver-in-the-loop

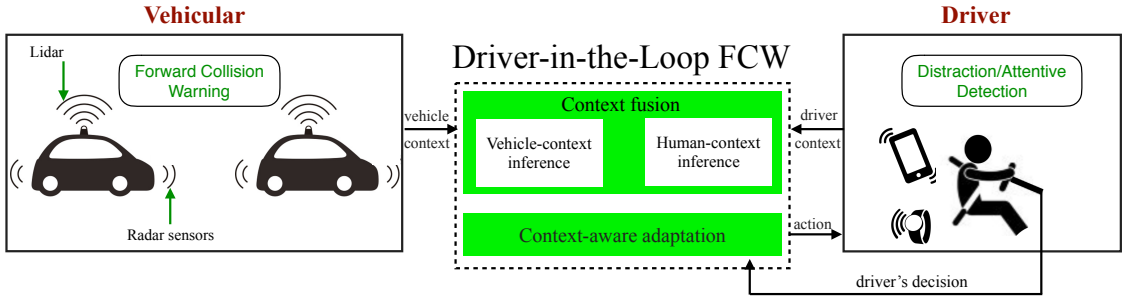

Figure 1: Sentio architecture. The context of the driver (attention) is inferred using the data collected by phone and wearables. The context of the vehicle (vehicle speed) and vehicle environment (distance to front collision) is collected by the vehicle sensors. The adaptation actions are then personalized based on the driver's decision.

FCW could adapt to the driver state and trigger the visual alarm early enough to accommodate the latency in the driver response.

Aided by the current developments in building machine learning based agents, recent literature in designing and building context-aware systems focused on using labeled data to train a machine learning classifier that can "infer" the human state. A smart FCW system not only needs to "infer" or "learn" the human preference but also continuously "adapts" and "takes actions" in the form of visual warnings. This feedback property opens the door to design a "Reinforcement Learning (RL)" based agent to build the proposed driver-in-the-loop FCW system. Unfortunately, applying standard RL algorithms like Q-learning, in the context of FCW, faces several challenges. In this paper, we identify these challenges and propose a modified Q-learning algorithm, named the "multisample Q-learning", to address these challenges. The proposed algorithm is then used to build a driver-in-the-loop FCW that adapts its behavior, at runtime, to the attention level, preference, and response time of human drivers. This online adaptation leads to a personalized driving experience as reported by several human drivers.

1.1 Related Work

1.1.1 ***Personalized Forward Collision Warning.*** Learning the driver behavior model is an essential step in personalized FCW. However, collecting sufficient data has been the primary setback of these models [25]. Using statistical modeling to reduce the required data set has been reported in [25]. Unfortunately, learning fixed parameters for a driver model, using offline data, can increase the false alarms rate if the driver behavior changes at runtime, which was not addressed by the proposed statistical modeling framework [25]. Real-time parameter identification of driver behavior is proposed in [43]. By assuming a fixed parametric model for the driver behavior, the work reported in [43] uses online data to update this model and adapts the FCW threshold according to the learned parametric model. This adaptation reduces the false warning rate if the driver behavior changes. Other tunings for the warning threshold have been addressed, by using GPS data [28], drivers' expected response decelerations (ERDs) [44], and drivers' longitudinal braking behavior using GMM [37].

Our work differs from previously reported work in the following sense. First, unlike [25], we avoid designing offline thresholds and focus on the problem of continuous adaptation to the driver behavior. Unlike the other work on online adaptation [25, 37, 44], we do not restrict our algorithm to particular parametric models. Instead, we use an entirely data-driven approach to adapt to the human behavior. Second, related work focuses on using only the information collected by the vehicle sensors (e.g., Radars) and ignore the fact that a rich set of sensory data obtained from mobile/wearable systems that can help in identifying the state of the driver.

1.1.2 ***Human-in-the-Loop Context-Aware Automotive Systems.*** The role of humans in automotive systems can be divided into three domains [30], (1) inside the vehicle cabin, (2) around the vehicle, and (3) inside the surrounding cars. While there has been a vast amount of work targeting each domain, we focus in this section on those related to the first category. We classify the work in this domain into two subcategories namely (i) intent detection and (ii) state detection. The work reported by [19] falls in the first subcategory, intent detection, in which a hidden Markov model is used as a probabilistic model that captures the intent of the driver. The second category is concerned with the attention of the human agent. The work reported by [1, 39] falls in this category. Other reported results in this category include studying the driver behavior to measure his level of fatigue through analyzing images of the driver [8, 31, 47], driver distraction [14, 31], stress driving [13, 33], and inattentiveness/distraction detection [1, 39]. Similarly, the work reported in [34] actively monitors the driver to detect his cellphone use. Identifying drunk drivers and calling the police for help has also been studied in [7]. Our work builds on top of recent successes reported in the second subcategory. We aim to design algorithms that make use of the inferred human state to design a personalized FCW that adapts to the driver state.

1.2 Paper Contribution

Sentio aims at putting the human state and preference into the loop of computation, more specifically, we make the following contributions:

- Designing a modified Q-learning named the "multisample Q-learning" algorithm that addresses inherent challenges of using the standard Q-learning algorithm in the context of FCW.
- We use the proposed multisample Q-learning to develop an RL agent for Sentio; a driver-in-the-loop FCW system. This RL agent continuously monitors the state of the driver and the environment surrounding the vehicle to release the FCW early enough to match the driver attention level and preference (regarding the relative distance between the driver car and other cars).
- We implemented a proof-of-concept of the proposed Sentio system that demonstrates the feasibility of our algorithm. We evaluated Sentio on human drivers using a virtualized simulated environment.

2 SENTIO SYSTEM ARCHITECTURE

A conceptual overview of the proposed driver-in-the-loop FCW (Sentio) architecture is shown in Figure 1. The proposed Sentio is divided into three main modules as follows:

2.1 Human Context-Inference

The first step towards a driver-in-the-loop FCW is to infer the current state of the human driver. While recently reported work in the literature showed how to understand complex human states [15, 22, 45], the objective of this paper is *not* to develop a new human context-inference engine. Instead, and to facilitate verifying the central concepts of the proposed Sentio, we will focus on a single facet of inferring human distraction. In particular, we will make use of recent studies that show how being engaged in a conversation can lead to a high distraction as well [17, 23]. Therefore, the current implementation of the human context-inference module outputs a binary signal (attentive/distracted) based on the available audio streams collected by the driver's phone. Since the design of the inference engine is *not* the main focus of the paper, we postpone the implementation details of this module until the evaluation section (Section 6). Indeed, the proposed Sentio can be generalized directly to any other human context-inference engine that may use more complex data streams (e.g., in-vehicle camera streams, ECG, heart rate) collected from various phone/wearables sensors.

2.2 Vehicle and Environment Context-Inference

The objective of this module is to utilize the car sensors to infer the context of the car and its surrounding environment. In particular, we are interested in two sensory information: (i) relative distance and (ii) relative velocity which can be directly inferred from the information collected by the car front-facing radars and LiDARs along with the car speedometer. Raw information from these two sensors can be captured by scanning the internal CAN bus of the car through the standard OBD-based CAN scanners.

The final output of this module is a quantized version of the relative distance. All relative distances below 7m are considered as one "dangerous" state[2] (very near to collision), and all relative distances above 14m are considered as one "very far" state[3] in which the system does not need to react. Similarly, the relative velocity is quantized, where all negative relative velocities are considered as one state, signaling the case when the velocity of the driving car is smaller than the leading car and hence a collision will never occur.

2.3 Context-Aware Adaptation Engine

The context-aware adaptation engine is the main contribution of the paper. It is responsible for (i) using both the driver and the vehicle context to—implicitly—infer the likelihood of approaching a dangerous state based on the driver attention level, his response time, and his driving preference, and (ii) alerting the driver early enough to avoid a possible crash.

Learning the best timing that best suits the human state is subjective to the human interaction and response to this alert which vary from one human to another. This variation gives rise to two clear challenges:

- **Variation between individuals:** Every human driver has his preference regarding the relative distance he likes to keep between his car and the leading car [43]. When an alert is signaled, the human observes the relative distance and decides, based on his preference, whether to comply with the alert signal by decelerating the car or ignore it [43].
- **Variation within the same individual:** Human driver, in general, can change his preferences over time [43]. In other words, a driver can prefer a different relative distance over time. This variation comes from the fact that other outside daily factors change the driver behavior which can not be entirely comprehended. Similarly, when the human attention level varies, his response time in observing the relative distance and taking action changes as well [6].

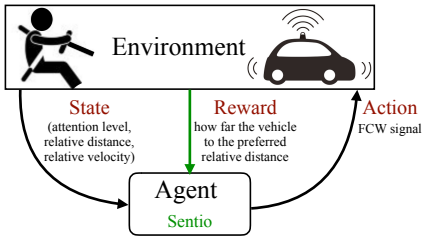

Figure 2: Modeling human-vehicular interaction using Reinforcement Learning where the environment is both the human and the vehicle and the agent is Sentio.

To address these challenges, the adaptation-engine continuously monitors the driver reactions to the issued FCW. If the driver ignores the warnings, our engine "learns" that the driver finds this warning false (earlier than the driver preference). If the driver reacts to the issued warning, our engine learns that this warning matches the driver preference. This continuous feedback loop of issuing warnings (actions) and monitoring the driver decision (response) fits naturally within the Reinforcement Learning (RL) paradigm.

In the RL problem, a software agent tries to learn the behavior of an environment by issuing actions and observing the change in the state of the environment with the purpose of maximizing a notion of a total reward. In the context of Sentio, as shown in Figure 2, the environment is both the human driver and the vehicle. Hence, the environment state consists of both the human state (attention level) and the vehicle state (relative distance and the relative velocity). The only action that is decided by the RL agent is when to issue the FCW while the reward of the system is how far the current relative distance is from the driver preferred relative distance. Therefore, to correctly design an RL algorithm we need to define its three main components namely (i) model for the environment, (ii) reward function, and (iii) learning algorithm. In the context of Sentio, and compared with the standard RL setup we face several challenges in defining these three components that we illustrate as follows:

CH1 Dynamic and time-varying rewards: The standard RL setup assumes the reward function to be time-invariant. That is, applying the same action to the same environment state shall lead to receiving the same reward. Unfortunately, in the context of Sentio, the reward received depends on the driver preference (the safe distance that he prefers to keep

[2]Using the 2 seconds rule [29] for safe driving with an average relative velocity of 15 kph (typical deceleration rate for a typical passenger car per second) between the driving car and the leading car.
[3]Maximum recommended safe distance using an average relative velocity of 15kph between the driving car and the leading car [29].

between the two cars) which varies over time and is unknown to the RL agent. This time-varying aspect precludes using techniques like Inverse Reinforcement Learning (IRL) which aims to construct the reward function from previously collected data.

CH2 Ignoring actions generated by the RL agent: In the standard RL setup, the actions produced by the RL agent are assumed to have a direct effect on the environment. In other words, it assumes that the environment always obeys the actions taken by the agent which implies that the rewards received reflect the actions decided by the RL agent. This assumption is violated in our setup since the human driver may or may not comply with the warnings triggered by the RL agent. That is, while the agent may decide to issue the FCW, the driver may choose to ignore the warning and accelerate the car. Hence, the reward that is received by the RL agent does not always reflect the RL agent actions but the decisions taken by the driver.

CH3 Reward time horizon: Finally, the standard online RL setup assumes that rewards received at each step (or execution) of the algorithm are due to the taken actions in this step. Again, such an assumption is violated in our setup since the human response (which is in the order of seconds) is not always instantaneous. Therefore, the environment may take several steps (or executions) until the effect of the action is observed and the corresponding reward is received. Moreover, the human response time is unknown, depends on his attention level, and varies across time [6]. While prior work addressed this challenge by adding an additional assumption that a delayed reward follows a Poisson distribution [2], in the case of Sentio, there is no evidence that the driver delay follows such distribution.

In the subsequent sections, we illustrate our solution to each of these challenges.

3 HUMAN DRIVER AS A MARKOV DECISION PROCESS

Solving the reinforcement learning problem starts by carefully modeling the environment—the human driver and the vehicle in our case—in a way that captures how the environment state changes in response to applied actions. Moreover, modeling the human driver has to take into account the two challenges mentioned earlier namely the variations between individuals and the variations within the same individual.

Accordingly, in Sentio, we model the change in the human and vehicle state as a Markov Decision Process (MDP) with unknown transition probabilities[4]. The states of the MDP are based on the context of both the driver (attention level) and the vehicle (relative distance and relative velocity), i.e., each state is a tuple $s = (a, \Delta d, \Delta v)$ where a denotes the human attention ($a = 0$ means the human is distracted and $a = 1$ otherwise), Δd denotes the relative distance,

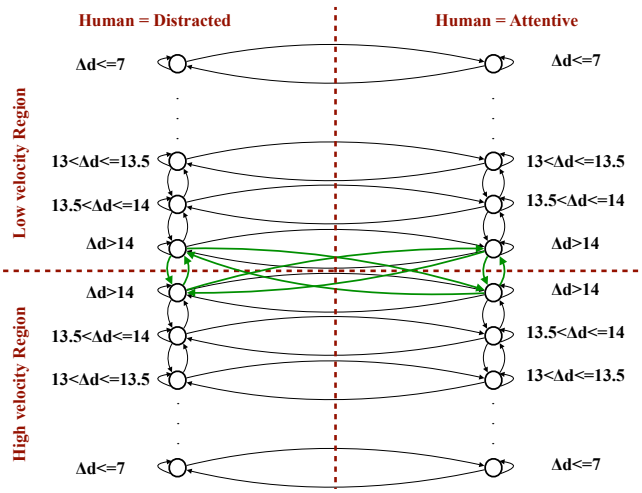

Figure 3: A Markov Decision Process model for the human driver and the vehicle. The states of the MDP corresponds to the state of both the driver (attention level) and the vehicle (relative distance and relative velocity). To capture the fact that human behaviors change over time and across different individuals, the transition probabilities between these states are assumed to be unknown and time-varying. To enhance readability, the relative velocity is quantized into two states (low relative velocity and high relative velocity) and only the transitions of the state (distracted, $\Delta d > 14$, low relative velocity) are shown.

and Δv denotes the relative velocity. As shown in Figure 3, for the same attention level and relative velocity, the relative distance can only increase or decrease by one level at a time (distance can not jump suddenly from $3m$ to $5m$ without being $4m$ in between[5]) and hence any two states $s = (a, \Delta d, \Delta v)$ and $s' = (a, \Delta d \pm q, \Delta v)$ (where q is the quantization of the relative distance) are connected. Similarly, the relative velocity can only increase or decrease by one level at a time, and we connect the states accordingly. We also assume that the human attention level can change at anytime and hence any two states $s = (a, \Delta d, \Delta v)$ and $s' = (a', \Delta d, \Delta v)$ where $a \neq a'$, for the same $\Delta d, \Delta v$, are connected. To capture the fact that human attention may change simultaneously with relative distance and/or relative velocity, any two states $s = (a, \Delta d, \Delta v)$ and $s' = (a', \Delta d \pm q, \Delta v \pm q)$ with $a \neq a'$ are also connected. Indeed, the choice of the quantization of both the relative velocity and relative distance affects both the performance and the complexity of the algorithm of Sentio.

This MDP takes as an input the actions taken by the RL agent. The RL action is whether to issue a warning (warning = 1) or not (warning = 0). Upon receiving any of these actions, the MDP changes its state with some probability. To capture the variation between individuals and the variation within the same individual, the transition probabilities of this MDP are assumed to be unknown and can change over time. In Section 5, we will make use of a reinforcement learning algorithm that continuously learns and updates these unknown transition probabilities and takes actions accordingly.

[4]Prior work in the literature prefers to model the human as Partially Observed Markov Decision Process (POMDP) [32]. POMDP models reflect the fact that sensor data are not capable of measuring the actual human state but measure only a function of the human state. However, POMDP based RL algorithms are computationally intractable hindering their practical use [27]. To overcome these limitations, we model the human as an MDP, and we rely on the increasing success of sophisticated machine learning inference algorithms in estimating the human state.

[5]This assumption is valid given a high enough sampling rate.

4 DYNAMIC & TIME-VARYING REWARDS

The objective of any RL agent is to generate actions that steer the environment from "bad" states into "good" states. The definition of "bad" and "good" is captured by assigning a reward value $R_a(s)$ to each action a in all the states s. Learning algorithms are then used at runtime to monitor the current reward and generate actions that maximize the total reward. In the standard definition of the RL problem, it is assumed that the reward $R_a(s)$ is a time-invariant (static) and is given as an input to the RL agent. However, as we argued before (recall challenge **CH1** in Section 2.3), the reward function depends on the driver preference (relative distance between his car and the leading car) which varies across time and is unknown *a priori*.

4.1 Reward Function Definition

To address this challenge (**CH1**), we define the reward function to consist of two components. One component that is static and can be described offline $R'(s)$, and the other component changes at runtime $I_a(t)$, i.e., we define the reward function as:

$$R_a(s,t) = I_a(t) \times R'(s)$$

where the index t is used to denote the fact that the reward function is time-varying. The fixed component $R'(s)$ reflects the prior bias that lower relative distances are more dangerous than higher relative distances and attentive states are more favorable than distracted states. The time-varying component $I_a(t)$ is binary indicator signal, i.e., $I(t) \in \{-1, 1\}$ reflects the driver acknowledgment to the actions (warnings) triggered by the agent (recall challenge **CH2** in Section 2.3). That is, the proposed Sentio will start first by taking action based on the offline portion of the reward function $R'(s)$. Once the action is taken (trigger an FCW), Sentio monitors the reaction of the driver. If the driver "acknowledges" the warning by applying the brakes, we conclude that the action taken by the RL agent is correct, and hence the final reward is $R_a(s,t) = 1 \times R'(s)$. On the other hand, if the driver does not acknowledge the action taken by the agent, we set $I_a(t)$ to be negative, and the final reward is equal to $R_a(s,t) = -1 \times R'(s)$. While the driver reaction may not be instantaneous (due to different driver response time, recall challenge **CH3** in Section 2.3), the proposed multisample Q-learning algorithm (discussed in Section 5) is designed to address this issue. In our problem setup, we set the reward component $R'(s)$ according to how far s is to the safest state s^*. The safest state s^* is the one where the relative distance is the furthest, the car speed is less than or equal to the leading car and the human is in attentive state. The reward $R'(s)$ is then calculated as:

$$R'(s) = \frac{1}{\text{shortest path between } s \text{ and } s^* + 1} \times 100.$$

4.2 Random Human Actions and Erroneous Rewards

A driver may apply brakes for various reasons other than collision avoidance, e.g., before taking a turn or changing lanes or just a random press on the brakes due to other distractions. If such random braking took place when the car state was within the modeled set of states (i.e., the relative velocity is positive, and the relative distance is below 14m), such human actions might affect the reward received by the RL agent. However, thanks to the continuous monitoring and online learning, this erroneously received rewards will be overridden by the subsequently received rewards. Indeed, between the instance in which the RL agent erroneous rewards and right rewards, the RL agent may cause false positive events (issuing FCW where none is needed) or false negative events (not issuing FCW where one is required). In Section 6, we evaluate the proposed system on human drivers which exhibits such erroneous behavior.

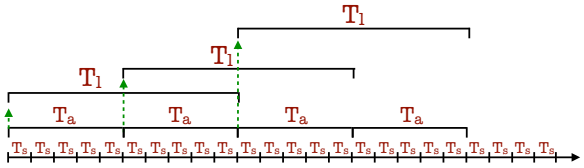

Figure 4: Multisample timescales. An action is taken by the agent every T_a samples. The reward is calculated after an action is taken by a time equals T_l. In this example, $T_a = 5T_s$ and $T_l = 10T_s$, where T_s is the state observing (sensor) rate.

5 MULTISAMPLE Q-LEARNING

Learning the optimal policy—action per state that maximizes the total reward—when the transition probabilities of the MDP model are unknown can be solved using RL. By applying an action in a particular state and observing the next state, the RL converges to the optimal policy that maximizes the reward function. This type of RL technique is called Q-learning algorithm.

5.1 Standard Q-Learning

The Q-learning algorithm assigns a value for every state-action pair. For each state s, the Q-learning algorithm chooses an action a (among the set of allowable actions) according to a particular policy. After an action a is chosen and applied on the environment, the Q-learning algorithm observes the next state s' of the environment and updates the q-value of the pair (s,a) based on the reward of the observed next state as follows:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[R_a(s) + \gamma \max_a Q(s',a) - Q(s,a)]$$

The hyperparameters γ and α are known as the discount factor and the learning step size, respectively. To choose an action a at each state s, an ϵ-greedy algorithm is adopted. In the ϵ-greedy algorithm, the RL agent chooses the action that it believes has the best long-term effect with probability $1-\epsilon$, and it picks an action uniformly at random, otherwise. In other words, at each iteration, the RL agent flips a biased coin and chooses the action with the maximum q-value with probability $1-\epsilon$ or a random action with probability ϵ. This hyperparameter ϵ (also known as the exploration versus exploitation parameter) controls how much the RL agent is willing to explore new actions, that were not taken before, versus relying on the best action that is learned so far.

Note that the standard Q-learning algorithm assumes that the environment reacts instantaneously to the RL actions, changes its state before the next iteration of the RL algorithm is executed, and the reward corresponding to this action is observed. In the context of Sentio, these assumptions are not valid. In particular, due to delays stemming from the human response as well as the mechanical response of the vehicle, the environment state does not change instantaneously in response to actions taken by the RL agent (recall challenge **CH3** in Section 2.3). To address this point, we propose a modified version of the Q-learning algorithm named "Multisample Q-Learning" algorithm.

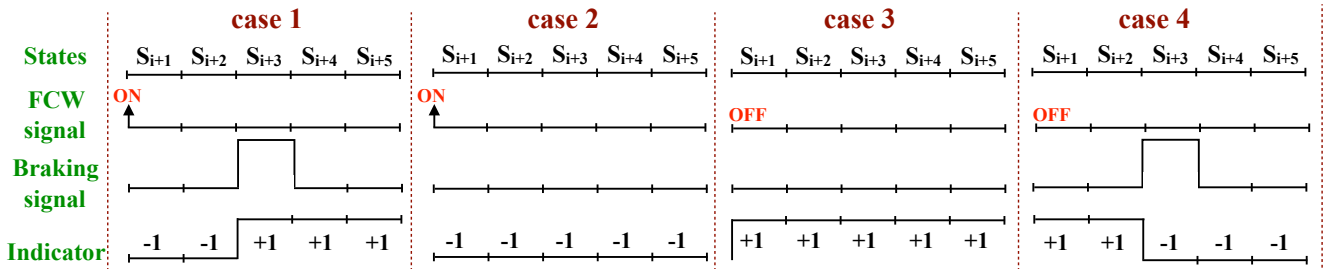

Figure 5: Handling the reward function for four different cases for driver behavior. Case 1: FCW is on and driver acknowledges it by pressing the brakes. Case 2: FCW is on and driver ignores it. Case 3: FCW is off and driver acknowledges it by not pressing the brakes. Case 4: FCW is off and the driver presses the brakes.To enhance the readability of the figure we removed the subscript a and the index t from the notation of $I_a(t)$.

5.2 Multisample Q-Learning Algorithm

A direct approach to address the challenge mentioned above is to collect multiple environment states over a fixed window (or horizon) after each action taken by the RL agent. Meaning, after the RL agent decides as action (e.g., trigger the FCW), the RL agent shall wait for a fixed horizon that captures the delay stemming from the environment response and treat all the information collected within that horizon as one change in the environment. Unfortunately, this approach will force the RL agent to wait for the whole horizon to pass before taking the next action. Taking into account that human response, and hence the fixed horizon, is in the order of seconds [38], we conclude that such approach will force the RL agent to take action every few seconds degrading its ability to react to various scenarios. Another approach is to discard all the out-dated information and take into consideration only the fresh information [41]. While this algorithm is guaranteed to converge to the optimal policy, discarding the information will lead to a significant increase in the convergence time [41] which will occur whenever the driver behavior changes over time.

Therefore, the proposed multisample Q-learning algorithm divides the Q-learning into three different overlapping timescales namely:

- **State observing rate (T_s):** the state of the environment should be monitored whenever new measurements from the sensors are available.
- **Actuation rate (T_a):** to ensure fast reactivity from the agent, the RL agent should decide an action and update (learn) the Q value at a relatively fast rate.
- **Learning rate (T_l):** the reward corresponding to a particular taken action should be evaluated at a relatively slow rate to take into account the delay in the environment change. Once a reward is calculated, the RL agent can update the Q value to reflect the learned rewards so far.

An illustration of the three timescales is shown in Figure 4. These distinctive rates raise other challenges that we address in the next subsections.

5.2.1 ***Finite Horizon Reward.*** Recall that the reward is computed as $R_a(s,t) = I_a(t) \times R'(s)$ where $R'(s)$ represents how far the current state is from the safest state, and $I_a(t)$ represents how happy the driver is with the taken action. Computing the reward across a horizon (instead of a single state) raises the question of how to aggregate all the monitored states within that horizon. For example, consider the case when the RL agent decides that at time t_1 and state s_1 to trigger the FCW. Starting from the time t_1 and until the end of the reward horizon, the driver chooses to acknowledge the warning by pressing the brakes for some period followed by releasing the brake pedal and accelerating the car. Multiple schemes can be proposed to aggregate the reward in this scenario. In the proposed multisample algorithm, we use the heuristic in which the indicator variable $I_a(t)$ latches to the change in the driver acknowledgment, i.e., once the driver acknowledges the RL warning, $I_a(t)$ will remain equal to one for the remainder of the reward horizon. This method is explained in Figure 5. Finally, the aggregate reward is computed as:

$$R_a(s_1, t_1) = \sum_{i=t_1}^{t_1+n} I_a(i) \times R'(s_i)$$

where n is the reward horizon. Note that, unlike standard Q-learning where the reward is not a function of time, the reward now is assigned to a particular action that is taken at time t_1 at the beginning of the reward horizon.

5.2.2 ***Learning Update Rate and Overlapping Rewards.*** To enhance the reactivity of the RL agent, the RL agent decides the next action even before the reward horizon of the previous action is collected and aggregated leading to an overlap between the effects of different actions. For example, consider the case when the RL agent decides that at time t_1 *not* to trigger the FCW. Before the reward is calculated for this action, the RL agent decides at time t_2 to trigger the FCW. Both the actions will affect the environment during the time $t_1 \leq t \leq t_1 + n$ in which the reward for the first action is being collected. This raises the question of how to accommodate overlapping effects on the collected reward. To address this question, we use multiple indicators $I_a(t)$ one per each action (the subscript a is to emphasize the fact that multiple indicator variables are used, one per each action) taken in the previous $t - n$ horizon. Each indicator is handled differently according to the rules depicted in Figure 5. We assume that the inaccuracy that may arise while calculating the rewards due to different actions in an overlapped reward horizon will have a minimal effect over a long time due to the feedback nature we have in Sentio. The effect of choosing the learning update rate is studied in the evaluation shown in Section 6. Algorithm 1 summarizes the details discussed in Sections 4 and 5.

6 EXPERIMENTAL RESULTS

We evaluate the proposed Sentio by recruiting a total of eleven human drivers (six males and five females) with ages that vary between 22 and 35 years. To ensure the safety of the recruited drivers, reduce liability, and meet the ethical standards of experimenting

Figure 6: Vehicle dynamics virtualization testbed used in the evaluation study .

Algorithm 1 Sentio Multisample Q-learning algorithm

Hyper parameters: Learning parameters: α, γ, ϵ
Time scales: $T_s \leq T_a \leq T_l$
Initialization routine:
$Q(s, a_{\text{no warning}}) = 1 \quad \forall s \in S$;
$Q(s, a_{\text{trigger warning}}) = 0 \quad \forall s \in S$;
Push the state s^* into the queue M;
numberOfHops = 0;
while M is not empty **do**
Dequeue state s from M;
$R'(s) = \frac{1}{\text{numberOfHops}+1} \times 100$;
Enqueue all states that can reach state s in M';
if M is empty **then**
Copy M' into M;
numberOfHops = numberOfHops + 1;
return $R'(s)$, $Q(s, a)$

State Observation Routine (activated every T_s)
Collect data from car and phone sensors;
Run inference to produce environment state s;
Push current state to the state stack S;

Actuation Routine (activated every T_a)
Pull the current state s from the state stack S;
Choose action a using ε-greedy policy;
Save action to the action list A;
Apply action to the driver;

Learning Routine (activated every T_l)
For all actions a in the stack A **do:**
τ = current time;
s' = current environment state;
s = the state at which the action a was taken;
$t = \tau - T_l$ (time at which the action a was taken);
$R_a(s, t) = 0$;
For all times t_i between $\tau - T_l$ and τ **do**:
Compute the indicator variable $I_a(t_i)$;
Get the state s_i at time t_i;
$R_a(s, t) = R_a(s, t) + I_a(t_i) \times R'(s_i)$
Update the Q matrix accordingly

$$Q(s, a) \leftarrow Q(s, a) + \alpha[R_a(s, t) + \gamma \max_a Q(s', a) - Q(s, a)]$$

on human subjects, we decided to use a simulation environment. In this environment, an industry-level physics simulator (named CarSim [24]) is used to simulate the physics of both the driving car and the leading car as shown in Figure 6. In particular, CarSim is used to simulate the longitudinal dynamics of the car. We interfaced CarSim with the Matlab virtual reality toolbox (VR) to virtualize the vehicle dynamics. In this setting, we define two physical car dynamics; a leading car and a following car as shown in Figure 6. Each car has parameters that specify its mass, the mechanical friction of the wheels on the ground, its acceleration and braking forces. We implemented Sentio on the following car (the yellow car in Figure 6). Humans drive the yellow car in the longitudinal direction using a driving wheel and a pedal that send driving signals to the physics-level simulator. To simulate distraction, the human drivers are engaged in a conversation that aims at raising their cognition load [23].

Our evaluation uses the following metrics:

(1) **Learning performance:** which is quantified by:
- **False Negative (FN) Rate:** False negatives occur whenever the car violates the safety distance (relative distance below 7m) as a result of the FCW decides not to warn the driver early enough. Lower FN rate reflects better performance.
- **False Positive (FP) Rate:** This metric quantifies the rate of false alarms. That is, the rate at which the FCW is triggered unnecessarily. Lower FPR reflects better performance.

(2) **Driver safety:** While different violations of the safety distance have different severity (in terms of the relative distance and the period for which the violation lasted), we quantify the driver safety using the **Violation Severity (VS)** which is computed as a weighted sum of the violated distance over the violation time. Lower SV reflects higher driver safety.

(3) **Driving experience:** A better driving experience is the one in which the driver does not use high braking intensity more often. Therefore, we use the histogram of the **Braking Intensity (BI)** as an indication of the driving experience.

We divide our evaluation into two sections. In the first section, we aim to tune the parameters of the RL algorithm (e.g., explore the tradeoff between exploration/exploitation, learning rate, and learning step size). In the second section, we use the proposed RL

algorithm (with the parameters tuned from the first set of experiments) to evaluate the driver safety and experience and compare it against the fixed-threshold FCW that is used in modern cars.

6.1 Parameter Tuning

To study the effect of each of the hyper-parameters and correctly tune them, we need a controlled environment in which we can repeat the same exact experiment multiple times using different tuning parameters. Indeed, human behavior varies between each experiment and is not reproducible. For that end, we started by using the physics simulator along with the interfacing wheel and pedal to collect several driving traces from all the recruited drivers aiming to build a human simulator that can be used for parameter tuning. Analyzing the 11 human driver traces, we can categorize the sample of drivers obtained into three categories—based on their reaction to the FCW—as follows:

- **Defensive drivers (2 out of 11 drivers)**: These drivers press the brakes directly once they hear the FCW. Afterwards, they check the relative distance to the leading car and decide whether the braking action they took was a right decision or it was a false alarm.
- **Aggressive drivers (1 out of 11 drivers)**: After hearing the FCW, these drivers do not react to the warning until they first check the environment and respond accordingly.
- **Assertive drivers (8 out of 11 drivers)**: After hearing the FCW, these drivers release the acceleration pedal (as a precaution), check the relative distance with the leading car, and then decide whether to press the brake or not.

Moreover, our data show four distinguishing aspects between the assertive drivers namely:

(1) **Braking intensity:** which defines how intense the driver tends to press the brake pedal.
(2) **Acceleration intensity:** which describes how fast/slow the driver prefers to drive based on the intensity of pressing the acceleration pedal.
(3) **Comfort braking distance:** which specifies the preferable relative distance that the driver likes to keep between his car and the leading car.
(4) **Response time:** which defines the time taken by the driver to observe the relative distance to the leading car and take action (whether presses the brake or ignores the warning).

According to these observations, we built a human driver simulator to mimic the behavior of the assertive drivers (since the majority of the driver sample we got was assertive). The simulator continuously picks—at random—the values for the braking intensity, acceleration intensity, and response time.

6.1.1 ***Experiment 1a: Parameter Tuning.*** We examine the trade-off between exploration and exploitation (ϵ) in choosing the appropriate action for the current context (vehicle context and human context) along with the learning parameters (the discount factor (γ) and the learning step size (α)). Figure 7 shows the traces of the relative distance across the different choice of parameters[6] with their corresponding false positives and false negatives pattern.

- **Effect of the learning step α:** First, we fix the values of ϵ and γ and sweep the value of the learning step α between $\alpha = 1$ and $\alpha = 0.6$. We observe that for higher values of α (Figure 7 - case 1) the number of false positives does not decrease with time which signals that the RL agent is not able to learn the human preference and not able to produce the warnings in a way that satisfies the driver. However, by decreasing the learning step α—which has the effect of asking the RL agent to use multiple data for learning instead of depending entirely on the latest learned data—we observe that the number of false positives decreases over time (Figure 7 - case 2). We conclude that a low value of the learning step α is needed.
- **Effect of the discount factor γ:** Fixing the value of α at the best value from the previous experiment $\alpha = 0.6$, we start to

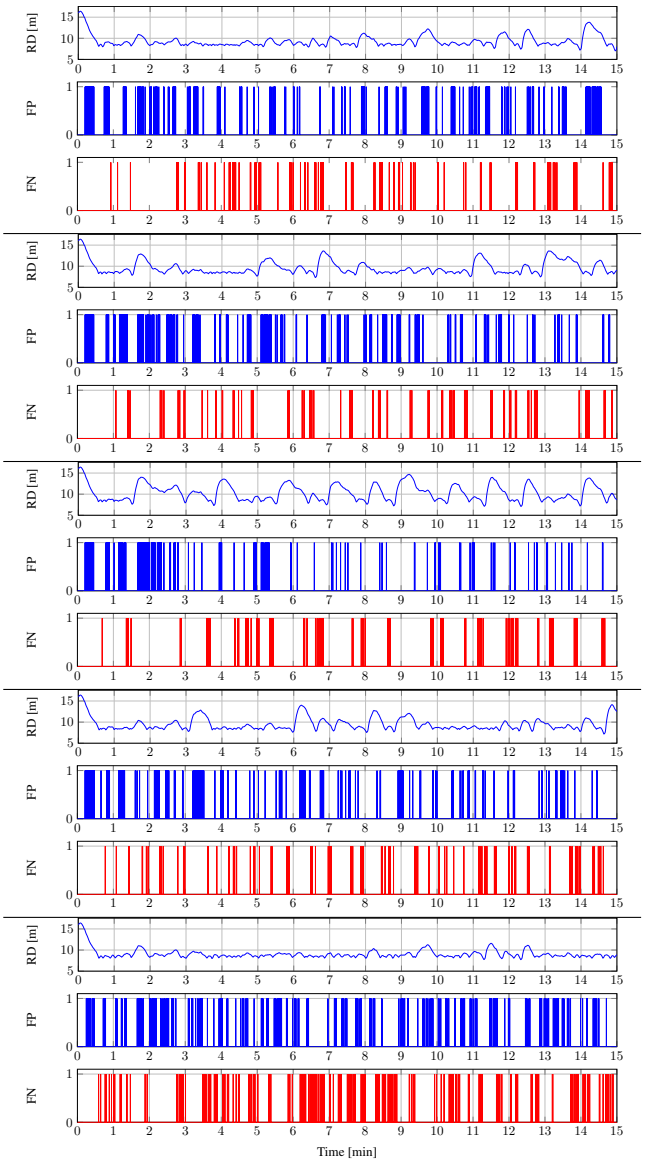

Figure 7: False positives and false negatives with different learning parameters for five driving traces across 15 minutes simulation time: (1) $\epsilon = 0.1, \alpha = 1.0, \gamma = 1.0$, (2) $\epsilon = 0.1, \alpha = 0.6, \gamma = 1.0$, (3) $\epsilon = 0.1, \alpha = 0.6, \gamma = 0.8$, (4) $\epsilon = 0.1, \alpha = 0.6, \gamma = 0.7$, and (5) $\epsilon = 0.5, \alpha = 0.6, \gamma = 0.8$.

[6]Due to space limit, we show only 5 choices of parameters.

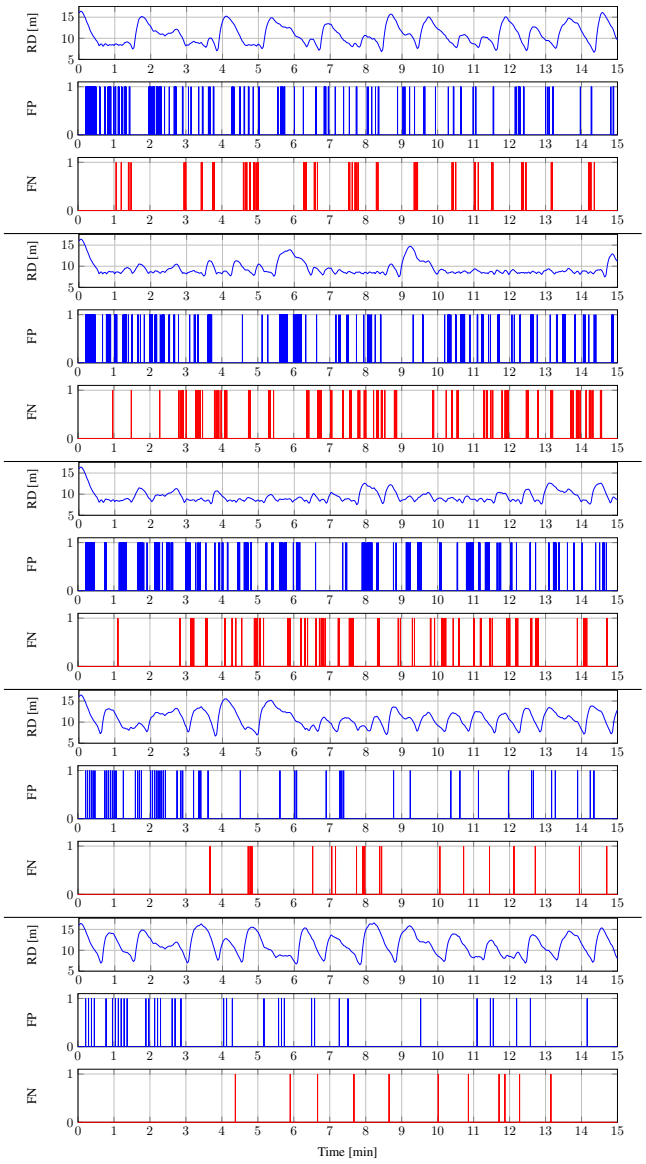

Figure 8: False positives and false negatives with different values for T_a and T_l for five driving traces across 15 minutes simulation time using $\epsilon = 0.1, \alpha = 0.6, \gamma = 0.8$: (1) $T_a = 0.5s, T_l = 2.5s$, and (2) $T_a = 0.5s, T_l = 4s$, (3) $T_a = 0.5s, T_l = 5s$, (4) $T_a = 2.5s, T_l = 5s$, (5) $T_a = 4s, T_l = 5s$.

sweep the values of the discount factor γ and compare the results (Figure 7 - case 2, case 3, and case 4). We observe again that lowering the discount factor γ leads to enhancing the number of false positives as seen by comparing the results in case 2 ($\gamma = 1.0$) and case 3 ($\gamma = 0.8$) of Figure 7. However, by decreasing the discount factor further, we note that the number of false positives starts to increase again as evident by case 4 ($\gamma = 0.7$) in Figure 7. We conclude that a value of $\gamma = 0.8$ achieves the best results.

- **Effect of the exploration/exploitation factor ϵ:** Finally, we study the impact of the exploration/exploitation factor ϵ. This factor controls the confidence of the RL agent on the current Q values. Lower values of ϵ reduce the probability of picking a random action and tend to decide actions based on the learned Q values. A higher value of ϵ asks the RL agent to explore more actions at random aiming to test actions not taken before. Comparing the results in Figure 7 for case 3 ($\epsilon = 0.1$) and case 5 ($\epsilon = 0.5$) we observe that increasing ϵ leads to an increase in both false positives and false negatives. We conclude that $\epsilon = 0.1$ leads to the best results.

6.1.2 ***Experiment 1b: Timescales Tuning.*** After fixing the values of the learning step α, the discount factor γ, and the exploration/-exploitation factor ϵ, we examine the other hyper-parameters in the multisample Q-learning algorithm (the state observation rate T_s, the actuation rate T_a, and the learning rate T_l). Since T_s is constrained by the availability of the sensory information and the state interference algorithm, we fixed it at 0.25 seconds and we examined different combinations for T_a and T_l. In Figure 8 we show the traces for five of these combinations. First, we fixed the value of the T_a at 0.5 seconds and changed the value of T_l. We noticed that the higher the value of T_l, the smoother the trace of the relative distance which indicates a better driving experience. This entails that the more we incorporate the time response of the human in T_l, the more the human enjoys a better driving experience. Next, we fixed T_l to 5 seconds and examined different values for T_a. Although FN and FP appeared to be less as we increased the value of T_a, the FNR increased as we increased T_a. In the case when T_a is 2.5 seconds the FNR increased to 3.11% compared to 2.44% in the case when T_a is 0.5 seconds. On the other hand, the FPR decreased as T_a increased. In particular, when T_a is 0.5 seconds the FPR is 4.05% compared to 3.25% when T_a is 2.5 seconds. This shows a tradeoff between smooth driving experience and low values for FPR/FNR. Hence, we chose the values of T_a and T_l to be 0.5 seconds and 5 seconds respectively which achieve the smoothest driving trace with low FNR at the expense of a little higher FPR.

6.1.3 ***Experiment 2: Tracking driver changing behavior.*** While the previous parameter tuning is studied by using the same type of driving behavior (assertive driving) and for a fixed driver preference, the next step is to test the performance of the tuned RL agent against different driving behaviors and for changing driver preference (with respect to the relative distance to the leading car). We use our human simulator to synthesize two driver behaviors; one simulates an assertive driver while the second simulates an aggressive driver over a 30 minutes of simulation time. During this time, the behavior of the driver (with respect to the preferred relative distance) changes. We imposed distraction over these two synthesized drivers (by increasing the response time and braking intensity) [6] and observed the difference between the proposed Sentio and the classical FCW (which uses a fixed ratio of relative distance and relative velocity to signal the alert) typically used in modern cars. Ultimately, we want the RL agent to learn to issue the alert ahead of time when the driver is distracted preventing the vehicle to violate the dangerous $< 7m$ safety distance.

To compare the performance of the two systems (fixed vs. Sentio), we plot both the driving trace (relative distance) along with the violation of the safety constraint (i.e., the amount of time and distance below $7m$) for each case in Figure 9. In the case of a fixed alert threshold, we observe that the violation in the case of aggressive drivers is higher compared to the assertive drivers. Comparing the

Figure 9: Relative distances and violations of two distracted drivers with a change in behavior over time over a simulation time of 30 minutes using Sentio and with a fixed threshold alert.

fixed threshold policy of both the assertive and aggressive drivers to the proposed Sentio, we observe that (1) the relative distance is fluctuating more whenever the driver is distracted. This reflects the fact that the simulated driver brakes only after hearing the alarm due to the distraction effect. This is more severe in the case of the aggressive driver due to using higher braking intensity (2) a reduction in the violation of the driver safety, thanks to the fact that the RL agent was able to learn the driver behavior and issue the warning early enough even if the driver preference changes over time. In particular, the VS metric is reduced by 85.8% for the assertive driver and reduced by 90.25% for the aggressive driver.

6.2 Human Driving Experience

After gaining confidence in the proposed Sentio in a controlled environment, the next step is to ask the same recruited human drivers to use the proposed Sentio. We compare their driving experience against (i) the fixed-threshold system, (ii) the standard Q-learning (where observing state, learning, and taking actions occur every sample), (iii) the delayed Q-learning (where observing state, learning, and taking actions occur after a fixed delay to take into account the human response time), and (iv) the proposed multisample Q-learning. Every driver is asked to drive for 30 minutes (average commute time in the US is 25.4 minutes [42]) in each experiment while one of the four algorithms is chosen randomly at runtime (to remove driver bias). To examine the effect of random behavior as discussed in Section 4.2, they are free to (1) brake at anytime regardless whether or not they hear an FCW and (2) decide their acceleration/deceleration on the pedal. Drivers are asked to engage in conversations with others while driving. A machine learning classifier is used to distinguish between attentive/distracted states. We trained a logistic regression model using recordings from scripted dialogues from screenplays in English supplemented with recordings from MUSCAN [36], CallFriend [4], CallHome [3], and Librivox [20]. Our dataset had 3 categories: dialogue, monologue, and non-speech. The first category contains recordings of both individuals engaged in dialogue. Additional recordings come from the CallFriend and CallHome datasets, which contain audio samples of two-person, unscripted telephone conversation. The second category, monologue, consists of recordings of one individual engaged in the script dialogue. Supplemental audio files from the Librivox corpus dataset, where each WAV file contains a spoken, English-only recording of a chapter in a book. The last category is the combination of audio clips from the music and noise portions of the MUSCAN dataset. One of the most widely-used features for acoustic signal processing is Mel Frequency Cepstral Coefficient (MFCC), as it is a fair approximation of human perceptions of pitch[12]. Other common acoustic features exist, such as Linear Predictive Coding (LPC), but we present strong and robust results with MFCC alone. Extracting features from long audio segments is computationally expensive. Hence, we split each WAV file into smaller chunks. It is difficult to predict a priori what time chunk length yields the best compromise between model accuracy and runtime efficiency. Thus, we tested three different chunk lengths: 10, 20, and 30 seconds and two models; support vector machines and logistic regression. Based on our results, we used the best performance which stemmed from using chunks of 20 seconds and a logistic regression model. Our trained model could predict if the person is engaged in a conversation with 97.2% accuracy on a 5-fold cross-validation within the training set and 97% accuracy on the test set.

We show in Figure 10 and Figure 11 the traces for the first 3.5 minutes (showing the learning transients) of two drivers along with the corresponding traces for the four metrics (VS, FP, FN, and BI histogram). We then report the results of the four metrics for all the 11 human drivers in Table 1. Below are the observations and conclusions drawn from these experiments.

- **Fixed threshold:** Fixed threshold leads to the worst driving experience whenever the driver is distracted. This is evident by the histogram of the braking intensity where we observe that drivers are obliged to use high braking intensity (pressing the braking pedal all the way to its maximum limit) to avoid a collision. Even with such driving experience, we observe that violations are not entirely removed, but yet drivers violate the safety constraint.
- **Traditional Q-learning:** This method leads to no learning and a random behavior of issuing the warning. This is best captured

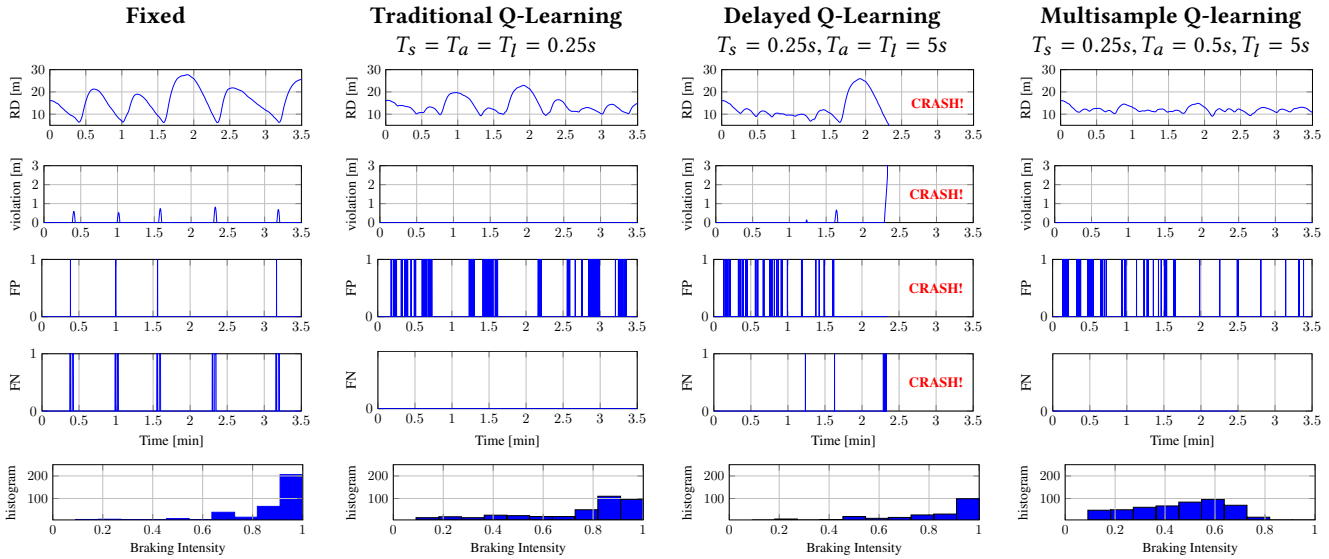

Figure 10: Driving Safety and Experience for Driver #1.

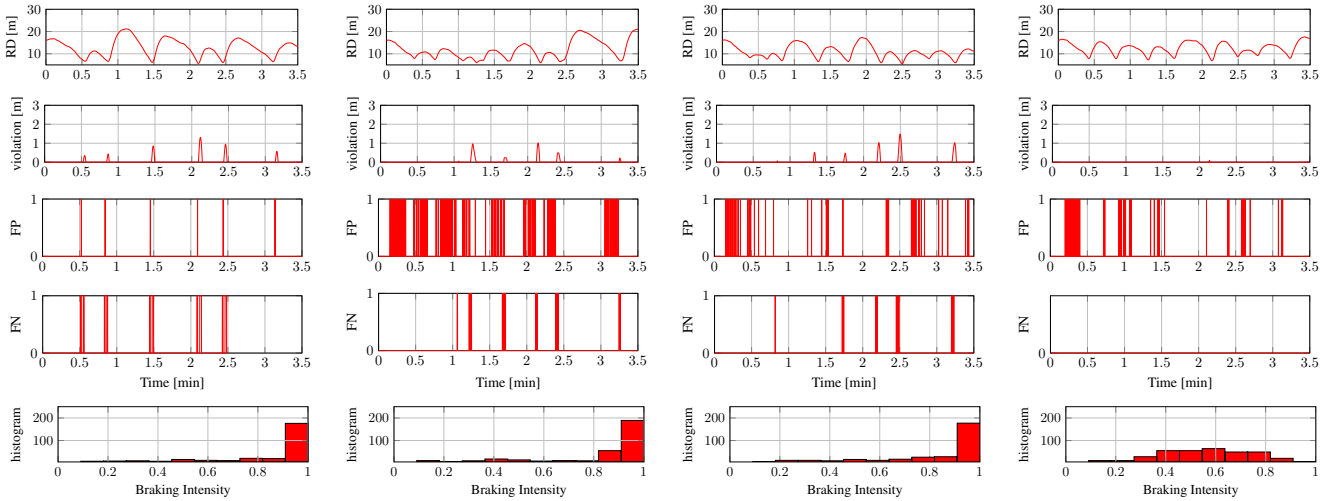

Figure 11: Driving Safety and Experience for Driver #2.

by the high amount of false positives across time. While this method reduced the violation compared to the fixed threshold, this is mainly due to a large number of warnings produced by this method (with most of them being false). This behavior can be accounted for the fact that rewards are computed based on the instantaneous values of the environment state without taking into account the delay in the human response.

- **Delayed Q-learning:** While the learning behavior (and hence the FP rate) are enhanced compared with the traditional Q-learning, we observe that it was not able to warn the first driver early enough to avoid the crash. This stems from the fact that this method learns and takes action only after the whole reward horizon has ended leading to missing critical events and hence caused a crash.
- **Multisample Q-learning:** Similar to the delayed Q-learning, we observe that the proposed multisample Q-learning can learn the preference of the driver leading to a decrease in both FN and VS metrics with the cost of an increase in the FP especially while learning. However, thanks to the multisampling of the proposed method, we observe an increased safety compared to the delayed Q-learning (measured by the VS metric). Moreover, we observe that the histogram of the braking intensity (and hence the driving experience) has improved significantly where the drivers tend to use smaller barking intensities thanks to the fact that warnings are produced early enough leaving the driver with enough time to brake slowly.

Table 1 shows the results for the same metrics for all the 11 humans across the entire 30 minutes experiment time. These results reflect similar conclusions. On average, Sentio reduced the violation severity by 94.28%, reduced the mean of the braking intensity by 20.97%, reduced the false negative rate from 55.90% down to 3.26%. These improvements come at the cost of increasing the false positives (false alarms) from 2.71% to 5.15%, on average. Indeed, most of these false alarms occur during the initial learning phase and their rate decreases afterwards. Finally, we informally asked the drivers about their experience; all drivers reported the excessive

	Fixed			Sentio		
	VS	BI (mean)	FNR [%] / FPR [%]	VS	BI (mean)	FNR [%] / FPR [%]
H1 (A)	85.379	0.852	54.5 / 0.22	0.501	0.419	0.58 / 1.88
H2 (S)	127.11	0.704	52.0 / 0.78	0.844	0.542	0.61 / 2.71
H3 (D)	54.294	0.41	61.61 / 0.49	0	0.496	0.66 / 3.45
H4 (S)	136.212	0.713	42.53 / 0.95	6.087	0.593	0.52 / 9.79
H5 (S)	295.411	0.702	99.0 / 2.46	11.03	0.536	18.69 / 8.37
H6 (S)	258.412	0.803	98.1 / 9.83	67.82	0.735	7.94 / 9.72
H7 (S)	105.205	0.84	52.9 / 6.33	0.707	0.749	1.85 / 7.81
H8 (D)	28.35	0.561	57.3 / 1.52	0.134	0.516	0.61 / 0.81
H9 (S)	185.755	0.739	19.7 / 1.28	1.560	0.604	0.39 / 4.53
H10 (S)	221.086	0.747	49.2 / 5.41	2.496	0.510	3.51 / 6.25
H11 (S)	121.158	0.796	28.1 / 0.62	1.289	0.516	0.51 / 1.28
AVG	147.124	0.715	55.90 / 2.71	8.406	0.565	3.26 / 5.145

Table 1: Comparison between the performance of fixed threshold FCW and multisample Q-learning in Sentio for 11 distracted drivers (A = Aggressive Driver, S = Assertive Driver, and D = Defensive Driver). Degradation in metric performance (with respect to Fixed policy) is marked in red.

false alarms at the beginning of the experiments. However, they felt that the false alarms rate decreased significantly while driving. Moreover, nine drivers assured that, in the case of Sentio, the alert was given at a proper time to examine the environment when they were distracted. However, the two defensive drivers reported that the alert was given too early and they had to brake unnecessarily.

6.3 Execution Time Analysis

We report the execution time of the proposed method. Recall that Algorithm 1 consists of different routines that are executed at different rates. Averaging out the execution time across all drivers, we observe that the initialization routine consumes 61.86 seconds. Fortunately, this routine is called offline, and hence it does not affect the online execution time. The state observation (and inference) routine consumes 20.4 ms, the actuation routine consumes 2.59 ms, while the learning routine consumes 104 ms which is negligible compared to the human response time.

7 DISCUSSIONS

Although Q-learning based algorithms enjoy strong mathematical guarantees regarding convergence to the optimal policy (action per state), the mathematical guarantees for its safety are still lacking. Recently, there have been new studies in the literature that extends RL algorithms to safe RL in which safety constraints are incorporated in the learning and deployment processes. One example of such work is constraining the exploration strategy using some prior knowledge to avoid going into risky situations [10, 16]. Nevertheless, the relation between the optimality of the policy and the safety of the policy is indeed an interesting area to explore and study in future work.

Indeed, Sentio does not come without limitations. As shown in the experimental results, the false alarm rate is higher on average compared to the fixed threshold FCW. A possible solution is to change the hyper-parameters over time. In particular, the exploration versus exploitation parameter ϵ plays a significant role in the increase of the false alarms. However, ϵ also plays a significant role in adapting to the changes in the driver behavior. One possible track is to change ϵ across time based on our confidence in how much the human behavior will be fixed. Moreover, the human state is complex and it is not merely binary (distracted/attentive) but a continuous range of values. A next step would be to explore more elaborate human models that take into account different human behavior and study the effect of such elaborate models on the performance of Sentio. Industrial-level testing and verification of Sentio across a broader range of drivers' states, weather conditions, and road conditions along with real-life deployment is the next step in this research.

8 CONCLUSION

While FCW is one of the highly adopted ADAS in vehicles, its primary focus is still dependent on the environment around the vehicle (relative distance and relative velocity). However, by taking the driver state into the loop of computation, the FCW can be personalized and provide a better experience. In this paper, we purposed Sentio, a driver-in-the-loop FCW that learns the driver preference as well as his attention level to signal the FCW at the right time. We proposed a variant of the Q-learning algorithm to solve our problem and enhance the learning rate of the driver preference. Our multisample Q-learning algorithm could track the change in the driving behavior across time and adapt the timing of the FCW accordingly without the need for offline training. By examining the results of driving traces on a car simulator for multiple drivers, Sentio shows a better performance than the fixed threshold FCW that is widely used in commercial vehicles.

ACKNOWLEDGMENTS

This research was supported in part by the NIH Center of Excellence for Mobile Sensor Data-to-Knowledge (MD2K) under award 1-U54EB020404-01, the U.S. Army Research Laboratory and the UK Ministry of Defense under Agreement Number W911NF-16-3-0001, the National Science Foundation under awards #OAC-1640813 and CNS-1329755, and the King Abdullah University of Science and Technology (KAUST) through its Sensor Innovation research program. The Microsoft Research PhD Fellowship has supported Salma Elmalaki. Any findings in this material are those of the author(s) and do not reflect the views of any of the above funding agencies. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

REFERENCES

[1] Christer Ahlstrom, Katja Kircher, and Albert Kircher. 2013. A gaze-based driver distraction warning system and its effect on visual behavior. *IEEE Transactions on Intelligent Transportation Systems* 14, 2 (2013), 965–973.

[2] Jeffrey S Campbell, Sidney N Givigi, and Howard M Schwartz. 2014. Multiple-model Q-learning for stochastic reinforcement delays. In *IEEE International Conference on Systems, Man and Cybernetics (SMC), 2014*. IEEE, 1611–1617.

[3] Alexandra Canavan, David Graff, and George Zipperlen. 1997. Callhome american english speech. *Linguistic Data Consortium* (1997).

[4] Alexandra Canavan and George Zipperlen. 1996. Callfriend american english-non-southern dialect. *Linguistic Data Consortium, Philadelphia* 10 (1996), 1.

[5] Lan-lan Chen, Yu Zhao, Jian Zhang, and Jun-zhong Zou. 2015. Automatic detection of alertness/drowsiness from physiological signals using wavelet-based nonlinear features and machine learning. *Expert Systems with Applications* 42, 21 (2015), 7344–7355.

[6] William Consiglio, Peter Driscoll, Matthew Witte, and William P Berg. 2003. Effect of cellular telephone conversations and other potential interference on reaction time in a braking response. *Accident Analysis & Prevention* 35, 4 (2003), 495–500.

[7] Jiangpeng Dai, Jin Teng, Xiaole Bai, Zhaohui Shen, and Dong Xuan. 2010. Mobile phone based drunk driving detection. In *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2010 4th International Conference on-NO PERMISSIONS*. IEEE, 1–8.

[8] Mandalapu Sarada Devi and Preeti R Bajaj. 2008. Driver fatigue detection based on eye tracking. In *Emerging Trends in Engineering and Technology, 2008. ICETET'08. First International Conference on*. IEEE, 649–652.

[9] DMV. [n. d.]. How Emotions Affect Driving. https://www.dmv.org/how-to-guides/driving-and-emotions.php. ([n. d.]).

[10] Kurt Driessens and Saso Dzeroski. 2004. Integrating guidance into relational reinforcement learning. *Machine Learning* 57, 3 (2004), 271–304.

[11] Salma Elmalaki, Lucas Wanner, and Mani Srivastava. 2015. Caredroid: Adaptation framework for android context-aware applications. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 386–399.

[12] Todor Ganchev, Nikos Fakotakis, and George Kokkinakis. 2005. Comparative evaluation of various MFCC implementations on the speaker verification task. In *Proceedings of the SPECOM*, Vol. 1. 191–194.

[13] Jennifer Healey, Rosalind W Picard, et al. 2005. Detecting stress during real-world driving tasks using physiological sensors. *IEEE Transactions on intelligent transportation systems* 6, 2 (2005), 156–166.

[14] Tim Horberry, Janet Anderson, Michael A Regan, Thomas J Triggs, and John Brown. 2006. Driver distraction: The effects of concurrent in-vehicle tasks, road environment complexity and age on driving performance. *Accident Analysis & Prevention* 38, 1 (2006), 185–191.

[15] Pedro Jiménez, Luis M Bergasa, Jesús Nuevo, Noelia Hernández, and Ivan G Daza. 2012. Gaze fixation system for the evaluation of driver distractions induced by IVIS. *IEEE Transactions on Intelligent Transportation Systems* 13, 3 (2012), 1167–1178.

[16] Rogier Koppejan and Shimon Whiteson. 2009. Neuroevolutionary reinforcement learning for generalized helicopter control. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*. ACM, 145–152.

[17] Terry C Lansdown and Amanda N Stephens. 2013. Couples, contentious conversations, mobile telephone use and driving. *Accident Analysis & Prevention* 50 (2013), 416–422.

[18] Donghoon Lee and Hwasoo Yeo. 2015. A study on the rear-end collision warning system by considering different perception-reaction time using multi-layer perceptron neural network. In *Intelligent Vehicles Symposium (IV)*. IEEE, 24–30.

[19] Stéphanie Lefèvre, Ashwin Carvalho, Yiqi Gao, H Eric Tseng, and Francesco Borrelli. 2015. Driver models for personalised driving assistance. *Vehicle System Dynamics* 53, 12 (2015), 1705–1720.

[20] LibriVox. [n. d.]. librivox-free public domain audiobooks. https://librivox.org. ([n. d.]).

[21] Benoit Mariani, Mayté Castro Jiménez, François JG Vingerhoets, and Kamiar Aminian. 2013. On-shoe wearable sensors for gait and turning assessment of patients with Parkinson's disease. *IEEE transactions on biomedical engineering* 60, 1 (2013), 155–158.

[22] Ralph Oyini Mbouna, Seong G Kong, and Myung-Geun Chun. 2013. Visual analysis of eye state and head pose for driver alertness monitoring. *IEEE transactions on intelligent transportation systems* 14, 3 (2013), 1462–1469.

[23] Jason S McCarley, Margaret J Vais, Heather Pringle, Arthur F Kramer, David E Irwin, and David L Strayer. 2004. Conversation disrupts change detection in complex traffic scenes. *Human factors* 46, 3 (2004), 424–436.

[24] Mechanical Simulation Corporation. [n. d.]. Mechanical Simulation. https://www.carsim.com. ([n. d.]).

[25] F. Muehlfeld, I. Doric, R. Ertlmeier, and T. Brandmeier. 2013. Statistical Behavior Modeling for Driver-Adaptive Precrash Systems. *IEEE Transactions on Intelligent Transportation Systems* 14, 4 (Dec 2013), 1764–1772. https://doi.org/10.1109/TITS.2013.2267799

[26] Subhas Chandra Mukhopadhyay. 2015. Wearable sensors for human activity monitoring: A review. *IEEE sensors journal* 15, 3 (2015), 1321–1330.

[27] Kevin P Murphy. 2000. A survey of POMDP solution techniques. *environment* 2 (2000), X3.

[28] Joao B Pinto Neto, Lucas C Gomes, Eduardo M Castanho, Miguel Elias M Campista, Luís Henrique MK Costa, and Paulo Cezar M Ribeiro. 2016. An error correction algorithm for forward collision warning applications. In *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 1926–1931.

[29] New York State Department of Motor Vehicles. [n. d.]. Defensive Driving. https://dmv.ny.gov/about-dmv/chapter-8-defensive-driving. ([n. d.]).

[30] Eshed Ohn-Bar and Mohan Manubhai Trivedi. 2016. Looking at humans in the age of self-driving and highly automated vehicles. *IEEE Transactions on Intelligent Vehicles* 1, 1 (2016), 90–104.

[31] Wang Rongben, Guo Lie, Tong Bingliang, and Jin Lisheng. 2004. Monitoring mouth movement for driver fatigue or distraction with one camera. In *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*. IEEE, 314–319.

[32] Stephanie Rosenthal and Manuela Veloso. 2011. Modeling humans as observation providers using pomdps. In *RO-MAN, 2011 IEEE*. IEEE, 53–58.

[33] Peter Rowden, Gerald Matthews, Barry Watson, and Herbert Biggs. 2011. The relative impact of work-related stress, life stress and driving environment stress on driving outcomes. *Accident Analysis & Prevention* 43, 4 (2011), 1332–1340.

[34] Keshav Seshadri, Felix Juefei-Xu, Dipan K Pal, Marios Savvides, and Craig P Thor. 2015. Driver cell phone usage detection on strategic highway research program (SHRP2) face view videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 35–43.

[35] Tal Shany, Stephen J Redmond, Michael R Narayanan, and Nigel H Lovell. 2012. Sensors-based wearable systems for monitoring of human movement and falls. *IEEE Sensors Journal* 12, 3 (2012), 658–670.

[36] David Snyder, Guoguo Chen, and Daniel Povey. 2015. Musan: A music, speech, and noise corpus. *arXiv preprint arXiv:1510.08484* (2015).

[37] Chen Su, Weiwen Deng, Hao Sun, Jian Wu, Bohua Sun, and Shun Yang. 2017. Forward collision avoidance systems considering driver's driving behavior recognized by Gaussian Mixture Model. In *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 535–540.

[38] Heikki Summala. 2000. Brake reaction times and driver behavior analysis. *Transportation Human Factors* 2, 3 (2000), 217–226.

[39] Ashish Tawari, Sayanan Sivaraman, Mohan Manubhai Trivedi, Trevor Shannon, and Mario Tippelhofer. 2014. Looking-in and looking-out vision for urban intelligent assistance: Estimation of driver attentive state and dynamic surround for safe merging and braking. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 115–120.

[40] The Statistics Portal. [n. d.]. Projected global ADAS revenue growth trend from 2012 to 2020. https://www.statista.com/statistics/442726/global-revenue-growth-trend-of-advanced-driver-assistance-systems/ . ([n. d.]).

[41] John N Tsitsiklis. 1994. Asynchronous stochastic approximation and Q-learning. *Machine learning* 16, 3 (1994), 185–202.

[42] U.S Census Bureau. [n. d.]. Commuting Times, Median Rents and Language other than English Use in the Home on the Rise. https://www.census.gov/newsroom/press-releases/2017/acs-5yr.html. ([n. d.]).

[43] Jianqiang Wang, Chenfei Yu, Shengbo Eben Li, and Likun Wang. 2016. A forward collision warning algorithm with adaptation to driver behaviors. *IEEE Transactions on Intelligent Transportation Systems* 17, 4 (2016), 1157–1167.

[44] Xuesong Wang, Ming Chen, Meixin Zhu, and Paul Tremont. 2016. Development of a Kinematic-Based Forward Collision Warning Algorithm Using an Advanced Driving Simulator. *IEEE Transactions on Intelligent Transportation Systems* 17, 9 (2016), 2583–2591.

[45] Martin Wollmer, Christoph Blaschke, Thomas Schindl, Björn Schuller, Berthold Farber, Stefan Mayer, and Benjamin Trefflich. 2011. Online driver distraction detection using long short-term memory. *IEEE Transactions on Intelligent Transportation Systems* 12, 2 (2011), 574–582.

[46] Feng You, Ronghui Zhang, Guo Lie, Haiwei Wang, Huiyin Wen, and Jianmin Xu. 2015. Trajectory planning and tracking control for autonomous lane change maneuver based on the cooperative vehicle infrastructure system. *Expert Systems with Applications* 42, 14 (2015), 5932–5946.

[47] Zhiwei Zhu and Qiang Ji. 2004. Real time and non-intrusive driver fatigue monitoring. In *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*. IEEE, 657–662.