Journal of Intelligent Transportation Systems: Technology, Planning, and Operations
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/gits20

# Learning Driver's Behavior to Improve Adaptive Cruise Control

Avi Rosenfeld [a], Zevi Bareket [b], Claudia V. Goldman [c], David J. Leblanc [b] & Omer Tsimhoni [c]

[a] Department of Industrial Engineering, Jerusalem College of Technology, Israel
[b] University of Michigan Transportation Research Institute, USA
[c] General Motors Advanced Technical Center, Israel
Accepted author version posted online: 21 Feb 2014.Published online: 21 Feb 2014.

To cite this article: Avi Rosenfeld, Zevi Bareket, Claudia V. Goldman, David J. Leblanc & Omer Tsimhoni (2014): Learning Driver's Behavior to Improve Adaptive Cruise Control, Journal of Intelligent Transportation Systems: Technology, Planning, and Operations, DOI: 10.1080/15472450.2014.889960

To link to this article: http://dx.doi.org/10.1080/15472450.2014.889960

# Learning Driver's Behavior to Improve Adaptive Cruise Control

[1]Department of Industrial Engineering, Jerusalem College of Technology, Israel

[2]University of Michigan Transportation Research Institute, USA

[3]General Motors Advanced Technical Center, Israel

Email: rosenfa@jct.ac.il, bareket@umich.edu, claudia.goldman@gm.com, leblanc@umich.edu, omer.tsimhoni@gm.com



## 1. Introduction

With the advance of automation and infotainment systems in vehicles, driving can be perceived as a comprehensive interactive activity occurring among human drivers and their vehicles. Classically, these users were considered the supervisors of the automated systems running in the cars. In the last decade, we have seen progress towards more autonomous driving automation, including features such as automatic lane centering, lane keeping, cruise control and adaptive cruise control.

In this paper, we concentrate on an Adaptive Cruise Control function and look into the future by attempting to learn automatically how such a system can adapt its settings to its user and context. Cruise control is a known technology that aids drivers by reducing the burden of longitudinal control of the car manually. This technology controls the vehicle speed once the user sets a desired speed. Cruise control is not only convenient, but it has the potential to improve the flow of traffic (Aerde and Rakha, 1999; Rakha et al., 2001; van Arem et al., 2006; Tapani, 2012), and can be effective in reducing driver fatigue and fuel consumption (Bishop, 2000). In this paper, we focus on a second generation of cruise controls– adaptive cruise control (ACC) as it was used in the Automotive Collision Avoidance System Field Operational Test (ACAS FOT) (Ervin et al., 2005). ACC is designed as a comfort-enhancing system, which is an extension of conventional cruise control (CCC). The ACC system relieves the driver from some of the longitudinal-control tasks by actually controlling speed and headway keeping, but the driver can choose to engage or disengage the ACC at any time. The major difference between ACC and CCC is the use of radar technology to maintain a preset distance between the vehicle with the ACC and other vehicles on the road. This distance is controlled by a “gap” parameter which sets the minimum gap (headway distance) to the vehicle in front of it.

While ACC adds a level of automation to the driving experience, it typically also requires the driver to set and adjust one more parameter, the gap setting, which controls how long of a headway the ACC car will maintain with a car in front of it. Such configurable settings are also quite typical for other driver assistance systems and advanced warning systems. Generally, there is a strong tendency to keep the number and complexity of such user-configurable settings to a minimum. It is therefore likely to allow for only one or two settings, which usually comprise of a few discrete choices. This oversimplification has several downsides. First, it does not allow for fine tuning continuous setting parameters (e.g., choosing a gap of 1.05 seconds out of six discrete settings of 1.0, 1.2, 1.4, 1.6, 1.8, and 2.0 seconds. Second, it results in likely ignoring many second-order parameters (e.g., setting the acceleration at which the vehicle approaches a gap setting). Third, combinations of settings are likely to be lumped together (e.g., a combined setting for longitudinal alerts). Fourth, there is very limited ability to adjust setting combinations to environmental and traffic conditions if only one or two settings are available (e.g., gap setting and maximum speed). Overall, the driver has a limited opportunity to customize their driving experience.

Currently, commercial ACC systems preset the gap value to a default value which can be adjusted by the driver manually based on her driving preferences. We envision that as the number of automated features grows, and as the number of user-preferred settings consequently grows, there will be an unavoidable need to add intelligence to set the default setting selection through context-dependent adaptation and user modeling to improve the driving experience.

This paper presents a learning agent based on three types of data: previous actions that a specific driver chose, road data within the immediate context of a drive, and a pool of other drivers' data. We created general driver profiles within this last set of data based on an extensive database of driving information that had been collected from 96 drivers (Ervin et al., 2005). We used post-processing of data from that study. Our general methodology is that once a new driver is identified we classify this driver as being similar to previously known drivers and set the initial gap value accordingly. We similarly found that using a combination of all types of data allowed us to more accurately learn when the ACC was both engaged and disengaged.

In the next section we generally discuss what types of data the ACC agent could potentially use in its learning component and previous work in personalizing ACC systems. In Section 3 we detail the ACC system and discuss in greater detail Fancher et al.'s driver model (Fancher and Bareket, 1996) that is one key component within the learned model. We also present the general structure of the ACC agent that we propose, and further detail how the learning component, which is the focus of this paper, is formed. In Section 4 we present how the data was collected for creating a general database of drivers' ACC decisions. Section 5 presents the success in creating and implementing this agent based on combining driver demographic data combined and Fancher et al.'s driver model. We report on the success in learning the ACC gap settings and when drivers engage and disengage the ACC. Section 6 concludes.



## 2. Related Work

The main research challenge of this study was to process real world data so as to obtain the most accurate and practical rules from the learning algorithms within the ACC system. The concept of using a group of characteristics to learn people's behavior has long been accepted by the user modeling community. Many recommender systems have been built on the premise that a group of similar characteristics, or a stereotype, exists about a certain set of users (Rich, 1979). Even more similar to our work, Paliouras et al. (Paliouras et al., 1999) suggested creating questionnaires, distributing them, and then creating decision trees to automatically define different groups of users. Similarly, our application assumes that some connection exists between users, which can be learned using machine learning techniques. We propose that this approach be applied to customize settings within an application, here ACC, and not within recommender systems.

Previous works within the last decade did study how to assist drivers in the task of longitudinal control (Naranjo et al., 2003, 2006). Within these approaches, rules were learned manually after having interviewed human drivers. Based on these rules the gap value was adjusted automatically to the conditions of the drive without considering the particular driver in the vehicle. However, we found that individual drivers differ in their driving styles and preferences. Therefore, the goal of this project was to attempt to create an intelligent ACC agent that could learn and react based on this data, such as by potentially setting this longitudinal value autonomously through adjusting its gap value per each driver.

In this work, we considered how machine learning could be used in conjunction with behavior driver models. While many driver models have been proposed (Michon, 1985), many of these models represent general descriptions of the cognitive state of the driver, or elements such as task analysis that do not readily lead themselves to be quantifiable. The general methodology of the learning agent, which is described in greater detail in the next section, is to augment information readily available to the ACC sensors about driving conditions with information gleaned from other drivers and the specific driver. Thus, using quantifiable attributes are critical for the learning agent to be successful.

Thus, in theory any model which categorizes through quantifying different types of driving, or drivers, could be used as part of these databases. For example, Evans et al., studied how personality types, either type A or type B, effects driving behavior among bus drivers. Over 50 years ago, it was hypothesized that certain people, “Type A”, are more prone to coronary heart disease (Friedman and Rosenman, 1959). These people are typified as being in a chronic state to achieve more in less time. Behavioral manifestations of this struggle or those who incessantly struggle to do more in less time, a high level of competitiveness, and a strong desire for high-achievement. In contrast, type B people are typified by having lower stress levels, they work steadily, and are not as competitive (Friedman and Rosenman, 1959; Evans et al., 1987). Even if a person's type could be readily quantified, or self diagnosed, Evans et al. did not find that this division of personality types did not affect driving performance. They found that while Type A drivers brake, pass, and blow their horns more often than Type B drivers, they both exhibited similar magnitudes in increases of physiological markers, such as blood pressure, while driving. Thus, we chose not to include this differentiation between drivers in this study. Nonetheless, it may be interesting to study in future work if this general personality distinction does in fact have any impact on ACC performance.



## 3. Combining ACC and Agent Technologies

The key contribution of this work is the synthesis of ACC and agent technologies to create a more effective system. Within this section we first provide an overview of the ACC hardware within the system. We then present Fancher et al.'s driver model (Fancher and Bareket, 1996) which is a general model which can be used to model driving behavior. Last, we present how an overview of the learning agent that combines Fancher's driver model with traditional learning techniques to form the basis for the improved system.

3.1. **On Overview of the ACC.** Figure 1 shows a picture of a steering wheel with the ACC technology. Note the existence of a "gap" switch on the left side of the figure. The ACAS system consists of the FCW (Forward Collision Warning, which is not pertinent to this paper and therefore not discussed here), and ACC systems. Figure 2 shows a functional schematic of the ACAS system. A closest in-path vehicle (CIPV) is selected from the radar tracks. The CIPV is a "movable" target, that is, one that either is moving or has been observed to move. The range and range-rate from the host vehicle to the CIPV are the primary variables used by the ACC system when controlling headway by modulating the throttle and the brakes. The maximum braking authority of the ACC

ACCEPTED MANUSCRIPT

controller is 0.3g. The ACC controller does not respond to stationary targets. Driver application of the brake always disengages ACC.

The driver vehicle interface (DVI) subsystem uses outputs of the ACAS processors to control the HUD and issues auditory warnings when necessary. The DVI also includes driver controls for the ACC operation, and to adjust brightness and vertical position of the HUD. Figure 3 shows the layout of the head up display. Vehicle speed is displayed in the upper left. System status is displayed on the lower right. Icons in the upper right are used to display information that either a vehicle has been detected or that a cautionary alert applies.



3.2. **Modeling the Driver's Behavior.** We found that a driver model by Fancher et al. (Fancher and Bareket, 1996) allowed us to effectively quantify different types of driving. In developing this measure, their work previously analyzed a group of 36 drivers and their acceptance of adaptive cruise control (ACC). They found that while all drivers enjoyed and accepted the ACC, their behavior could be divided into three types. Each group demonstrated specific driving tendencies which impacted their headway and closing speeds relative to vehicles ahead during manual driving (i.e., without cruise control engaged). In very general terms, these groups were assumed to be: one that is most aggressive, another that is least aggressive, and a third that is in between. Although it is clear that more detailed grouping may exist, and that a different profiling of the drivers' population can be made, for the purpose of this study the characterization analysis was aimed at identifying the above three grouping types. The three driving styles are: 1. Hunters (aggressive drivers who drive faster than most other traffic and use short headways); 2. Gliders (the least aggressive drivers who drive slower than most traffic or commonly have long headways); and 3. Followers (whose headways are near the median headway and usually match the speed of surrounding traffic).

To classify each driver, the following data were computed:

(1) **Average Range (R)** that is the average of all the headway-distance data of the particular driver under the constraints outlined above.

(2) **Average Range-Rate (Rdot)**: that is the average of all the closure-rate data of the particular driver under the constraints outlined above.

(3) **Average Headway Time (Th)** that is the average of all the instantaneous headway times (R(t)/V(t)) of the particular driver under the constraints outlined above.

While the properties **R**, **Rdot**, and **Th** are situation-dependent, and in particular are all directly related to the traffic density, they do nonetheless highlight differences in driving patterns. On a factual level, drivers with have higher **Th** and **R** are further away from the car in front of them than those with lower levels. Thus, one may suppose that these drivers simply were in areas with less traffic. However, as per results from Fancher et al. (Fancher and Bareket, 1996) in previous work, the key is understanding that different drivers react according to these three categories, even within similar traffic situations. Figure 4 highlights this point. It depicts the averages $\overline{Rdot}$ and $\overline{Th}$ of all the drivers on a scatter plot. The average $\overline{Rdot}$ is a measure of whether the driver tends to travel faster ($\overline{Rdot} < 0$) or slower ($\overline{Rdot} > 0$) than the neighboring and preceding vehicles. $\overline{Th}$ is the gap divided by speed and provides a measure of gap in units of time.

The center of the square bounded by the solid black and red lines represents a value of Rdot=0 and Th=1.7. These values were taken based on their empirical observations as a delineator between different types of driving under the same types of conditions. Based on these clusters they divided drivers into three classes:

(1) To the left and below the black line are the Hunters, depicted by circles.

(2) To the right and top of the red line are the Gliders, depicted by dots, and

(3) Those within the rectangle bounded by the red and black lines are the Followers, depicted by ’x’.

In numerical terms, Hunters have headway times (Th) less than about 1.2 seconds and they tend to travel by at least 1 mph faster than neighboring vehicles in the traffic stream. Gliders have headway times greater than 2.2 seconds and they tend to travel by at least 1 mph slower than neighboring vehicles in the traffic stream. The Followers lie within the bounds between Hunters and Gliders. As you can see from this figure, the Gliders, represented by the X points within the square, are typically not closing on the car in front of them and thus have Rdot values near zero. The Hunters, represented by the O's, are typically closing on the car in front them (Rdot being near -0.5 or less). The Gliders, represented by the filled in dots, have both the large headway times (Th), and also are not gaining on the car in front of them (Rdot values of -0.5 and greater).

As we now detail, we found that Fancher et al.'s behavior model (Fancher and Bareket, 1996) was crucial for accurately predicting a driver's ACC preferences. We now detail how we constructed a learning agent that used these driver classes as attribute types (hunter, glider or follower) which were used in addition to other demographic information to better predict a driver's ACC use. This learning was crucial for several aspects of the transportation application we are creating, as we now detail.

3.3. **Agent Structure and Learning Approach.** We now describe the ACC agent at the core of our system for which a schematic is found in Figure 5. In this section we discuss key elements from the agent's structure and its decision making approach. Please note that the agent uses as input three types of information: the ACC's sensors that are autonomously measured by the system at any given time, a database of information of previous driving information of other drivers, and information of the current driver's decisions. The agent must process these inputs to reason about the ACC should be engaged and disengaged, and what parameters the ACC should be set to. Clearly, the agent must not take certain actions, such as engaging the ACC without warning, or even suddenly changing the gap settings within the system, as this may endanger the driver. Thus, the agent must reason about what should be presented to the user, an issue often referred to by agent researchers as adjustable autonomy (Scerri et al., 2002). Specifically, the agent must then address how adjustable autonomy issues must be resolved in deciding when gap settings can be safely set (such as before a driver begins driving), and what and how the system should present information about other ACC changes such as recommendations that the driver should engage or disengage the ACC. These issues must then be applied to the agent's sensors. Note that as per Figure 5, we also allow that the agent integrate explicit feedback by the driver with the system to facilitate better support for a given driver.

This paper focuses on the learning component at the core of the agent. Specifically, we focus on three key tasks: When should the agent recommend the driver to engage the ACC, when should the agent recommend the driver disengage the ACC, and what value(s) should the ACC be set to. The first task focuses on the types of driving conditions that exist before drivers typically engage the ACC. Here the goal is that the agent recommend, when these conditions are met. The second task focuses on when the driver disengages the ACC. Again, the goal is not to have the agent autonomously disengage the ACC, but instead provide a recommendation to the driver. In the third task, focuses on values that control the gap the ACC autonomously maintains from the vehicle in front of it. Current ACC systems allow the user to set the gap value between a finite number of possible values (1–6 in our case) with each number representing the distance in car lengths that the system maintains. Once the car comes within this distance, typically due to traffic or another car cutting in front of the driver, the system brakes the car until the desired gap distance is achieved. Currently, one gap value is set as the default (in our case this value was 6) and the user may change it during his driving as he wishes. One goal of the learning agent is to set this initial value to one that will likely be better than the one current default value that is being used. In theory, the agent could also potentially recommend that the driver adjust the ACC based on monitoring her past driving behavior. While we do not study this problem, the approach for doing so is similar to that used to set the initial ACC value based on the learning approach that we describe below.

The system's learning agent uses the methodology in Figure 5. We have previously explored an approach where machine learning is used in conjunction with generalized cognitive and behavioral models from other disciplines, including experimental economics and psychology. In general, we previously found that merging this synthesis yields one of two general results. In some cases, there

ACCEPTED MANUSCRIPT

is ample data about people's interaction within the system allowing us to form an accurate model of people's behavior using machine learning alone. While in these types of cases machine learning does not provide a more accurate model than the best psychological model, machine learning can be used to confirm the effectiveness of a given model in predicting people's behavior. This can be particularly useful if multiple cognitive models were possible allowing us to judge which model is best without human bias. Additionally, knowing which cognitive model is applicable to a given problem allows us to quickly form an accurate model of people's behavior, even with limited or noisy data (Rosenfeld and Kraus., 2009; Rosenfeld and Kraus, 2012). In contrast, in more complex environments a lack of data makes it unfeasible to elicit an accurate model with machine learning alone. In these types of problems we found that using attributes from cognitive models allowed for significantly more accurate models than models created from machine learning or the cognitive models alone (Rosenfeld and Kraus., 2009; Rosenfeld and Kraus, 2012; Zuckerman et al., 2011).

All of these learning tasks have unique challenges. The first two tasks seem easier as they involve a relatively simple binary task (should the ACC be on or not). However, in both of these learning tasks, we are confronted by the known dataset imbalance problem (Chawla et al., 2002). In many real-world problems, as is the case here, each class is not equally represented. In fact, in the specific case of the ACC engagement task, over 90% of manual driving cases continue their manual driving, and in only a small percentage of cases do people engage the ACC. From a statistical perspective, a classifier could then naively classify all cases as being in the majority case and still have extremely high accuracy. However, because only the "minority" cases are relevant, novel methods are needed to find them.

While several algorithms exist for addressing the data imbalance problem, we specifically focused on the MetaCost algorithm. MetaCost is a general algorithm for making any type of classification learning algorithm cost sensitive, allowing us to stress certain categories more than others. MetaCost has the advantage of working well with any classification algorithm, as it operates by wrapping a cost-minimization procedure around any learning algorithm (Domingos, 1999). We opted to use this algorithm because of its flexibility and the it facilitates in controlling the bias size given to the minority case. We found that by adjusting the cost for mislabeling the small, but important minority class, we were able to effectively learn with high recall when people engage or disengage the ACC. This was done through creating a cost matrix for the learning task. In this cost matrix, the majority case was left with zero cost for misclassifications, while the minority case was given a cost between 0.5 and 0.9 for misclassifying these important minority events. Empirical results found in Section 5 demonstrate the effectiveness of this approach.

The learning challenge for the gap setting was due to the variety of different (6) values. In order to address this challenge, we considered two different types of learning models. The first type of model was a regression model. Regression models operate by statistically predicting the value of a continuous dependant variable from a number of independent variables. In this problem, the goal was to predict the gap value a given driver would select based on the independent variables of the current driving conditions. The second type of model was a discrete model. Specifically, here we learn which of the discrete gap values a driver will likely choose given all possible values given current driving conditions.

Our goal was to use the output of either model to automatically set the gap value. Towards this goal, the second model is seemingly the better choice as its output directly correlates to a value within the system. In contrast, the regression model outputs a decimal value (e.g. 3.5) that must be first rounded to the closest value within the system to be used. However, the advantage of this model is that a mistake between two close values (e.g. 3.5 being close to 3 and 4) is not as mathematically significant as mistakes between two extreme values (e.g. between 2 to 6). In contrast, the discrete decision tree model weighs all types of errors equally. In practice, the regression model will likely be more useful if the user is willing to accept errors between two similar values.

While many machine learning approaches could serve as the base of these models, we intentionally implement the learning agent with decision trees. In contrast, logit or probit regression algorithms could have been used in the first model, and Bayesian networks or neural networks could have been used as alternates within the second model. However, we intentionally used Quinlan's M5 regression decision tree (Quinlan, 1992) and C4.5 discrete decision tree (Quinlan, 1993) algorithms for each of these models. The advantage of using these decision tree approaches did not necessarily lie within the accuracy of these algorithms, as we did find that other algorithms did yield similar results. Instead, the advantage in using these algorithms did lie in their ability to output the exact if-then rules behind the model. This in turn allows the system designer to analyze the agent's rules, an idea we develop in the results section (Section 5).



## 4. Learning from Drivers' Previous Interactions

Referring back to Figure 5, note that one key source of data for the learning agent is a database of information of people's historical driver patterns. One key element of the agent is a general database learning from other people's driver. Data for this database was taken from the Automotive Collision Avoidance System Field Operational Test (ACAS FOT) (Ervin et al., 2005) (see Figure 7).

In the ACAS FOT study a group of 96 drivers was presented with a vehicle fitted with the ACC which they used for a period of 4 weeks. Before entering the study, all drivers had at least a good knowledge of what conventional cruise control (CCC) does. The study's organizers then explained how adaptive cruise control (ACC) is different, including the front sensor and its function in braking the vehicle. They confirmed that the drivers understood that the system operates just like CCC until the sensor detects a vehicle in front. Then the system will slow the car down to maintain the desired (set) headway gap. They were explicitly told that the system will not brake for them to avoid a crash and that they can always override the accelerator by depressing it, and to disengage the system by touching the brake (just like CCC). They stressed they are always in charge. For the first week, the ACC system was not available, allowing drivers to acclimate to their vehicles. If the driver engaged the cruise control during this period, it simply maintained speed just like the conventional system (CCC). For the next three weeks, if the driver chose to engage the cruise control, it functioned as ACC.

In general, three different datasets were considered. The first, and most basic, dataset were objective characteristics that can be studied based on the location of the vehicle itself, e.g., gap to the lead vehicle, vehicle speed, longitudinal acceleration, road type (country, city, or highway), weather (including day or night) and road density (is there traffic). A second dataset added driver characteristics. These properties focus on driver demographics such as age, gender, income level (high, medium, low), and education level (High School, Undergraduate, and Graduate ). The ACAS FOT data consists of a good mixture of these demographics with a 51% male to 49% female split, 31% young (aged 20–30), 31% middle aged (aged 40–50), and 38% older drivers (aged 60–70), and people from a variety of education and socioeconomic levels. The last dataset also used the drivers' observed behavior from the first week to label drivers as hunters, gliders or followers as per Fancher et al.'s previous work (Fancher and Bareket, 1996).

The experimental design of the ACAS FOT was a mixed-factors design in which the between-subjects variables were driver age and gender, and the within-subject variable was the experimental treatment (i.e. ACAS-disabled and ACAS-enabled). The drivers operated the vehicles in an unsupervised manner, simply pursuing their normal trip-taking behavior using the ACAS test vehicle as a substitute for their personal vehicle. Use of the test vehicles by anyone other than the selected individuals was prohibited. The primary emphasis on user selection for the field operation test was to roughly mirror the population of registered drivers, with simple stratification for age and gender. No attempt was made to control for vehicle ownership or household income levels. Thus, although the ACAS FOT participants may not be fully representative of drivers who might purchase such a system, they were selected randomly and represent a wide range of demographic factors.

## 5. Results

In this section we present results for the three previously defined problems: predicting a driver's gap value within the ACC using both discrete and regression models, predicting when a driver will engage the ACC, and predicting when a driver will disengage the ACC. In all three problems we present how the driver type and other demographic information helped improve the model's accuracy. We include an analysis as to how the driver type impacted the drivers' desired headway setting, with the ACC both engaged and disengaged. Additionally, we analyze which attributes were most prominent in this application, how we avoided overfitting, and how we addressed the dataset imbalance problem within this application.

5.1. **Setting the ACC's Gap Value.** Figure 8 presents the accuracy of the decision tree model to learn a driver's preferred gap value in the discrete model. Clearly, adding the demographic data here is crucial, as the model's accuracy drops from over 66% accuracy with this data to less than 37% accuracy without this. As a baseline, we also include the naive classifier, which is based on the most common gap value– here the value of 6, which is also the system's default. Note that the naive model had an accuracy of nearly 27%, far less than other models. The user's type did improve accuracy, as adding this information increased the model's accuracy to near 70%, a significant improvement over the accuracy without the type (a nearly 4% increase of 66.07 to 68.86). In line with our previous work (Rosenfeld and Kraus., 2009), we hypothesized that adding this behavior model yields smaller increases if it can be learned from other attributes within the data. Here, we believed that adding information about drivers' type is less important, as their type was already implicit in other attributes, such as the driver's demographics.

To support this hypothesis, we constructed a discrete decision tree (again C4.5) to learn the driver's type. We found that this value could be learned with over 95% accuracy (95.22%) when learned with the full Reptree ($T_{max}$)– which strongly supports our hypothesis. We present a pruned version of this tree ($T_{Depth}$ = 4) within Figure 9. From an application perspective, we were not shocked to find that a driver's age factored heavy in their driving behavior. This characteristic is factored in actuary's insurance tables, and is a known factor in car insurance premiums (Chiappori and Salanie, 2000). Note that this characteristic was the first level below the root of the tree, demonstrating this quality. However, possibly equally interesting is that we found education, not gender, to be the next most important factor, as it formed the second level within the decision tree. This factor is often not considered by insurance companies (Chiappori and Salanie, 2000) due to several concerns, including privacy concerns, however, this issue may be worth revisiting. Only in the third level did we find the popular characteristic of gender to factor in, but income also weighed in as an equally important factor. Overall, we found that young men or women with only a high school degree tended to be "hunters" or those with extremely aggressive driving habits, college educated women, and people with higher degrees but lower paying jobs tended to be the less aggressive "gliders". Middle aged men with high school degrees, all middle aged people with college degrees, and people with higher degrees but lower paying jobs also typically belonged to the middle "gliders" category. But older women with college degrees, people with low or medium paying jobs with only high school degrees, and all older people with higher degrees tend to be of the least aggressive "follower" type. Naturally, exceptions existed, and this simplified tree is only approximately 75% accurate. Nonetheless a general direction is evident from this tree, and was one that the content experts felt was not overfitted.

Similarly, it was important to find a decision tree that models drivers' gap value that is not overfitted. Note from Table 1 that the model accuracy given all data is nearly 70%. However, while this value is based on the mathematically sound C4.5 algorithm (Quinlan, 1993), the content experts again felt this decision model was overfitted. Note that the total length of this tree was over 1500 lines long, representing tens of pages of if-then statements that the experts thought were unintuitive and overfitted. We then proceeded to reduce the size of the tree to generalize the model, thus, in the opinion of our content experts, preventing this phenomenon. However, as Table 1 demonstrates, reducing the tree size does not improve the model's accuracy, as previous theoretical works suggest (Esposito et al., 1997), but did produce trees that were acceptable to the content experts. Note from Table 1 that raising $T_{Depth}$ yields only marginal increases in the model's accuracy after $T_{Depth} = 4$. In general, we found that the experts were happy with much smaller trees, but those with similar accuracy. For this problem, we display in Figure 10 the resultant tree of $T_{Depth} = 4$ which is only 6% less accurate than the full tree in Table 1. However, for comparison, the full tree produced with the unpruned C4.5 algorithm has a total size of 1313 leaves and branches, while the pruned tree only has a total size of 50 leaves and branches. Thus, from an application perspective, this tree was strongly favored by the experts, even at the expense of a slightly less accurate model. For example, note that in Figure 10 there is no rule for a gap value of 4. This occurred because a decision tree of height 4 did not clearly differentiating it between this and other values. However, this does not necessarily mean that these smaller trees were inferior. In fact, in line with other work (Rosenfeld et al., 2012), the experts thought that adding additional rules made the trees overfitted, one type of the "curse of dimensionality". Overall, the rules themselves are still heavily influenced by the driver type and demographic information, with driver type being the first level of the tree and the second and third levels of the tree again being primarily based on demographics such as age, gender, education, and income level.

Similarly, we were able to create an accurate regression model, the results of which are found in Figure 11. Within these models, correlation values can range from 1.0 (fully positive correlated) to -1.0 (fully negatively correlated), with 0 meaning no correlation. We found a model with both demographic and type data yielded a correlation of 0.78, while without this information the accuracy dropped to 0.75. Using only vehicle specific data yielded a model of only 0.4, and the naive model (here using the average gap value of about 3.5) yielded a value of nearly 0. Again, we found that the type only slightly improved the model's accuracy, as much of this information was already subsumed within the drivers' demographics. The experts again opted for a reduced model, despite the sacrifice of slightly less accuracy.

5.2. **Time Headway Analysis.** We found that the drivers' driving type strongly impacted their desired headway distance to the closest car. Figure 12 shows the mean time driver (per trip per driver) in each time-headway interval for each of the three driver types while driving manually within the first week of the study. When driving manually, the interval at which drivers spend the most time is at about 0.8 to 1.2 seconds, which is true for all driver types. Gliders keep a slightly longer time headway than Hunters and Followers and their time headways are spread more evenly over longer intervals. Note that the ACC radar system has a maximal range of 100 meters. Thus, this graph indicates the average headway only in situations where moderate to heavy traffic exists (e.g. a car was within 100 meters of the driver).

Figure 13 shows the mean time driven (per trip per driver) in each time-headway interval for each of the three driver types. When ACC is engaged, each driver type has a typical gap setting, which corresponds to the particular mode of the time headway. For example, hunters usually keep a mean time headway of 1.0 seconds, which corresponds to the smallest gap value possible. Glider, on the other hand, peak at about 2.0 seconds, which corresponds to the maximum gap value.

A comparison between Figure 12 and Figure 13 reveals several interesting facts: (1) all driver types chose longer time headways when ACC was engaged relative to when they controlled the vehicle manually; (2) Gliders chose the longest time headway showing the largest relative change in time headway; (3) Gliders increased the amount of time spent with ACC engaged, keeping about 2.0 seconds from the vehicle in front of them. Also note that hunter drivers that drove manually often drove with a headway of less than one second, the minimum gap setting in the ACC system. While a gap setting of less than one can theoretically be implemented within the ACC system, it was intentionally not done to allow for a fail-safe where the driver could override the ACC if she felt it was not braking in time to prevent an accident.

Figure 14 shows the relative portion of time spent with ACC or conventional cruise control (CCC) engaged relative to manual driving in the baseline period and in the ACC enabled period. In the baseline period Hunters spent very little time (9%) with CCC engaged. With ACC engaged, Hunters substantially increased their engagement time, spending 39% of their driving time with ACC engaged. Similarly, both Followers and Gliders increased their relative time engaged in ACC. Interestingly, Gliders moved from using CC 27% of the time in the baseline period to using ACC 74% of the time in the ACC engaged period. In fact, while Hunters always spent more time manually controlling their vehicles than with either CCC or ACC engaged, Gliders spent most of their time engaged in ACC rather than being in manual control, which was not the case for Gliders in the baseline condition. From these result it seems that gliders are the most receptive for the ACC technology, while hunters are the least willing to relinquish their control to the ACC. Nonetheless even hunters, as well as other drivers, were more willing to use a ACC system than the conventional cruise control.



5.3. **Predicting when the Driver will Engage and Disengage the ACC.** An additional goal of the learning agent was to focus on two additional learning tasks: to identify when people activate the ACC and when they deactivate it. The goal behind the gap value task was to allow an autonomous agent to set, at least initially, this value within the ACC. However, by understanding when people are more likely to use this product we can hopefully increase its acceptability and use. Similarly, by understanding when people disengage the ACC we can hopefully create new generations of this technology where people will use it longer and not feel compelled to disengage it. In both of these learning tasks, we are confronted by the known dataset imbalance problem (Chawla et al., 2002) first mentioned in Section 3. In this paper, we constructed two models for these two problems based on the same three types of datasets. The first model is a basic discrete decision tree algorithm (C4.5) without any modification. As was the case in the task of setting the gap value, we considered attributes based on the behavior type model, driver demographics (e.g. gender, age, and income level) and the vehicle's characteristics (e.g. gap to the lead vehicle, vehicle speed, and road type). In the second model, we again used the same three datasets, but created a learning bias to find the important minority cases. We specifically focused on the MetaCost algorithm to accomplish this (Domingos, 1999).

Table 2 displays the complete results demonstrating the tradeoff between a model's accuracy and the success in finding the minority cases in the task of predicting when a driver engages or disengages the ACC. The first four rows represent different models created to predict when a person would activate the ACC. The first model is the standard decision tree algorithm C4.5. In addition, we considered three weight biases within the MetaCost algorithm: 0.5, 0.7 and 0.9. Note that raising these weights allows us to give greater weight to the minority case, thus increasing the recall of cases found, but at a cost to the overall accuracy of the model. A cost of 1.0 means that the MetaCost algorithm will assign infinite weight for a given category. Accordingly, the algorithm will always classify all instances as belonging to the minority case. While this will result in full recall for the minority case (1.0), it will come at a cost of a model which overall is very inaccurate. Conversely, a cost of 0.0 for the minority case will result in no bias for these cases, resulting in overall lower recall. Thus, we report on middle values of 0.5, 0.7 and 0.9.

For each of these models we trained four different models: one created with all information, one without the type information but with the demographic information, one without the type and without the demographic information, and a naive model that assumes the majority case– that a person continues driving in manual mode. The accuracy of each of these models is found in the first four rows in Table 2, and the corresponding recall levels for these models are found in the second four rows of the table. Similarly, we also considered the task of predicting when a person turns off the ACC, and trained models based on the same four algorithms with the same four datasets. The results for the accuracy and the recall of these models are found in the last two sets of four rows of Table 2.

Ideally, one would wish for a perfect model: e.g. one with 100% accuracy and recall of all cases. Unfortunately, this is unrealistic, especially in tasks which are prone to variations due to noise. In this domain, the noise comes from two factors: Noise from people's lack of consistency, and noise from unmeasured factors from within the environment. For example, this study did not consider traffic density in lanes to the right or left of the driver– something that clear may impact a driver's decision. Nonetheless, the overall conclusion is that by adding more information, and specifically about a person's demographics, we were able to achieve higher overall accuracies with better recall.

We would like to present the driver a recommendation as to when to engage the ACC. Towards this goal, we wished to set the desired confidence level of the model, as found based on the recall of the minority class, before presenting a recommendation to the user. Figure 15 displays the interplay between the overall model's accuracy and the recall within the minority cases in the task of predicting when a driver engages the ACC. Again, the most desirable result is one in the upper right corner– high accuracy **and** recall. However, as one would expect, and as evident from Table 2, the naive case of continuing without engaging the ACC constitutes over 91% of the cases, but this model will have recall of 0 for the minority case. By modifying the weights within the MetaCost algorithm we are able to get progressively higher recall rates over the basic decision tree algorithm. Also note that the model trained with all information achieves better results than one without the type and demographic information.

Similarly, Figure 16 displays the same interplay between the overall model's accuracy and the success in finding the minority cases in the task of predicting when a driver disengages the ACC. In this task, the naive case assumes that the driver will continue with the ACC constitutes over 86% of the cases, but this model will have recall of 0 for the minority case (see the left side of Figure 16). Note that we were again able to raise the recall within the minority case by creating weight biases of (0.5, 0.7 and 0.9), but again at the expense of a lower overall accuracy. However, as opposed to the engage ACC task, we noticed that the gain from the demographic and type information was not large. In fact, upon inspection of the output trees, we noticed to our surprise that people's decision to disengage the ACC was more dependent on how quickly the ACC slowed the vehicle down, and not on the overall behavior of the driver. Thus, it should be noticed that simply adding attributes is not a panacea for higher accuracy– it only improves accuracy when relevant to the learning task at hand.

Overall, these results suggest that finding attributes beyond the observed data can be critical for accurately modeling a person's behavior. Similar to previous studies that found that other behavioral theories can better predict people's actions (Rosenfeld and Kraus, 2012; Zuckerman et al., 2011), this work found that a driver's preferred gap value could be better predicted by using a model of driving behavior (Fancher and Bareket, 1996). Even if this measure was not readily available, an accurate estimate of this value could be learned based on a driver's demographic data.

Practically, we are studying how either or both of these attributes can be used. The advantage to using the demographic data alone is that ostensibly it can be provided before the driver begins using the car (e.g. in the showroom) and thus can be used to accurately model the driver from the onset. However, people may be reluctant to provide this information due to privacy concerns. Using driver profiling information is relatively difficult to calculate and is based on observed behavior over a period of time (Fancher and Bareket, 1996). Thus, this value cannot be used to initially set values within the ACC. However, this data can be collected without privacy concerns and can be used to further improve the system's accuracy over time.

## 6. Conclusions

Adapting automated processes to better serve humans is a challenging task because it is difficult to predict setting values. Humans are characterized by inconsistent behaviors (or at least seemingly inconsistent), may have difficulties in defining their own preferences, are affected by their emotions, and are affected by the complexity of the problems they face together with the context of these problems. In particular, human drivers also need to react fast enough to road conditions and changes in traffic. This task was particularly challenging as incomplete information was not only inherent about drivers' preferences, but also from the domain itself. For example, this study did not have complete knowledge about traffic in the area of the driver, and even issues such as traffic patterns or weaving behavior from surrounding drivers were not explicitly measured. Nonetheless, we report on the success of how we quickly and accurately learned the ACC gap value given historical data of the many drivers from the ACAS field test data (Ervin et al., 2005). Successful learning of the gap value in this task should serve as a necessary but not sufficient step towards prediction of settings in driving assistance systems and in other systems and situations. As with other user adaptation systems, the tradeoff between the added value and potential inaccuracy of the prediction system will determine its usefulness and acceptance.

We empirically studied two types of learning models: regression and discrete models. By combining the driver type with other data, we achieved a prediction accuracy of nearly 70% within the discrete model (Figure 8) and a correlation of 0.78 within the regression model (Figure 8). However, when we used only the driver type information and removed the demographic information these models dropped to an accuracy of 46% and 0.55 respectively. These experiments emphasize the need to construct models which not only consider driving data such as the car's speed, road condition, and weather, but also include driver demographic information and a behavior model about the driver's type (Fancher and Bareket, 1996). These results stress the fact that drivers may be very different from each other and previous approaches that set the gap value similarly for all drivers (Naranjo et al., 2003, 2006) are less effective. Therefore, driver characterization is essential for adapting automated systems in the vehicle. These differences among humans are made more salient when trying to learn when users engage or disengage from an automated system such as the ACC. Reactions could be very different, teaching us also about the tendencies of users towards automation.

We present solutions for two practical challenges in applying learning algorithms to this challenging domain: preventing overfitted models, and creating effective models in cases where a strong majority category existed but the important events were in the minority category. We address the overfitting issue by creating simplified decision trees, and present an analysis of the if-then rules produced by these trees. We use the MetaCost algorithm (Domingos, 1999) to learn from unbalanced data sets. We present extensive results details of this application and how these algorithms were used within this challenging transportation domain.

One way of building machines that could interact successfully with humans is by adapting their automated processes to their users. In order to do so, these machines need to consider characteristics of human behaviors, including for example: inconsistent behaviors, having difficulties in defining the user's own preferences, emotional influences, and problem complexity and context effects. Therefore, adapting automated processes is a challenging task. By understanding the current state of acceptance of automated systems and learning about differences among human users, we can improve the next generations of adaptive automated systems adjusted to their particular human users.

One of the larger goals of this paper is to encourage people who build applications to consider incorporating data from external measures, such as psychological or behavioral models. As was true in other domains we studied (Rosenfeld and Kraus, 2012; Zuckerman et al., 2011), modeling user behavior based on cognitive models alone, such as the driver type possible in this domain (Fancher and Bareket, 1996), is not sufficient. Instead, we advocate for synthesizing data gleaned from behavioral models in conjunction with observed domain data, something that we believe can be effective in many other domains as well.



## 7. Acknowledgements

This work was supported in part by JCT grant #963. We thank the anonymous reviewers for their helpful comments and Hadassah Rosenfeld for her help with graphic design work.

## References

Aerde, M. V. and Rakha, H. (1999). A framework for the evaluation of system safety benefits of intelligent cruise control systems. *ITS Journal - Intelligent Transportation Systems Journal*, 5(2):163–189.

Bishop, R. (2000). Intelligent vehicle applications worldwide. *IEEE Intelligent Systems*, 15(1):78–81.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res. (JAIR)*, 16:321–357.

Chiappori, P.-A. and Salanie, B. (2000). Testing for asymmetric information in insurance markets. *Journal of Political Economy*, 108(1):56–78.

Domingos, P. (1999). Metacost: a general method for making classifiers cost-sensitive. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 155–164, New York, NY, USA. ACM Press.

Ervin, R., Sayer, J., LeBlanc, D., Bogard, S., Mefford, M., Hagan, M., Bareket, Z., and Winkler, C. (2005). Automotive collision avoidance system (acas) field operational test methodology and results, (us dot hs 809 901). washington, dc: Department of transportation.

Esposito, F., Malerba, D., and Semeraro, G. (1997). A comparative analysis of methods for pruning decision trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:476–491.

Evans, G. W., Palsane, M. N., and Carrere, S. (1987). Type a behavior and occupational stress: a cross-cultural study of blue-collar workers. *J Pers Soc Psychol*, 52(5):1002–7.

Fancher, P. and Bareket, Z. (1996). A comparison of manual versus automatic control of headway as a function of driver characteristics. *3rd Annual World Congress on Intelligent Transport Systems*.

Friedman, M. and Rosenman, R. H. (1959). Association of specific overt behavior pattern with blood and cardiovascular findings: Blood cholesterol level, blood clotting time, incidence of arcus senilis, and clinical coronary artery disease. *Journal of the American Medical Association*, 169(12):1286–1296.

Michon, J. (1985). A critical view of driver behavior models: What do we know, what should we do? *Human Behavior and Traffic Safety*, 28(2):485–524.

Naranjo, J. E., Gonzalez, C., Garcia, R., and de Pedro, T. (2006). ACC+Stop&go maneuvers with throttle and brake fuzzy control. *Intelligent Transportation Systems, IEEE Transactions on*, 7(2):213–225.

Naranjo, J. E., Gonzalez, C., Reviejo, J., Garcia, R., and de Pedro, T. (2003). Adaptive fuzzy control for inter-vehicle gap keeping. *Intelligent Transportation Systems, IEEE Transactions on*, 4(3):132–142.

Paliouras, G., Karkaletsis, V., Papatheodorou, C., and Spyropoulos, C. D. (1999). Exploiting learning techniques for the acquisition of user stereotypes and communities. In *UM99 User Modeling: Proceedings of the Seventh International Conference*, pages 169–178.

Quinlan, J. R. (1992). Learning with continuous classes. In *5th Australian Joint Conference on Artificial Intelligence*, pages 343–348.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning – Morgan Kaufmann Series in Machine Learning*.

Rakha, H., Hankey, J., Patterson, A., and Aerde, M. V. (2001). Field evaluation of safety impacts of adaptive cruise control. *Journal of Intelligent Transportation Systems*, 6(3):225–259.

Rich, E. (1979). User modeling via stereotypes. *Cognitive Science*, 3:329–354.

Rosenfeld, A. and Kraus., S. (2009). Modeling agents through bounded rationality theories. In *IJCAI-09*, pages 264–271.

Rosenfeld, A. and Kraus, S. (2012). Modeling agents based on aspiration adaptation theory. *Autonomous Agents and Multi-Agent Systems*, 24(2):221–254.

Rosenfeld, A., Zuckerman, I., Azaria, A., and Kraus, S. (2012). Combining psychological models with machine learning to better predict people's decisions. *Synthese*, 189(1):81–93.

Scerri, P., Pynadath, D. V., and Tambe, M. (2002). Towards adjustable autonomy for the real world. *J. Artif. Intell. Res. (JAIR)*, 17:171–228.

Tapani, A. (2012). Vehicle trajectory effects of adaptive cruise control. *Journal of Intelligent Transportation Systems*, 16(1):36–44.



ACCEPTED MANUSCRIPT

van Arem, B., van Driel, C. J. G., and Visser, R. (2006). The impact of cooperative adaptive cruise control on traffic-flow characteristics. *IEEE Transactions on Intelligent Transportation Systems*, 7(4):429–436.

Zuckerman, I., Kraus, S., and Rosenschein, J. S. (2011). Using focal point learning to improve human-machine tacit coordination. *Autonomous Agents and Multi-Agent Systems*, 22(2):289–316.

| $T_{Depth}$ | Accuracy [%] |
|---|---|
| 2 | 47.55% |
| 3 | 56.41% |
| 4 | 62.43% |
| 5 | 65.46% |
| 6 | 67.51% |
| 7 | 68.50% |

TABLE 1. Analyzing the tradeoff between the model's accuracy and the height of the tree $T_{Depth}$.

| ACC On – Accuracy | All Info | Without Type | Without Demographics | Naive |
|---|---|---|---|---|
| C4.5 | 92.67 | 92.32 | 91.22 | 91.27 |
| MetaCost 0.5 | 92.42 | 91.97 | 90.97 | 91.27 |
| MetaCost 0.7 | 91.93 | 91.38 | 90.37 | 91.27 |
| MetCost 0.9 | 87.99 | 86.60 | 77.12 | 91.27 |
| ACC On – Recall | All Info | Without Type | Without Demographics | Naive |
| C4.5 | 0.35 | 0.32 | 0.07 | 0 |
| MetaCost 0.5 | 0.40 | 0.36 | 0.13 | 0 |
| MetaCost 0.7 | 0.45 | 0.42 | 0.18 | 0 |
| MetCost 0.9 | 0.63 | 0.61 | 0.51 | 0 |
| ACC Off – Accuracy | All Info | Without Type | Without Demographics | Naive |
| C4.5 | 88.71 | 88.64 | 88.42 | 86.37 |
| MetaCost 0.5 | 88.59 | 88.55 | 88.14 | 86.37 |
| MetaCost 0.7 | 87.68 | 87.49 | 87.31 | 86.37 |
| MetCost 0.9 | 82.03 | 82.23 | 81.15 | 86.37 |
| ACC Off – Recall | All Info | Without Type | Without Demographics | Naive |
| C4.5 | 0.35 | 0.32 | 0.07 | 0 |
| MetaCost 0.5 | 0.40 | 0.36 | 0.13 | 0 |
| MetaCost 0.7 | 0.45 | 0.42 | 0.18 | 0 |
| MetCost 0.9 | 0.63 | 0.61 | 0.51 | 0 |

TABLE 2. Analyzing the tradeoff between overall model accuracy (first four lines) and recall of the minority cases (second four lines) in both the task of when people turn the ACC on (top) and off (bottom).

ACCEPTED MANUSCRIPT

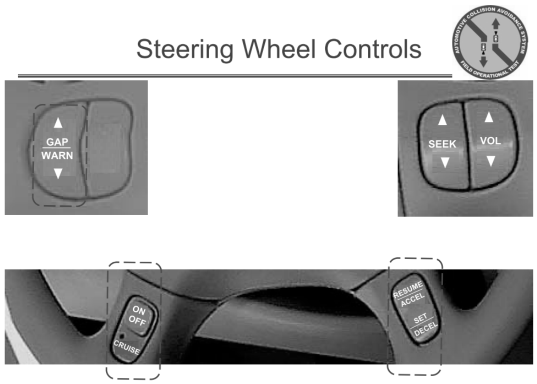


FIGURE 1. A steering wheel fitted with ACC technology.

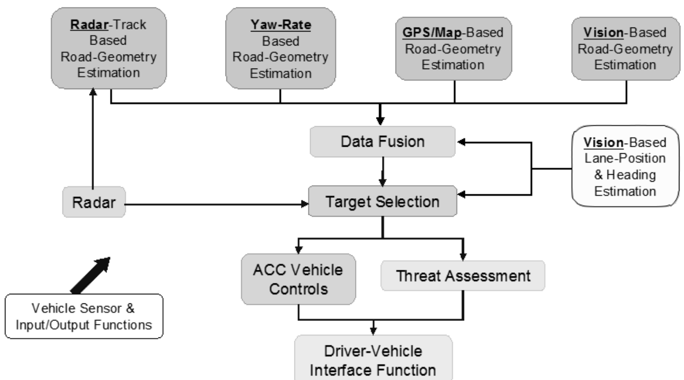


FIGURE 2. ACAS functional schematic

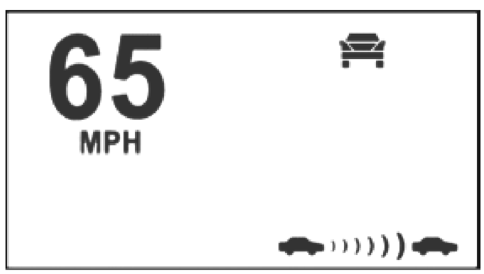


FIGURE 3. Head up display configuration

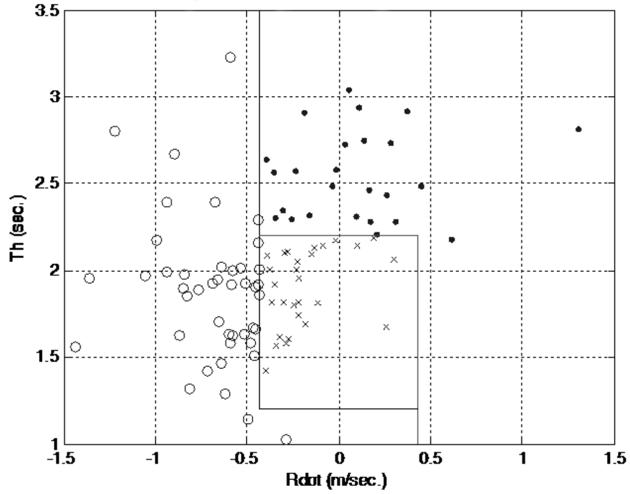


FIGURE 4. Classification of drivers based on their average headway times versus their average speeds

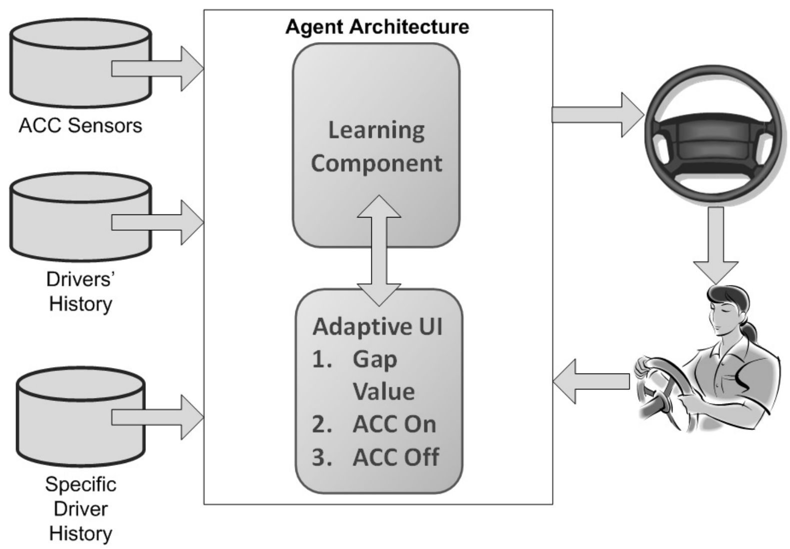


FIGURE 5. The ACC's Agent Model

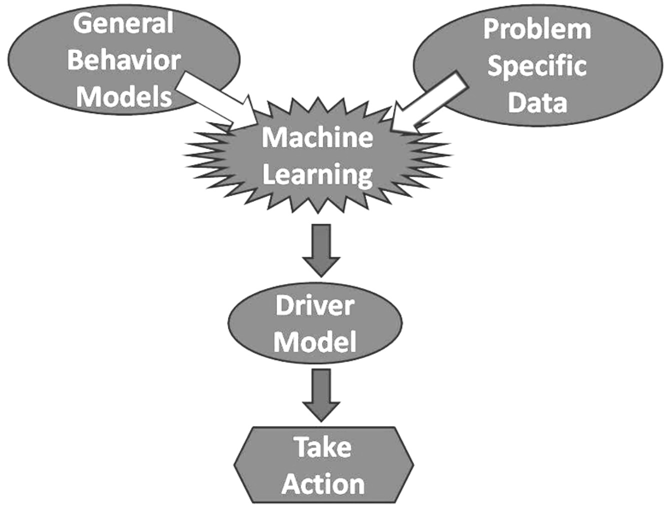


FIGURE 6. A high-level overview of the methodology of combining cognitive models with machine learning.

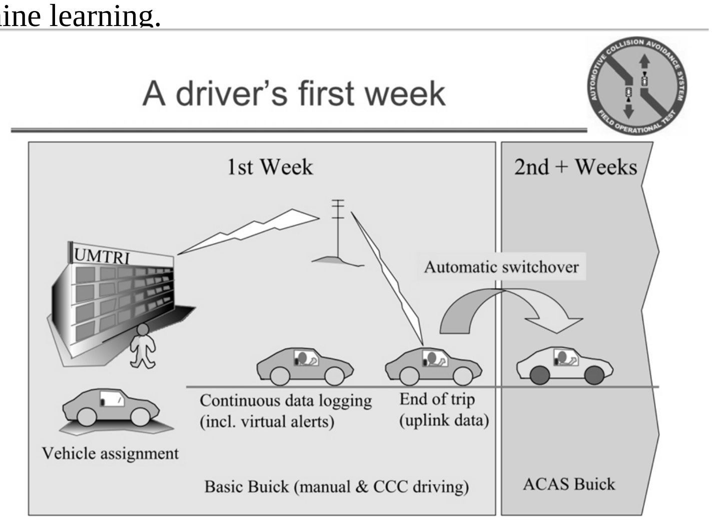


FIGURE 7. An overview of the data collection process within ACAS FOT study.

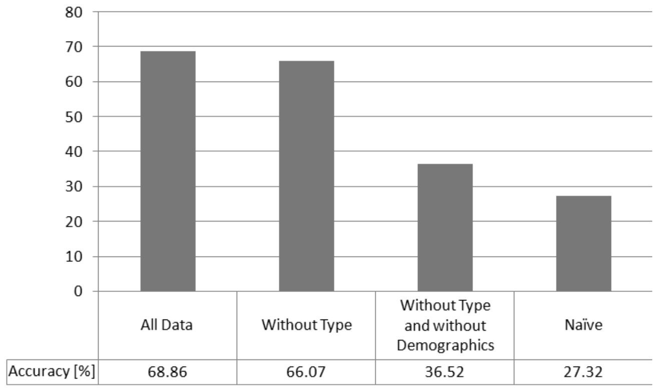


FIGURE 8. The importance of driver type and demographics in predicting the gap value within the ACC for a discrete decision tree model.

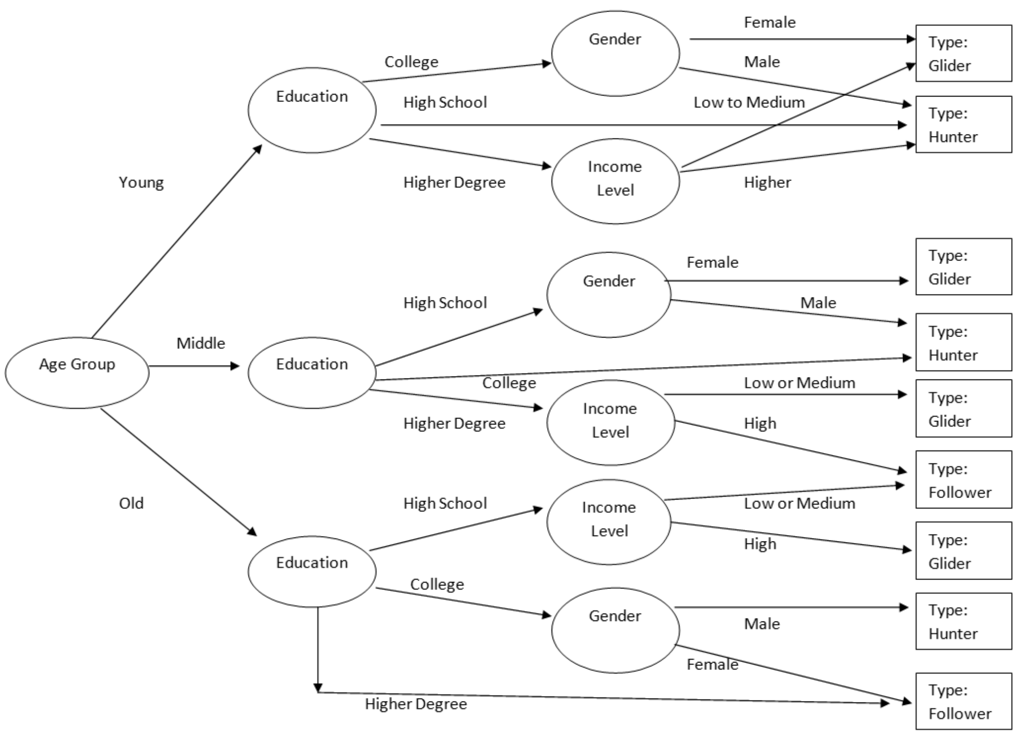


FIGURE 9. The decision tree for learning a driver's "type".

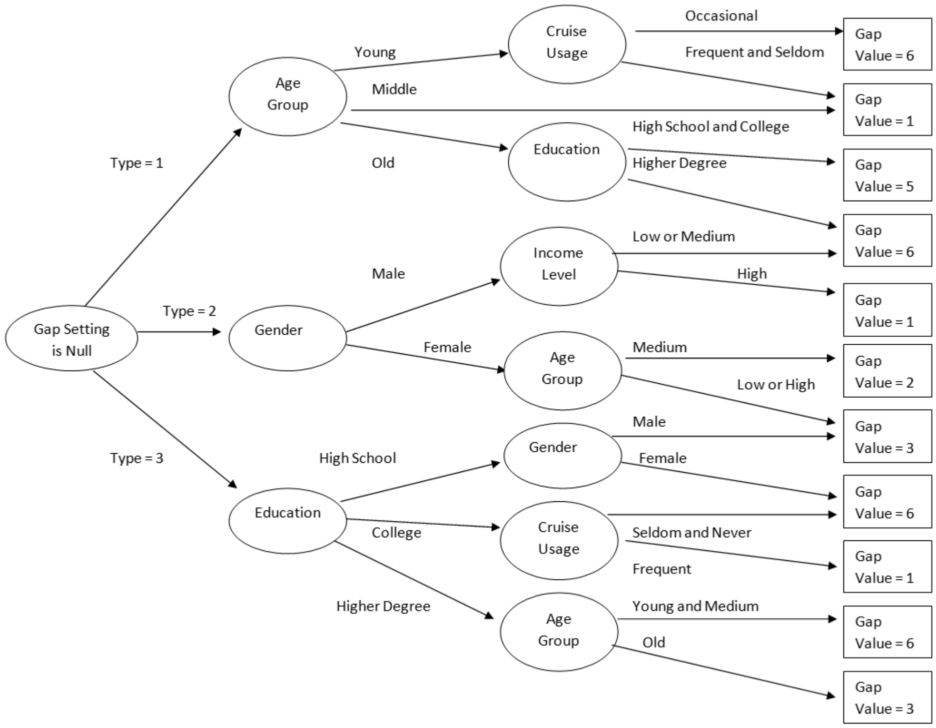


FIGURE 10. The decision tree for learning the ACC's Gap Value for $T_{Depth}$ =4.

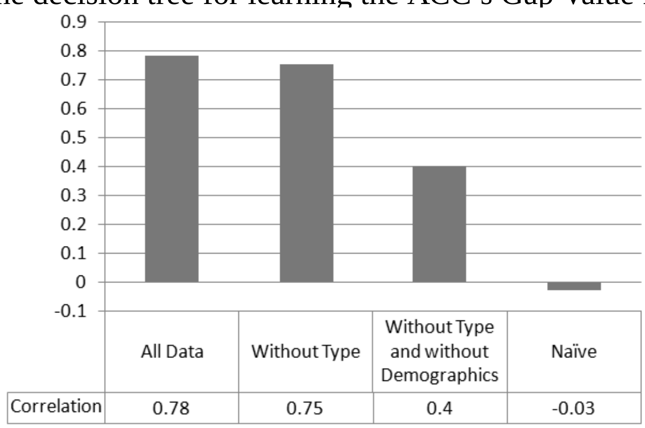


FIGURE 11. The importance of driver type and demographics in predicting the gap value within the ACC for a regression model.

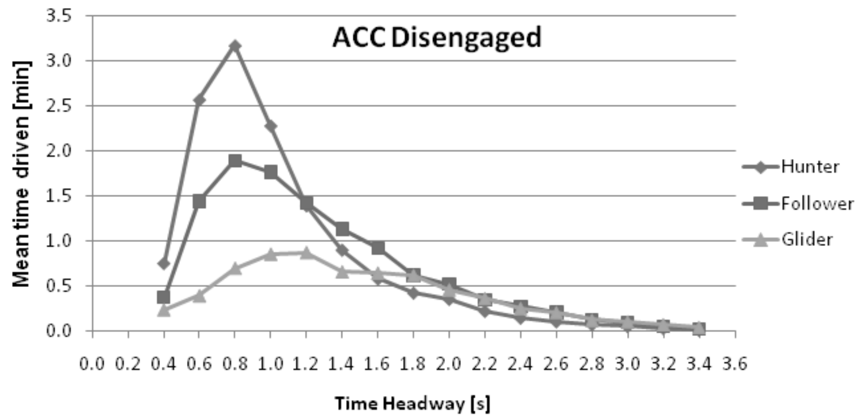


FIGURE 12. The importance of driver type and demographics in predicting the gap value within the ACC for a regression model.

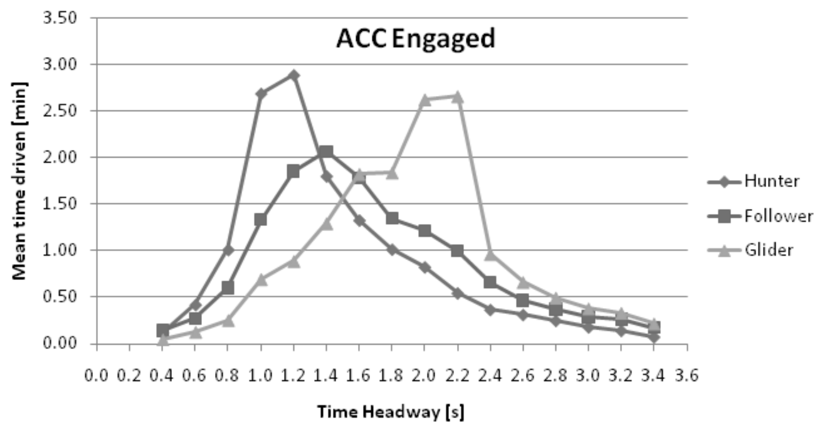


FIGURE 13. The importance of driver type and demographics in predicting the gap value within the ACC for a regression model.

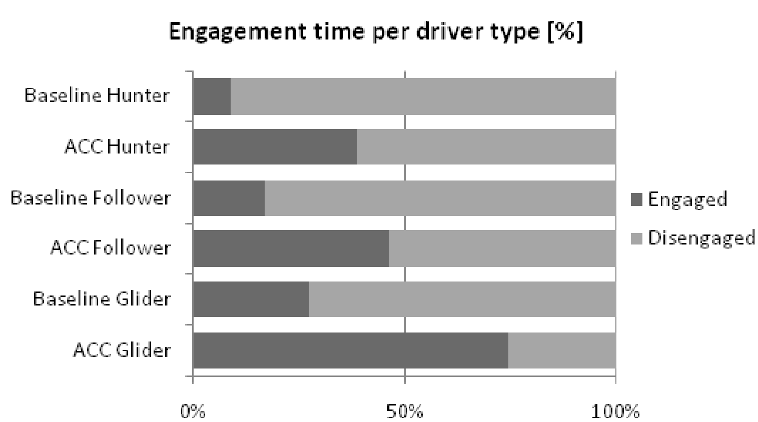


FIGURE 14. The importance of driver type and demographics in predicting the gap value within the ACC for a regression model.



ACCEPTED MANUSCRIPT

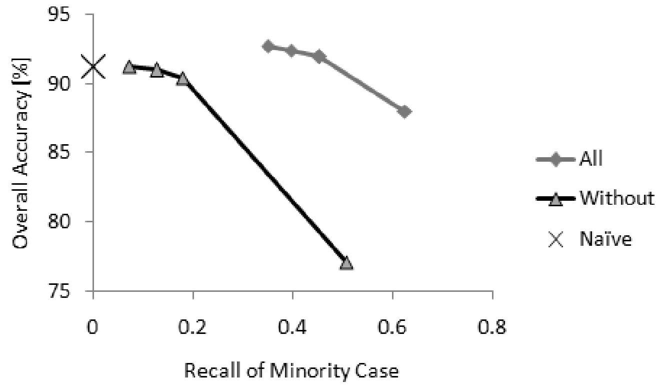


FIGURE 15. Comparing the overall model accuracy and recall for cases for engaging the ACC


FIGURE 16. Comparing the overall model accuracy and recall for cases for disengaging the ACC