
Comparing Support Vector Machines (SVMs) and Bayesian Networks (BNs) in detecting driver cognitive distraction using eye movements

Yulan Liang and John D. Lee*

The Department of Mechanical and Industrial Engineering
The University of Iowa
Iowa City, IA 52242
USA
{yulliang, jdlee}@engineering.uiowa.edu

Abstract

Driver distraction is an important and growing safety concern as information technologies, such as navigation systems and internet-content services, have become increasingly common in vehicles. To allow people to benefit from these technologies without compromising safety, an adaptive In-Vehicle Information System (IVIS) is needed. Such systems can manage drivers workload and mitigate distraction by monitoring and responding to driver states and roadway conditions. For the system to be effective, however, it is critical that driver distraction can be identified accurately in real time. This chapter discusses approaches to identify driver cognitive distraction. Eye movements and driving performance were chosen as promising indicators. A robust data fusion system using data mining techniques was proposed to integrate these indicators to detect when a driver was distracted. The objective of this chapter is to compare two data mining methods for identifying driver cognitive distraction using eye movements and driving performance, Support Vector Machines (SVMs) and Bayesian Networks (BNs), respectively. We trained and tested Dynamic BN (DBN), Static BN (SBN), and SVM models with experimental data and assessed model performance using testing accuracy and two single-detection-theory measures. The DBN method, which considers behavior change over time, produced the most accurate and sensitive models. The DBNs and SVM models were more accurate and had higher hit rate than the SBN models. These results indicate that both sequential changes in eye movements and driving performance are important predictors of drivers cognitive distraction. If time-dependent relationships are ignored the SVM method has advantages over the SBN method because of its computational ease and flexible parameter-choosing strategy. Thus, a hybrid method combining the DBN and SVM methods may create models that perform better than any of three types of models that have been developed to date.

Driver distraction is an important safety problem [28]. The results of a study that tracked 100 vehicles for one year indicated that nearly 80% of crashes and 65% of near-crashes involved some form of driver inattention within three seconds of the event. The most common form of inattention included secondary tasks, driving-related inattention, fatigue, and combinations of these [14]. In-vehicle information systems (IVIS), such as navigation systems and internet services, introduce various secondary tasks into the driving environment that can increase crash risk [1, 27]. Thus, in future it would be very beneficial if IVIS could monitor driver distraction so that the system could adapt and mitigate the distraction. To not disturb driving, non-intrusive and real-time monitoring of distraction is essential.

This chapter begins to address this issue by describing techniques that draw upon the data from a video-based eye tracking system to estimate the level of drivers cognitive distraction in real time. First, three types of distractions and the detection techniques are briefly described. Then, the chapter focuses on the cognitive distraction and brings out a general procedure for implementing a detection system for such distraction. Data mining methods are proposed to be promising techniques to infer a drivers cognitive state from their behavior. Next, Support Vector Machines (SVMs) and Bayesian Networks (BNs) are used and compared in this application. Finally, several issues associated with the detection of cognitive distractions are discussed.

1 Types of distraction

Three major types of distraction have been widely studied: visual, manual, and cognitive [28]. These distractions deflect drivers visual attention, manual operation, and cognitive resources away from the driving control task, respectively, and result in the degradation of driving performance and even cause fatal accidents. Visual distraction and manual distraction can be directly observed through the external behaviors of drivers, such as glancing at billboards or releasing the steering wheel to adjust the radio. Visual distraction usually coexists with manual distraction because visual cues provide necessary feedback when people perform manual tasks. Visual and manual distractions interrupt continuous visual perception and manual operation essential for driving and results in the absence of visual attention on safety-critical events. In the 100-vehicle study, visual inattention contributed to 93% of rear-end-striking crashes. Interestingly, in eighty-six percent of the rear-end-striking crashes, the headway at the onset of the event the led to the crash was greater than 2.0 seconds [14]. These facts show that visual distraction dramatically increases rear-end-striking crashes because two seconds are long enough for an attentive driver to avoid a collision. The degree of visual distraction is proportional to the eccentricity of visual-manual tasks to the normal line of sight [16].

Unlike visual and manual distraction, cognitive distraction is internal and impossible to observe from external behavior. With visual distraction it is possible to detect when the eyes are off the road, but with cognitive distraction there is no direct indicator as to when the mind is off the road. Nonetheless, the effects of cognitive distraction on driving performance are as negative as those of visual distraction. A meta-analysis of twenty-three studies found that cognitive distraction delayed driver response to hazards [10]. For example, drivers reacted more slowly to brake events [16, 17] and missed more traffic signals [32] when they performed mental tasks while driving, such as using auditory e-mail systems, performing math calculation, or holding hand-free cell-phone-conversations.

To identify how visual distraction delays reaction time several researchers have created predictive models that quantify the risks associated with visual or manual distraction from drivers glance behavior. It was found that frequent glances to a peripheral display caused drivers to respond slowly to breaking vehicles ahead of them. The reaction time, the time from the time the lead vehicle began to brake until the driver released the

accelerator, could be predicted by the proportion of off-road glances in this reaction period using a linear equation: $(accelerator - releasereactiontime) = 1.654 + 1.581(off - roadproportion)$ [38]. This relationship accounts for 50% of reaction-time variance.

Another study took historic performance into account. It used a linear function of current glance duration away from road, β_1 , and total glance duration away from road during the last three seconds, β_2 , to calculate warning threshold, γ , as: $\gamma = \alpha\beta_1 + (1 - \alpha)\beta_2$. γ influenced the frequency of alarms for reminding drivers when they were too engaged in a visual-manual task, and α presented the weights of the two glance durations on γ [7]. Using this equation, those drivers who had been identified as risky drivers received more warnings per minute than non-risky drivers for a broad range of α and γ . These studies found diagnostic measures, such as frequency and duration of off-road glances, and used these measures to predict the degree to which visual and manual distractions delayed drivers reaction time to critical roadway events. At the same time, these predictive models can be used to monitor drivers visual and manual distraction non-intrusively in real time because remote eye tracker cameras can collect and output eye movement data without disturbing driving.

However, identifying cognitive distraction is much more complex and less straightforward than visual and manual distraction. There is no clearly diagnostic predictor for cognitive distraction. In controlled situations, four categories of measures are used to assess workload and cognitive distraction: subjective measures, secondary task performance, primary task performance, and physiological measures [37, 39].

Subjective measures and secondary task performance can not be used to identify cognitive distraction for future IVIS. Commonly-used subjective measures include the NASA Task Load Index (NASA-TLX) and the subjective workload assessment technique (SWAT). Collecting subjective measures disturbs normal driving and cannot provide an unobtrusive real-time indicator. The secondary task method for assessing workload would require the driver to perform a task in addition to driving and whatever interaction he or she might have with a in-vehicle system and so is clearly inappropriate for measuring distraction as an unobtrusive real-time indicator of distraction.

Other measures, such as primary task performance and physiological measures, present promising predictors for real-time estimation of cognitive distraction. The primary task refers to driving control task. The commonly-used driving performance measures include lane position variability, steering error, speed variability, and so on. These measures can be collected non-intrusively using driving simulators or sensors on instrumented vehicles in real time. Physiological measures, such as heart-rate variability, pupil diameter, eye movements, represent promising sources of information regarding the drivers state. Although monitoring heart rate and pupil diameter in vehicles is difficult due to sensor limits, it is feasible to track eye movements of drivers in real time with advanced eye tracking systems.

Eye movement patterns change with different levels of cognitive distraction. Cognitive distraction disrupts the allocation of visual attention across the driving scene and impairs information processing. Recarte and Nunes [25] found that increased cognitive load was associated with longer fixations, gaze concentration in the center of the driving scene, and less frequent glances at mirrors and at the speedometer. Cognitive distraction impaired the ability of drivers to detect targets across the entire visual scene [26, 35], and reduced implicit perceptual memory and explicit recognition memory for items that drivers fixated [31]. One study that systematically examined the sensitivity of various eye movement measures to the complexity of in-vehicle cognitive tasks found that standard deviation of gaze was the most sensitive indicator for the level of complexity [35].

Of the four categories of potential measures of cognitive distraction, eye movements and driving performance are the most suitable measures for estimating cognitive distraction [20, 21, 39]. Cognitive distraction presents a great risk in driving because it delays drivers response to critical events and is more difficult to de-

text than visual distraction. However, recent developments suggest eye movements and driving performance offer a promising basis for detecting cognitive distraction.

Although cognitive, visual, and manual distractions are described separately they coexist in most situations. For example, when entering an address into a GPS while driving, drivers need to recall the address and then glance at the system to enter it. This leads to both a visual and a cognitive distraction. Ultimately algorithms that detect visual distraction and cognitive distraction will need to work together to provide comprehensive prediction of driver distraction. However, the balance of this chapter focuses on the challenging task of estimating cognitive distraction using eye movements and driving performance measures.

2 The process of identifying cognitive distraction

Detecting cognitive distraction is complex procedure and requires a robust data fusion system. Unlike visual and manual distraction, the challenge of detecting cognitive distraction is to integrate multiple data streams, including eye movements and driving performance, in a logical manner to infer the drivers cognitive state. One way to address this challenge is by using data fusion. Data fusion systems can align data sets, correlate relative variables, and combine the data to make detection or classification decision [36]. One benefit of using a data fusion perspective to detect cognitive distraction is that data fusion can occur at different abstract levels. For instance, sensor data are aggregated to measure driver's performance at the most concrete level, and then these performance measures can be used to characterize driver's behavior at higher levels of abstraction. This hierarchical structure can logically organize the data and inferences and reduce parameter space in detection procedure. The fusion systems also can continuously refine the estimates made at each level across time, which enables a real-time estimation of cognitive distraction.

To implement a data fusion system, there are two general approaches: top-down and bottom-up. The top-down approach identifies the targets based on the known characteristics, such as shape and kinematic behavior. In the detection of cognitive distraction, the top-down approach is presented by using drivers' behavioral response of people under high levels of cognitive load that reflect existing theories of human cognition, such as Multiply Resource Theory [37] and ACT-R [29]. The limitation of the top-down approach is that it is impossible to implement data fusion without complete understandings of the underlying process—something that is lacking in the area of driver distraction.

The bottom-up approach overcomes this limitation and uses data mining methods to extract characteristics of the targets from data directly. Data mining includes a broad range of approaches that can search large volumes of data for unknown patterns, using techniques such as decision trees, evolutionary algorithms, support vector machines, and Bayesian networks. These methods are associated with multiple disciplines (e.g., statistics, information retrieval, machine learning, and pattern recognition) and have been successfully applied in business and health care domains [3, 33]. In driving domain, decision tree, Support Vector Machines (SVMs), and Dynamic Bayesian Networks (DBNs) have successfully captured the differences in behavior between when people drive normally and when distracted and produced promising results in detecting cognitive distraction [20, 21, 39].

The strategies of constructing data fusion systems include using the top-down approach alone, the bottom-up approach alone, or the mixed approach that combines top-down and bottom-up. The choice of the strategies depends on the availability of domain knowledge, as shown in Table 1. When the targets are understood very well, the data fusion system can be constructed using only the top-down approach. Currently, most data fusion systems use this strategy. Nevertheless, the lack of domain knowledge presents an important constraint of this top-down-alone strategy in some domains, such as the detection of cognitive distraction. The bottom-up-alone and mixed strategies overcome the limitation. Oliver and Horvitz [24]

		Data fusion strategies		
		Top-down approach	Mixed approach	Bottom-up approach
Domain knowledge	All available	√	√	√
	Partially available	-	√	√
	Not available	-	-	√

Table 1. Matrix of data fusion strategies and the availability of domain knowledge

have demonstrated the effectiveness of these two strategies. They successfully used Hidden Markov Models (HMMs) and DBNs to construct the layered data fusion system for recognizing office activities by learning from sound and video data.

Detecting cognitive distraction requires a bottom-up data mining strategy because the effects of cognitive demand on driving have not been clearly understood. Although some theories of human cognition can help explain drivers' behavior, most theories only aim to describe, rather than predict, human performance and can not be used to detect cognitive distraction. Some theories, like ACT-R, represent promising approaches that are beginning to make predictions regarding distraction and driver behavior. On the other hand, various data mining methods have been used to detect cognitive distraction. Zhang et al. [39] used a decision tree to estimate driver cognitive workload from glances and driving performance. In two other studies, Support Vector Machines (SVMs) and Bayesian Networks (BNs), successfully identified the presence of cognitive distraction from eye movements and driving performance [21, 20]. Thus, the strategies using bottom-up and mixed approaches are suitable for data fusion to detect cognitive distraction.

The procedure of detecting driver cognitive distraction can be formulized in two sequential stages, feature refinement and state refinement, as shown in Figure 2. Feature refinement uses the top-down approach and transforms sensor data (such as eye and driving performance raw data) into performance measures based on an existing understanding of what measures may be most sensitive to distraction. The sensor data are collected at a high frequency (e.g. 60Hz) and include eye tracking systems and measures of vehicle speed and driver steering inputs. Feature refinement transforms raw data into eye movements described as fixations, saccades, and smooth pursuits according to the speed and dispersion characteristics of these movements. Various eye movement measures (such as fixation duration and saccade distance) are then calculated to describe drivers' scanning activity. Indicators of cognitive distraction, such as the standard deviation of gaze, are used as inputs for the next stage. Next, state refinement fuses these measures to infer a driver's cognitive state. In this stage, data mining methods are applied to train detection models from the data.

3 Data mining techniques to assess cognitive state

Different data mining methods produce different models. Numerical and graphical models represent two of the most promising classes of models for assessing cognitive state. Typical numerical models include SVMs, linear regression, and polynomial fitting, and typical graphical models include decision trees, neural networks, BNs. Each of these classes of models have advantages in detecting cognitive distraction. Most numerical models have mature techniques for training and testing, and have fewer computational difficulties compared to graphical models. Therefore, for detecting distraction in real time, numerical models usually cause few computational delays compared to graphical models. Nonetheless, graphical models explicitly represent the relationships between drivers' cognitive state and performance, which helps summarize knowledge from resulted models. SVMs and BNs are the representatives of numerical and graphical models, respectively.

3.1 Support Vector Machines (SVMs)

Originally proposed by Vapnik [34], Support Vector Machines (SVMs) are based on the statistical learning theory and can be used for pattern classification and inference of non-linear relationships between variables [6, 34]. This method presents several advantages when estimating cognitive distraction. First, the relationship between human cognition and visual behavior can seldom be represented by a linear model. With various kernel functions, SVMs can generate both linear and nonlinear models with equal efficiency [2]. Second, SVMs can extract information from noisy data [4] and avoid overfitting by minimizing the upper bound of the generalization error [2] to produce more robust models compared to those that minimize the mean square error. SVMs have outperformed the linear logistic method in detecting cognitive distraction [21]. Nonetheless, SVMs can not explicitly present the relationships learned from data.

3.2 Bayesian Networks (BNs)

Bayesian Networks (BNs) represents a graphical approach that represents conditional dependencies between random variables. A BN includes circular nodes depicting random variables, and arrows depicting conditional relationships. As shown in the left graph in Figure 3.2, the arrow between variable nodes H and S indicates that S is independent of any other variable given H . BNs are either Static (SBNs) or Dynamic (DBNs). SBNs (Figure 3.2, left) describe the situation that does not change in time. DBNs (Figure 3.2, right) connect two identical SBNs at successive time steps and model a time-series of events according to a Markov process. The state of a variable at time t depends on either variables at time t or time $(t - 1)$ or both. For example, the state of S_t depends on both S_{t-1} and H_t .

Like SVMs, BNs can model complex, non-linear relationships. However, BNs aim to identify relationships between variables, and to use these relationships to generate model prediction. BNs can explicitly present the relationships learned from data. As a consequence, studying trained BN helps identify cause-effect links between variables, and the hierarchical structure of BN provides a systematic representation of these relationships. BNs are applicable to human-behavior modeling and have been used to detect affective state [19], fatigue [13], lane change intent during driving [15], pilot workload [9], and driver cognitive distraction [20]. One disadvantage of BNs is that they become computationally inefficient when models have a large number of variables (i.e., 20). Another disadvantage is that training techniques for BN are less robust than those available for SVMs.

4 Comparisons of SVMs and BNs

Although both SVMs and BNs have been successfully applied to detect cognitive distraction, it is not clear which methods might be more effective and how they might differ from each other when detecting cognitive distraction. Here, we compare the effectiveness of SVMs, SBNs and DBNs methods in detecting driver cognitive distraction from eye movements and driving performance. Nineteen performance measures were used to detect cognitive distraction. The training data were randomly selected from experimental data collected in a simulated driving environment. The models were evaluated using testing accuracy and two signal-detection-theory measures including sensitivity and response bias. DBNs, which consider time-dependent relationships, were expected to have the best performance. Of the two methods that only consider the relationship at single time point, it was expected that SVMs would perform better than SBNs because SVMs have fewer computational difficulties in terms of training and testing.

Types	Dispersion	Velocity
Fixation	Small ($\leq 1^\circ$)	Low, random direction
Saccade	Small ($> 1^\circ$)	400 – 600°/sec, straight
Smooth pursuit	Target decided ($> 1^\circ$)	1 – 30°/sec, target trajectory

Table 2. The characteristics of fixations, saccades, and smooth pursuits.

4.1 Data source

Data were collected using a fixed-based, medium-fidelity driving simulator. The driving scenario consisted of a straight, five-lane suburban road (two lanes of traffic in each direction divided by a turning lane). It was displayed on a rear-projection screen at 768 x 1024 resolution, producing approximately 50 degrees of visual field. During the experiment, participants, 35-55 years old, were required to drive six 15-minute drives: four IVIS drives and two baseline drives. During each IVIS drive, participants completed four separate interactions with an auditory stock ticker system. The interactions involved participants continuously tracking the price changes of two different stocks and reporting the overall trend of the changes at the end of the interaction. In baseline drives, participants did not perform the task. During all drives, participants were instructed to maintain vehicle position as close to the center of the lane as possible, to respond to the intermittent braking of the lead vehicle, and to report the appearance of bicyclists in the driving scene. Nine participants' eye and driving performance data was collected using a Seeing Machines faceLAB eye tracker and the driving simulator at a rate of 60 Hz. Further details can be found in a companion study [21].

4.2 Feature refinement

The raw eye data describing dynamic change of gaze vector-screen intersection coordinates were then translated into sequences of three eye movement components: fixations, saccades, and smooth pursuits. Segments of the raw data were categorized based on two characteristics: dispersion and velocity (see Table 2). Dispersion describes the span (in radians of visual angle) that the gaze vector covered, and velocity describes the speed (in radians of visual angle per second) and direction of the gaze vector (in radians) during a movement. The identification process began with a segment of six frames; based on the characteristics demonstrated in these frames, the posterior probabilities of the eye movements were calculated for the segment (see Figure 4.2). If the highest probability was greater than a certain threshold, the segment was identified as that eye movement. The segment was then increased by one frame, and the process was repeated to check if the eye movement continued in the new frame. If no movement could be identified, the segment was decreased by one frame, and the posterior probabilities were calculated again. If only three frames remained in the segment, the eye movement was identified using only the speed characteristic. When speed was high, the movement was labeled as a saccade; when the speed was low, it was labeled as a smooth pursuit. After each eye movement was completely identified, the identification process began again with a new six-frame segment. The likelihood of eye movements and the threshold of posterior probability were chosen according to the literature [12], and adjusted according to the particular characteristics of the data.

After identification, measures of eye movements were summarized over various windows to create instances that became the model inputs. Sixteen eye-movement measures included the mean and standard deviation of fixation duration, horizontal and vertical fixation location coordinates, pursuit duration, pursuit distance, pursuit direction and pursuit speed, mean blink frequency, and percentage of time spent performing pursuit movements. Fixation and pursuit duration and the percentage of time spent in pursuit movements

represent the temporal characteristics of eye movements, horizontal and vertical position of fixation relates to spatial distribution of gaze, and standard deviation more explicitly represents the variability of gaze. The pursuit distance, direction and speed capture the characteristics of smooth pursuits. Saccade measures were not included because the study was interested in how cognitive distraction may interfere with drivers' acquisition of visual information, which is greatly suppressed during saccades [12].

The driving simulation generated steering wheel position and lane position at 60 Hz. Steering error was calculated at 5 Hz, which describes the difference between the actual steering wheel position and the steering wheel position predicted by a second-order Taylor expansion [23]. Driving measures were summarized across the same window as eye movement measures. The driving performance measures consisted of the standard deviation of steering wheel position, mean of steering error, and the standard deviation of lane position.

4.3 Model training

For each participant, three kinds of models were trained with the randomly-selected data and the best model settings obtained from the previous studies [21, 20]. Testing accuracy and two signal detection theory measures including sensitivity and response bias were used to assess the models.

The best parameter settings for each kind of model were selected. IVIS drives and baseline drives were used to define each driver's cognitive state as distracted or not distracted. These became the prediction targets. The 19 performance measures including 16 eye movement measures and 3 driving performance measures that were summarized across a window (5, 10, 15, or 30 seconds long) were used as predictive evidence. SVM models used a continuous form of the measures, while BN models used a discrete form.

To train SVM models, the Radial Basis Function (RBF), $K(x_i, x_j) = e^{-\gamma|x_i - x_j|^2}$, was chosen as the kernel function, where x_i and x_j represent two data points and γ is a pre-defined positive parameter. The RBF is a very robust kernel function. Using the RBF, it is possible to implement both non-linear and linear mapping by manipulating the values of γ and the penalty parameter C , a pre-defined positive parameter used in the training calculation [11]. In training, we searched for C and γ in the exponentially growing sequences ranging from 2^{-5} to 2^5 , using 10-fold-cross-validation to obtain good parameter settings [5]. LIBSVM Matlab toolbox [5] was used to train and test the SVM models. BN training included structure learning, which tested the existence of conditional dependencies between variables, and parameter estimation, which decided the parameters of the existing relationships. With 19 measures, the structures of the BNs were constrained so that the training procedure was computationally feasible. For SBNs, the direction of the arrows with the target node (distraction or not distraction) was from the target node to performance nodes. The performance nodes could connect with one another. For DBN models (see Figure 4.3, the arrows within a time step were present only in the first time step and constrained as SBNs. After the first step, the arrows were only from the nodes in the previous time step to the current one. The BN models were trained using a Matlab toolbox [22] and an accompanying structure learning package [18].

Three types of models were trained with randomly selected data for each participant. The training data took about 2/3 of the total data. The other 1/3 was used as testing data. The training data for SVM and SBN models were time-independent instances, and the training data for DBN models were 120-second sequences. In total, there were 108 models trained, 36 models (9 participants x 4 window size) in each of DBNs, SBNs, and SVMs.

4.4 Model evaluation

Model performance was evaluated using three different measures. The first was testing accuracy, the ratio of the number of instances correctly identified by the model to the total number of testing instances. The other

two measures were associated with signal detection theory: sensitivity (d'), and response bias (β), which were calculated according to (1) and (2).

$$d' = \Phi^{-1}(HIT) - \Phi^{-1}(FA) \quad (1)$$

$$\ln(\beta) = \frac{|\Phi^{-1}(FA)|^2 - |\Phi^{-1}(HIT)|^2}{2} \quad (2)$$

where HIT is the rate of correct recognition among distracted cases, FA is the rate of incorrectly identifying the driver as distracted, and Φ^{-1} presents the function of calculating the z-score. d' represents the ability of the model to detect driver distraction. The larger d' value, the more sensitive the model. β signifies the strategy of the model. When β equals 0, the model classifies segments as distracted nor not distracted at equal rates, and false alarms and misses (not detecting distraction when it is present) tend to occur at similar rates. When $\beta < 0$, the models are classified as liberal and are more likely to overestimate driver distraction and to have higher false alarm rates than miss rates. When $\beta > 0$, the models are classified as conservative and are more likely to underestimate driver distraction and have more misses than false alarms. Both d' and β can affect testing accuracy. Separating sensitivity to distraction from model bias [30] makes for a more refined evaluation of the detection models.

4.5 Model comparison

We did a 3x4 (model types: DBN, SBN, and SVM by window size: 5, 10, 15, and 30 seconds) factorial analysis on three model-performance measures using a mixed linear model with participants as a repeated measure. We then performed post hoc comparisons using the Tukey-Kramer method with SAS 9.0.

DBN, SBN and SVM models were significantly different for testing accuracy and sensitivity (testing accuracy: $F_{2,16} = 6.6, p = 0.008$; sensitivity: $F_{2,16} = 32.5, p < 0.0001$). Shown on the left in Figure 4.5, DBNs and SVMs produced more accurate models than SBNs (DBNs: $t_{16} = 3.4, p = 0.003$; SVMs: $t_{16} = 2.77, p = 0.01$). The DBN and SVM models here had similar accuracy ($t_{16} = 0.66, p = 0.5$). On the right side of Figure 4.5, the DBN models are significantly more sensitive than the SVM and SBN models (SBN: $t_{16} = 7.7, p < 0.0001$; SVM: $t_{16} = 6.1, p < 0.0001$). Here, however, the SVM and SBN models have similar sensitivity ($t_{16} = 1.6, p = 0.13$). These comparisons indicate that using similar training data, DBNs can capture more differences in driver distraction and generate more sensitive models than the SBNs and SVMs. Although the SVM and SBN models showed similar sensitivity, the SVM models had an advantage in testing accuracy, perhaps due to their robust learning technique.

The decision bias presented marginal differences among the three types of models ($F_{2,16} = 2.8, p = 0.09$). The DBN models were more liberal than the SBN and SVM models (DBN: -1.85 ; SBN: -0.47 ; SVM: -0.55) with marginal significance (SBN: $t_{16} = 2.1, p = 0.051$; SVM: $t_{16} = 2.0, p = 0.06$). The SBN and SVM models had similar response biases ($t_{16} = 0.1, p = 0.9$), which were not different from the neutral model that is characterized by zero (SBN: $t_{16} = 1.1, p = 0.3$; DBN: $t_{16} = 1.3, p = 0.2$). These results can be used to explain the discrepancy in the comparisons of the DBNs' and SVMs' testing accuracy and sensitivity. Although being less sensitive than the DBN models, the SVM models reached the similar accuracy to the DBN models by using a more neutral strategy. Nevertheless, to explain how SBN and SVM models resulted in different accuracy given their similar sensitivity and response bias, the analyses of hit and false alarm rates were needed.

As be seen in Figure 4.5, the refined comparisons on false alarm and hit rates show that DBNs increased, and SVMs marginally increased, their hit rate compared to SBNs (overall: $F_{2,16} = 4.8, p = 0.02$; DBN: $t_{16} = 3.1, p = 0.008$; SVM: $t_{16} = 2.0, p = 0.06$). False alarm rates of the three types of models were

similar ($F_{2,16} = 1.1, p = 0.4$). This indicates that the DBN and SVM models reached higher accuracy than the SBN models by elevating hit rate detection.

The effect of window size interacts with model type to affect the false alarm rate (window size: $F_{3,24} = 3.0, p = 0.052$; interaction: $F_{6,46} = 2.0, p = 0.08$). No effect or interaction was found for testing accuracy, sensitivity, response bias, or hit rate. False alarm rate increased with window size from 5 to 15 seconds, and then decreased at 30 seconds (see dotted line in Figure 4.5). The DBN models followed this trend, but the magnitude of the change was more dramatic than that seen with the SBN and SVM models.

In summary, DBNs produced more sensitive models than SBNs and SVMs. The DBN and SVM models were more accurate and had higher hit rates than the SBN models. However, the effects of response bias on the three types of models were only marginally significant. Window size and its interaction with model type did not affect testing accuracy, sensitivity, response bias, or hit rate, but marginally affected false alarm rate.

5 Discussion

Compared to SBN and SVM, the DBNs, which model time-dependent relationships between drivers' behavior and cognitive state, produced the most accurate and sensitive models. This indicates that changes in drivers' eye movements and driving performance over time are important predictors of cognitive distraction. At the same time, the SVM models detected driver cognitive distraction more accurately than the SBN models. This suggests that the SVM learning technique has advantages over the BN technique. The cross-validation seems to have resulted in better parameters for the SVMs. We used a 10-fold cross-validation to search for the best C and γ values in the range of 2^{-5} to 2^5 . One possible result of selecting good parameters may be evident in the marginally increased hit rates and relatively lower false alarm rates of the SVM models, although the difference in false alarm rate was not significant. In addition, SVMs have less computational difficulties than BNs. It took less than one minute for SVMs to train a model, but approximately a half hour for SBNs and even longer for DBNs using an equal number of training data.

A good real-world detection system would need to both accurately detect driver cognitive distraction and minimize the false alarm rate to promote acceptance of the system. An interesting finding of this paper is that window size had a marginal effect on false alarm rate, although this variable did not affect the other four measures. This effect was particularly salient for DBNs. It also means that either a small (5 s) or large (30s) window size used to summarize drivers' performance measures will improve the false alarm rate without affecting overall model performance. This trend can be used in practice to improve system acceptance. However, as shown in Figure 4.5, the false alarm rates are still relatively high. Reducing false alarm rate is an important issue that future studies need to address.

Based on these comparisons, a hybrid learning algorithm that combines the time-dependent relationship and SVM learning technique could result in even better-performing models for detecting driver cognitive distraction from eye movements and driving performance. Possible ways to integrate SVMs and DBNs include bagging or paralleling. Bagging describes using multiple models to make prediction on one target. Bagging involves first training multiple (an odd-number of) SVM and DBN models with different training datasets to form a set of models. Then, each model makes prediction for the same datum (or case). The final decision of cognitive state for this datum (distracted or not distracted) depends on the vote of all models in the set. This method can reduce the variance of prediction and avoid overfitting.

Paralleling involves connecting two models sequentially. For example, if some aggregated descriptions of drivers' behavior, such as eye scanning patterns, are demonstrated to be essential for identifying cognitive distraction. We can first use SVMs to build the models that can identify the eye scanning patterns from eye movement measures and then use DBN models to infer cognitive states with the scanning patterns identified

from the SVM models. Such an approach combined a Bayesian Clustering by Dynamics and SVM model to forecast electricity demand [8]. Combinations of these two approaches can be used in the state refinement shown in Figure 2 to implement detection of cognitive distraction.

Models developed using data mining methods can be reciprocal with the top-down theories. For example, the relationship identified from BNs may help discover evidence to support a current theory or to create a new theory regarding the cognitive processes that underlie cognitive distraction. The theories related to human cognition, in turn, can provide some pre-knowledge that can be integrated into bottom-up models, such as the initial structure and structural constraints of BN models. Data mining and cognitive theories can directly cooperate to identify cognitive distraction. For instance, data mining methods can be used to summarize driver's performance into an aggregated characteristic of driving behavior, such as seriously-impaired driving behavior or diminished visual scanning field. Then, a model based on a cognitive theory can take the characteristic as an input to identify a driver's cognitive state. Thus, combination of top-down and bottom-up models may provide more comprehensive prediction for cognitive distraction than either alone.

However, some practical limitations exist in implementing such a detection system in the real world no matter which approach will be chosen to build a detection model. The first is to the problem of obtaining consistent and reliable sensor data, such as eye data [35]. For example, eye trackers may lose tracking accuracy when vehicles are travelling on rough roads or when the lighting conditions are variable. More robust eye tracking techniques are needed to make these detection systems truly reliable. Second, limitation is the delay from the time when driver's cognitive state changes to when distraction state is identified. Three source of delay have been identified [21]. They are the delay caused by response time of sensors, the time used to reduce sensor data and make prediction, and window size used to summarizing performance measures. The magnitude of each delay was approximately estimated in a previous study and depends on factors such as the computational efficiency of the algorithm [21]. Depending on how the estimated distraction is used, such delays could lead to instabilities as a distraction countermeasure is initiated just as the driver returns to the task of driving and is no longer distracted.

We have discussed how to detect visual and cognitive distraction as if they occur in isolation. In real driving environment, cognitive distraction often occurs with visual and manual distraction. To obtain comprehensive evaluation of driver distraction, the detection needs to cover all kinds of distractions. Two models that detects visual and one cognitive distraction need to work together in future IVIS. To simplify the detection, the procedure can begin by checking the level of visual and manual distraction because detecting visual-manual distraction is much easier to detect than cognitive distraction. If visual distraction is detected then it may not be necessary to check cognitive distraction.

6 Acknowledgments

The work presented here is part of the SAFETY VEHICLE(s) using adaptive Interface Technology (SAVE-IT) program that was sponsored by the National Highway Traffic Safety Administration (NHTSA) (Project Manager: Michael Perel) and administered by the John A. Volpe National Transportation Systems Center (Project Manager: Mary D. Stearns). We are grateful for the valuable comments offered by the graduate students in the Cognitive Systems Laboratory as well as for the assistance of Teresa Lopes at University of Iowa Public Policy Center in preparing this manuscript.

References

1. H. Alm and L. Nilsson. The effects of a mobile telephone task on driver behaviour in a car following situation. *Accident Analysis and Prevention*, 27(5):707–715, 1995.
2. S. Amari and S. Wu. Improving support vector machine classifiers by modifying kernel functions. *Neural Networks*, 12:783–789, 1999.
3. P. Baldi and S. Brunak. *Bioinformatics: The machine learning approach*, 2nd edition. MIT Press, 2001.
4. H. Byun and S. W. Lee. Applications of support vector machines for pattern recognition: A survey. In *Proceedings of Pattern Recognition with Support Vector Machines: First International Workshop, SVM 2002, Niagara Falls, Canada*, 2002.
5. C.-C. Chang and C.-J. Lin. Libsvm : a library for support vector machines. <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
6. N. Cristianini and J. S. Taylor. *An introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, Cambridge, 2000.
7. B. Donmez, L. N. Boyle, J. D. Lee, and G. Scott. Assessing differences in young drivers' engagement in distractions. In *the 85th Transportation Research Board Annual Meeting, Washington, D.C.*, 2006.
8. S. Fan, C. X. Mao, J. D. Zhang, and L. N. Chen. Forecasting electricity demand by hybrid machine learning model, 2006.
9. M. Guhe, W. Liao, Z. Zhu, Q. Ji, W. D. Gray, and M. J. Schoelles. Non-intrusive measurement of workload in real-time. In *Proceedings of Human Factors and Ergonomics Society 49th Annual Meeting, Orlando, FL*, pages 1157–1161, 2005.
10. W. J. Horrey and C. D. Wickens. Examining the impact of cell phone conversations on driving using meta-analytic techniques. *Human Factors*, 48(1):196–205, 2006.
11. C.-W. Hsu, C.-C. Chang, and C.-J. Lin. A practical guide to support vector classification. <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
12. R. J. K. Jacob. *Eye tracking in advanced interface design*, chapter Advanced Interface Design and Virtual Environments, pages 258–288. Oxford University Press, Oxford, UK, 1995.
13. Q. Ji, Z. Zhu, and P. Lan. Real-time nonintrusive monitoring and prediction of driver fatigue. *IEEE Transactions on Vehicle Technology*, 53:1052–1068, 2004.
14. S. G. Klauer, V. L. Neale, T. A. Dingus, D. Ramsey, and J. Sudweeks. Driver inattention: A contributing factor to crashes and near-crashes. In *Proceedings of Human Factors and Ergonomics Society 49th Annual Meeting, Orlando, FL*, pages 1922–1926, 2005.
15. T. Kumagai and M. Akamatsu. Prediction of human driving behavior using dynamic bayesian networks. *IEICE Transactions on Information and Systems*, E89-D:857–860, 2006.
16. D. Lambale, T. Kauranen, M. Laakso, and H. Summala. Cognitive load and detection thresholds in car following situations: Safety implications for using mobile (cellular) telephones while driving. *Accident Analysis and Prevention*, 31:617–623, 1999.
17. J. D. Lee, B. Caven, S. Haake, and T. L. Brown. Speech-based interaction with in-vehicle computers: The effect of speech-based e-mail on drivers' attention to the roadway. *Human Factors*, 43(4):631–640, 2001.
18. P. LeRay. The bnt structure learning package v.1.4.
19. X. Li and Q. Ji. Active affective state detection and user assistance with dynamic bayesian networks. *IEEE Transactions on Systems, Man, and Cybernetics–Part A: Systems and Humans*, 35:93–105, 2005.
20. Y. Liang, J. D. Lee, and M. L. Reyes. Non-intrusive detection of driver cognitive distraction in real-time using bayesian networks. In *Proceedings of TRB 86th Annual Meeting, Washington, D.C.*, 2007. paper number: 07-3101.
21. Y. Liang, M. L. Reyes, and J. D. Lee. Real-time detection of driver cognitive distraction using support vector machines. *IEEE Transactions on Intelligent Transportation Systems*. accepted.
22. M. P. Murphy. The bayes net toolbox.
23. O. Nakayama, T. Futami, T. Nakamura, and E. R. Boer. Development of a steering entropy method for evaluating driver workload, 1999. Presented at the International Congress and Exposition, Detroit, Michigan.
24. N. Oliver and E. Horvitz. *A comparison of HMMs and dynamic Bayesian networks for recognizing office activities*, chapter Lecture Notes in Computer Science, pages 199–209. SSpringer Berlin / Heidelberg, 2005.

25. M. A. Recarte and L. M. Nunes. Effects of verbal and spatial-imagery tasks on eye fixations while driving. *Journal of Experimental Psychology: Applied*, 6(1):31–43, 2000.
26. M. A. Recarte and L. M. Nunes. Mental workload while driving: Effects on visual search, discrimination, and decision making. *Journal of Experimental Psychology: Applied*, 9(2):119–137, 2003.
27. D. A. Redelmeier and R. J. Tibshirani. Association between cellular-telephone calls and motor vehicle collisions. *The New England Journal of Medicine*, 336(7):453–458, 1997.
28. M. A. Regan, J. D. Lee, and K. D. Young. *Driver Distraction: Theory, Effects and Mitigation*. CRC Press. (in preparation).
29. D. D. Salvucci and K. L. Macuga. Predicting the effects of cellular-phone dialing on driving performance. *Cognitive Systems Research*, 3:95–102, 2002.
30. H. Stanislaw and N. Todorov. Calculation of signal detection theory measures. *Behavior Research Methods, Instruments, & Computers*, 31(1):137–149, 1999.
31. D. L. Strayer, F. A. Drews, and W. A. Johnston. Cell phone-induced failures of visual attention during simulated driving. *Journal of Experimental Psychology: Applied*, 9(1):23–32, 2003.
32. D. L. Strayer and W. A. Johnston. Driven to distraction: Dual-task studies of simulated driving and conversing on a cellular telephone. *Psychological Science*, 12(4):462–466, 2001.
33. P.-N. Tan. *Introduction to data mining*. Pearson Addison Wesley: Boston, 2005.
34. V. N. Vapnik. *The nature of statistical learning theory*. Springer, New York, 1995.
35. T. W. Victor, J. L. Harbluk, and J. A. Engstrom. Sensitivity of eye-movement measures to in-vehicle task difficulty. *Transportation Research Part F*, 8:167–190, 2005.
36. E. L. Waltz. Information understanding: integrating data fusion and data mining processes. In *Proceedings of Proceedings of the 1998 IEEE International Symposium on Circuits and Systems, ISCAS '98, Monterey, CA*, volume 6, pages 553–556, 1998.
37. C. D. Wickens. Multiple resources and performance prediction. *Theoretical Issues in Ergonomics Science*, 3(2):159–177, 2002.
38. H. Zhang, M. R. H. Smith, and G. J. Witt. Identification of real-time diagnostic measures of visual distraction with an automatic eye-tracking system. *Human Factors*, 48(4):805–822, 2006.
39. Y. Zhang, Y. Owechko, and J. Zhang. Driver cognitive workload estimation: A data-driven perspective. In *Proceedings of IEEE Intelligent Transportation Systems Conference, Washington, D.C.*, pages 642–647, 2004.

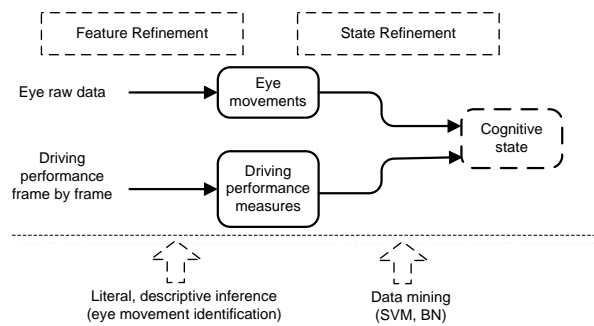


Fig. 1. Data fusion that transforms raw driving and eyemovement data into estimates of drivers cognitive distraction.

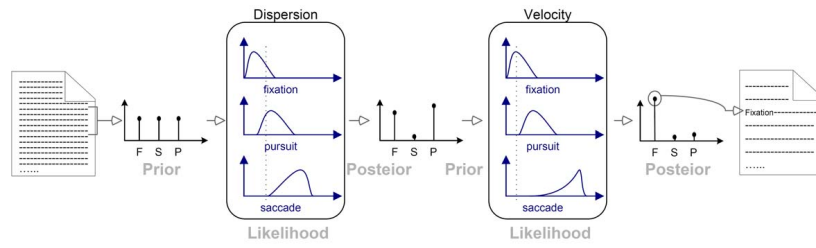


Fig. 2. Examples of a SBN and a DBN.

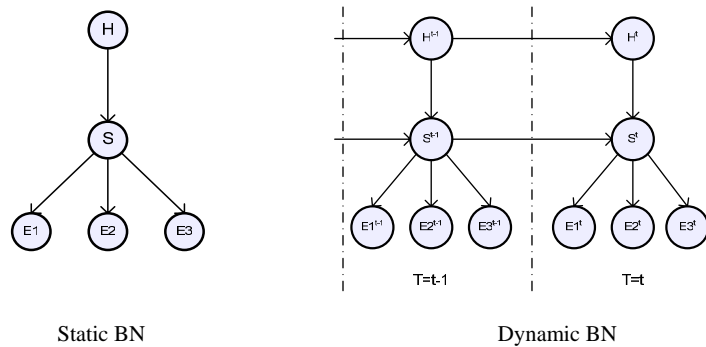


Figure 2 Examples of a SBN and a DBN.

Like SVMs, BNs can model complex, non-linear relationships. However, BNs aim to identify relationships between variables, and to use these relationships to generate predictions. BNs can explicitly present the relationships learned from data. As a consequence, studying a trained BN helps identify cause-effect links between variables, and the hierarchical structure of BN provides a systematic representation of these relationships. BNs are applicable to human-behavior modeling and have been used to detect affective state (Li & Ji, 2005), fatigue (Ji, Zhu, & Lan, 2004), lane change intent during driving (Kumagai & Akamatsu, 2006a, 2006b), pilot workload (Guhe et al., 2005), and driver cognitive distraction (Liang, Lee et al., in press). One disadvantage of BNs is that they become computationally inefficient when models have a large number of variables (i.e., 20). Moreover, compared to SVMs training techniques for BN are less robust than those available for SVMs.

Comparisons of SVMs and BNs

Although both SVMs and BNs have been successfully applied to detect cognitive distraction, it is not clear which might be more effective and how they might differ from each other when detecting cognitive distraction. Here, we compare the effectiveness of SVMs, SBNs and DBNs methods in detecting driver cognitive distraction using eye movements and driving performance. Nineteen performance measures were used to

Fig. 3. Illustration of the algorithm used to identify eye movements.

Fig. 4. Constrained DBN Structure, where solid arrows represent relationships within the first time step, dotted arrows represent relationships across time steps, and H and E presents the predictive target and performance measures, respectively.

Fig. 5. Comparisons of testing accuracy and sensitivity.

Fig. 6. The comparisons of hit and false alarm rates

Fig. 7. The comparison of false alarm rate for DBNs and the average