

# A Platform for the Development and Evaluation of Passive Safety Applications\*

Piotr Szczurek, Bo Xu, Ouri Wolfson, Jie Lin

**Abstract**— In this paper, we present a platform for aiding in the development and evaluation of novel ITS passive safety applications. Such applications work by having vehicles detect certain events that may be dangerous to other vehicles and disseminating reports about these events using wireless communication. A vehicle receiving the report about the event can then be warned. However, a large number of false warnings will lead to driver desensitization, which will reduce the safety benefit. To overcome this issue, a relevance estimator that will determine for which reports a warning will be given has to be devised for each new application. Our platform allows for an easy, fast method of developing these estimators and evaluating them in simulations. We demonstrated the feasibility of this approach with three example applications.

## I. INTRODUCTION

Recently, various Intelligent Transportation System (ITS) applications have been proposed for the purpose of increasing vehicular safety [1]-[4]. Many of those are passive safety applications that work by warning the drivers of specific events detected by other vehicles. The warnings are activated as a result of the information being generated by the vehicles and are broadcast through wireless communication devices. After seeing a warning, drivers can take the appropriate actions to mitigate any potentially dangerous situations. Through this warning mechanism, an ITS safety application can provide a safety benefit by reducing the risk of vehicular collisions. Examples of proposed applications that fit into this category include the Electronic Emergency Brake Lights (EEBL), the Highway Merge Warning (HMW), and the Control Loss Warning (CLW) [2], [3] (see subsection I.A). Although there is a great potential safety benefit from using these applications, the information these systems disseminate may not always be relevant to every vehicle. For example, a vehicle receiving a report about emergency deceleration occurring on a far away road segment is unlikely to affect it. There is thus a possibility of many false warnings being shown to the driver. Over time, this can lead to driver desensitization, which means the driver is likely to ignore the warnings [5]. To prevent this, the application developer must provide a method for estimating the relevance of the information to each vehicle that receives it. This allows for a warning to be displayed only when it is necessary, which increases the

drivers' trust in the system. The specification of the relevance estimator is hence an important part of the application development process. However, the determination of relevance can be uncertain and may depend on many factors, such as the positions, speeds, headings, and accelerations of vehicles, the vehicle types, or the road characteristics. It can therefore be difficult to develop an appropriate relevance estimator for a novel application. Determining its effectiveness will also be difficult, because in real life, events (e.g. emergency deceleration) about which the application will notify the vehicles may be rare.

The contribution of this paper is the introduction of the platform which enables entrepreneurs of novel passive safety applications an easy, fast method for the development and evaluation of passive safety applications. We postulate that determining relevance is a major stumbling block in such applications, and consequently we focus on relevance estimators. The proposed platform can be used for any passive safety application that aims to warn vehicles about a need for taking evasive actions through the dissemination of information about dangerous events on the road. It is most useful for applications in which the relevance of information is uncertain and determined by many factors. The platform relies on a relevance estimation method based on the Observe-Driver-and-Learn (ODaLe) principle, which was successfully used in prior work for determining the relevance of information for both safety ([6]) and non-safety ([7]) applications. In this paper, we show how three example applications (EEBL, HMW, and CLW) can be used within our proposed platform.

In the next subsection we will describe the three applications and then present relevant work in subsection I.B. In section II, we present a model for the type of applications that can be used within our proposed platform. The platform itself is then presented in section III. Section IV describes the evaluation and results for testing the platform. Lastly, section V provides a conclusion.

### A. Application Examples

#### 1) Emergency Electronic Brake Light (EEBL)

This application disseminates a report whenever a vehicle performs emergency deceleration. The report warns drivers and allows them to start braking earlier, thereby decreasing the chance of a vehicular collision. Intuitively, the relevance of an EEBL report depends on many features, including the positions of the receiving and sending vehicles, their velocities and lane positions, and the density of vehicles on the road.

\*This work was supported in part by the National Science Foundation IGERT program under Grant DGE-0549489.

Piotr Szczurek, Bo Xu, and Ouri Wolfson are with the Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607 USA (e-mail: pszczu1@uic.edu, boxu@cs.uic.edu, wolfson@cs.uic.edu).

Jie Lin is with Department of Civil and Materials Engineering, University of Illinois at Chicago, Chicago, IL 60607 USA (e-mail: janelin@uic.edu).

### 2) Highway Merge Warning (HMW)

In HMW, vehicles on the highway generate reports whenever they enter a segment of the road that merges with an on-ramp segment, where vehicles attempt to join the traffic on the highway. Receiving a report from the HMW application allows the merging vehicles to be warned whenever there is a possibility of a collision with the oncoming traffic. A driver that is shown a warning can slow down and be more alert when merging. For HMW, an intuitive formula for relevance may be derived based on the chance of a collision using the time-to-collision. The *time-to-collision* is the time a vehicle needs to reach the closest point intersecting the path of another vehicle.

### 3) Control Loss Warning (CLW)

A Control Loss Warning is used to warn drivers whenever a vehicle on the road loses control. This event may be detected by a stability control system of a vehicle. When a control loss occurs, a vehicle may become disabled and stop on the road and lane on which it was driving. In other cases, the vehicle may turn and end up in an adjacent lane.

## B. Related Work

Previous approaches for relevance estimation focused on particular applications. Some methods relied on rules or heuristics, that were based on some intuitive understanding of what makes a report relevant. For example, it has been proposed that the relevance can be determined based on the direction and position of the vehicle [2], [8]. The intuition is that a report from a vehicle travelling in the opposite direction may not be relevant. Another method was based on a heuristic, called the time-space relevance factor, which assumed that reports which are newer and originated closer to the vehicle will be more relevant [9]. Other methods utilized analytically derived formulas based on some assumptions. For example, assumptions about driver reactions can be used to calculate the minimum safety gap a vehicle must maintain to avoid a collision, which can determine the relevance of reports [10]. Other methods determined relevance based on assumptions of the vehicle's paths and the calculation of the time-to-collision [11], [12]. However, such a formula depends on the assumption of vehicles maintaining their current speeds and headings.

The disadvantage of all these approaches is that the assumptions or intuitions on which they are based may not hold, which can result in an increase in the number of false warnings. The techniques can also be difficult and time consuming to adapt for novel applications. In comparison, our platform uses a method based on the ODaLe principle, that has shown to outperform heuristic or analytic methods in many applications [6], [7]. It is also a general method that can be applied effectively to many novel applications.

## II. THE ITS SAFETY WARNING APPLICATION MODEL

The ITS safety warning application model represents any application in which the following occurs. A single vehicle, called the *sending vehicle*, detects some event that might be potentially dangerous to it or other vehicles, such as emergency deceleration or a control loss. Once this occurs, it

disseminates information to other vehicles to alert them of the possible need for an evasive action. Any vehicle that receives such information, called the *receiving vehicle*, will then estimate whether the information is relevant to its driver. If so, the vehicle will issue a warning so that the driver can perform any necessary actions to mitigate the potentially dangerous situation. Concrete examples of applications to which this model applies will be explained in section IV. In the next subsection, we discuss our assumptions about vehicle capabilities in our model. In subsection II.B, we will then discuss driver reactions to emergency situations and define the concept of emergency deceleration. Lastly, in subsection II.C, we define the class of ITS safety warning applications.

### A. Vehicle Capabilities

Our model assumes of a set of vehicles is present on the road, each of which is either participating or non-participating. A *participating vehicle* is equipped with a computing device with storage capabilities, a wireless communication device, a warning indicator, a digital map, and a set of sensors (accelerometer, GPS, and gyroscope). Conversely, a *non-participating vehicle* lacks all of the equipment carried by participating vehicles. For participating vehicles, the communication device is used to transmit data to other vehicles. The warning indicator is used to warn drivers about the need to initiate some evasive action (e.g. braking). The *digital map* contains information about the road on which the vehicle is driving, including the number of lanes and their exact positions. We assume that the set of sensors is the same among all participating vehicles. Every  $T$  milliseconds, called the *reporting period*, the sensors are used to generate a set of values describing the state of the vehicle. We will call this set the *vehicle state vector*. A common value for  $T$  for many applications is 100 milliseconds. Each vehicle state vector includes eight elements, grouped into two sets: vehicle statics and vehicle dynamics. Vehicle statics is information that is known to the vehicle at all times, such as the vehicle type and its tire type. Vehicle dynamics is information generated from the sensors and includes the position, speed, acceleration, heading angle, current lane id, and the current road segment id of the vehicle.

### B. Driver Reactions to Emergency Situations

In emergency situations, vehicles may perform certain evasive actions to try to avoid a collision. The warning indicator of the participating vehicle will function to alert the driver to take one of these actions. Typically, the immediate driver reaction to a dangerous situation will be to press hard on the brakes, which will cause the vehicle to decelerate with a maximum deceleration rate, which we call *Tsevere brake*. Observe that *Tsevere brake* is specific to a vehicle type, load, and weather conditions. We will state that a vehicle *emergency decelerates* when its deceleration equals *Tsevere brake*. Note that other reactions may be possible. For example, a vehicle may quickly change lanes to avoid a collision instead of decelerating.

### C. The ITS Safety Warning Application

An *ITS Safety Warning Application* is a program running continuously on the computer of each participating vehicle. It is divided into two processes: report generation and report reception. The report generation process is responsible for creating and disseminating new reports. It starts by retrieving the state of the vehicle sensors and uses it, along with the digital map, to determine the vehicle dynamics. It then combines this information with known vehicle statics and forms a vehicle state vector. Each application is characterized by a *report trigger*, which is a condition that must be satisfied for the inputs. For example, a report trigger for EEBL would be emergency deceleration. The state vector and the digital map are used to check the report trigger condition. If the condition is satisfied, a new *vehicle safety report* is created and then disseminated. Otherwise, there is no new report and in both cases the process repeats itself after waiting  $T$  ms for the next reporting period. The process is shown in fig. 1.

The report reception process is responsible for checking for newly incoming reports, determining its relevance, and warning the driver. When a vehicle receives a report, it uses a relevance estimation model to classify a report as either relevant or irrelevant. If the report is relevant, a warning indicator will then be activated. Otherwise, the report will be ignored. The feedback from this information can then be used to adapt the model. Fig. 2 illustrates this process.

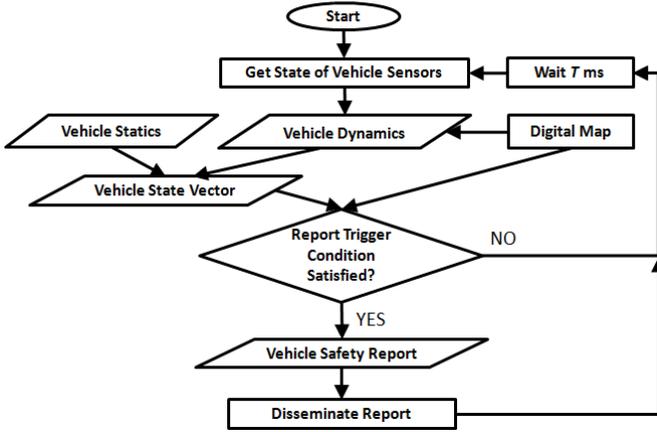


Figure 1. Report Generation Process

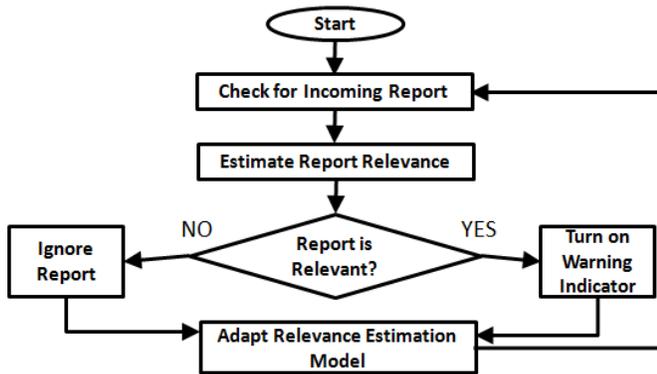


Figure 2. Report Reception Process

### III. THE PLATFORM

Our relevance estimator development and evaluation platform is based on the ODaLe principle and provides a set of tools for generating relevance estimation models for arbitrary ITS safety warning applications. The tools use the MITSIM traffic simulator [13], extended to enable the simulation of ITS safety applications, and the Weka Learning Toolkit [14] for performing machine learning. The tools also provide an ability to evaluate the safety benefit of the application for the learned relevance estimation model. In the next subsection, we will describe the relevance estimation method. In subsection III.B, we will explain the process of using the platform for developing and evaluating the relevance estimation models.

#### A. Developing Relevance Estimation Models using the Observe-Driver-and-Learn (ODaLe) Principle

ODaLe is a method for learning the relevance of information. The idea is to observe the driver's reaction after a report was generated by the application and use this information as an input to a machine learning algorithm. The objective is to learn a *relevance function*, which maps the reports to a value between 0 and 1, with 0 being completely irrelevant and 1 being completely relevant. The method works in two stages: the learning stage and the usage stage. In the learning stage, which we explain in the next subsection, the relevance function is instantiated using machine learning techniques. In the usage stage, explained in subsection III.B.2, the learned relevance function is used to decide whether to turn on the warning indicator. Since the learning of the initial model needs to be trained on many cases that may seldom occur in real life, a simulator like MITSIM is used for this purpose. Once the relevance estimation model is found and evaluated in simulations, it may then be deployed in real vehicles.

##### 1) Learning Stage

Every vehicle starts in the learning stage, which proceeds as follows. First, the warning indicator is disabled to allow us to observe how a driver normally reacts after a report is received. We check whether the report was relevant by monitoring the vehicle's behavior after the report was received for a specified fixed amount of time, called the *Reaction-delay*. The *Reaction-delay* is a parameter of the method that is the length of the period of time  $P$  after which the method determines whether the report is relevant or not.  $P$  starts when the report is generated and its optimal length, *Reaction-delay*, was found experimentally to be 5 seconds. A *relevant report* is one in which received the driver of the receiving vehicle performs an emergency deceleration within *Reaction-delay* seconds after the report was received. This definition stems from the intuition that a temporal correlation (occurrence within *Reaction-delay*) between receiving a report and undertaking emergency action implies the report necessitated the emergency deceleration. Therefore the report can be considered relevant. Note that with this definition of relevance, we imply that the warning indicator is meant to alert the driver to decelerate. The definition could easily be extended to include other types of emergency actions, such as lane changes. However, this would change

the meaning of the warning indicator and possibly confuse the drivers about which action to take (i.e. change lanes or decelerate).

Then, for every report received by a vehicle, we calculate values for a set of features and determine whether the report was relevant. Examples of features include the distance between the vehicles and the vehicle density. Generally, there is a common set of features that are useful in ITS safety applications. We have identified a large set of these features and make them available in our system. However, since the relevance of a report in an application may not depend on all of the features, it is necessary to choose a relevant subset of features. In other words, for each application, learning will be done based on a distinct set of features. Feature selection can be done manually by selecting the most intuitive features for the application. However, when there are many correlated features, manual selection may not yield optimal results. Therefore, our platform provides an automatic feature selection method using the algorithm described in [15]. This algorithm evaluates all subsets of features (assuming exhaustive search) using a correlation based heuristic. The method chooses a subset of features that are highly predictive, but have low correlation among them.

Once the feature values are calculated and the relevance of the report determined, the system then generates a training example. A *training example* consists of a set of report feature values and a label, positive or negative, indicating its relevance. A positive label is assigned when the report was determined to be relevant, and is then called a *positive training example*. Similarly, a training example is assigned a negative label when the report was not relevant, and is called a *negative training example*. The training examples that are generated are added to a training example set stored by each vehicle. Once a sufficient number of training examples is generated, the set of all stored examples is input to a machine learning algorithm, which uses the training examples to learn the relevance function. Examples of machine learning algorithms that can be used for this purpose are the Naïve Bayes and the Logistic Regression methods [16]. After the machine learning algorithm finds the relevance function, the method proceeds to the usage stage.

## 2) Usage Stage

In the usage stage, the warning indicator is enabled and new training examples are no longer generated. Instead, when a report arrives at a vehicle, the feature values are calculated and the learned relevance function is used to calculate the relevance of the report. The decision to turn on the warning indicator then depends on a fixed threshold called *Twarning*. When the relevance value of the report exceeds *Twarning*, the warning indicator will then turn on. Otherwise, if the value is less than or equal to *Twarning*, the report will be ignored.

The value of *Twarning* will affect the trade-off between safety and the number of false warnings. If set low, warnings will appear more frequently, causing more false warnings to be shown. Setting the threshold high will have the opposite effect. We set the *TWarning* threshold based on the set of generated training examples. We set its value such that the false warning rate is equal to the missed warning rate. The

*false warning rate* is the number of false warnings divided by the total number of warnings predicted by the learned model. A false warning was counted when the model predicted a warning for a negative training example, meaning the warning should not have been given. The *missed warning rate* is the number of missed warnings, divided by the number of positive training examples. A missed warning was counted for every positive training example in which the relevance estimation method would not have given a warning based on the set of feature values associated for that example.

## B. The Relevance Estimation Process

Fig. 2 shows the process of using the platform to develop and evaluate relevance estimation models for ITS Safety Warning applications. The user first defines the inputs consisting of a report trigger along with a set of scenarios. The report trigger is specified by a set of conditional statements, using predefined variables (e.g. current vehicle deceleration, speed, etc.) and constants (e.g. vehicle's maximum deceleration). Each scenario consists of the road network definition, an incident setting, and the ITS safety warning application parameter settings. The road network definition includes the geometry of roads, the speed limits, the departure rates, and other parameters that are normally used within MITSIM. The incident setting can be used to generate special events such as a stopped vehicle or a vehicle that lost control. The ITS application settings include the participation rate and the vehicle reaction to a report, which determines the behavior of the vehicle after it receives a warning. The behavior can be set to an emergency deceleration, where the vehicle immediately starts decelerating at the maximum rate until stopped, or a slow down, in which case the vehicle's desired speed is reduced to 25% of the calculated value.

Once the inputs are defined, multiple runs of the simulation are executed and the training examples are generated. The runs are executed in the learning mode, which means that vehicles will not react to the reports and will therefore follow their normal behavior, as specified by the MITSIM simulator. After the training examples are created, the system will then use half of these (the training set) for generating a model and the other half (the testing set) to evaluate it in terms of the false warning rate. The generation of the models is done by first running the automatic feature selection algorithm, then using one of the two available machine learning algorithms to learn a model. After this is done, the system will find the optimal value for the *TWarning* parameter and output the false warning rate. The system will also output the false warning rate associated with the *AlwaysWarn* baseline method, which assumes a warning will be given for every testing example.

The last step in the process is to evaluate the safety benefit of the application in terms of the average number of collisions per simulation run. This is done by re-executing the simulations using the previously learned model in MITSIM. The simulations are done in usage mode, which means that whenever a report is received, a relevance estimator module is used to determine its relevance. Then if

a report is determined to be relevant, the vehicle will react as specified in the inputs. The simulation can be set to use the learned model or one of two baselines, *AlwaysWarn* or *NeverWarn*. The *NeverWarn* baseline means that all reports are ignored by the vehicles. After the relevance model has been evaluated, it can then be deployed to real vehicles that use the application. This initial relevance model will be based on the simulated behavior of many drivers. However, the ODaLe principle can be utilized after deployment to fine-tune the model to individual drivers. The exact method for the adaptation of the model to individuals is part of our future work.

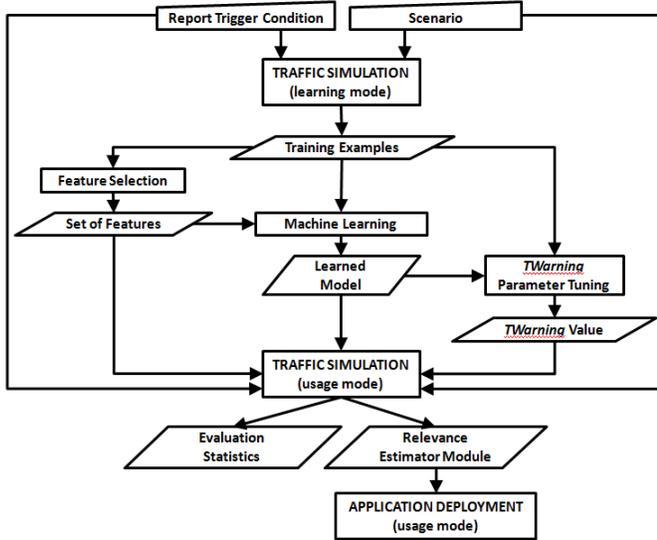


Figure 3. The process for developing relevance estimation models for ITS Safety Warning Applications using the proposed platform.

#### IV. EVALUATION

To evaluate the effectiveness of the platform, we have developed the relevance estimation models for the three examples applications and tested them in terms of the number of collisions and the false warning rate. Four methods were compared for each application: ML-NB, ML-LR, *AlwaysWarn*, and *NeverWarn*. ML-NB and ML-LR methods used the learned naïve Bayes and the logistic regression relevance estimation models, respectively. For each application, we set up a typical scenario and specified the report trigger. This is described in subsections IV.A through IV.C. We then used our developed tools for running the simulations and generating the relevance estimation models. The results are discussed in subsection IV.D.

##### A. Evaluation Environment for EEBL

For EEBL, the scenario consisted of a single straight road with a varied number of lanes (1-3). For each run, we also varied the departure rate (400-1200 veh/h/lane), and the speed limit (55-65mph). A stopped vehicle incident scenario was specified to occur at the end of the segment. The report trigger was defined as acceleration equal to the maximum deceleration. Vehicles' reaction to a report was emergency deceleration.

##### B. Evaluation Environment for CLW

For the CLW application, the scenario was the same as for the EEBL application, except for the incident, which was specified as a control loss instead of the immediate stop. The report trigger was set to control loss equal true. Vehicle's reaction to a report was a slow down.

##### C. Evaluation Environment for HMW

In HMW, the scenario consisted of two merging roads. The first road (highway) was a straight and divided into three segments, 200 feet in length. The first and the third segments were single lane segments, while the second segment had two lanes. The second road (on-ramp road) consisted of a one lane straight segment connected to a curved, one lane on-ramp segment that merged into the left lane of the second segment of the first road. For each run, we varied speed limits on the highway (55-65mph) and the on-ramp (25-35mph) and the departure rates (400-800 vehi/h/lane for on-ramp, 1400-1800 veh/h/lane for highway). The report was triggered whenever the vehicle entered the second highway segment. Vehicle's reaction to a report was a slow down.

#### D. Results

##### 1) Automatic Feature Selection

The result of using the automatic feature selection for each application is shown in the table below.

TABLE I. LIST OF ITS SAFETY WARNING APPLICATIONS WITH THE FEATURES SELECTED BY THE AUTOMATIC FEATURE SELECTION ALGORITHM

Application	Selected Features
EEBL	distance, difference-in-accelerations, difference-in-speeds
HMW	difference-in-accelerations, difference-in-speeds, difference in time-to-collision
CLW	difference-in-accelerations, difference-in-speeds

In the case of EEBL, the algorithm picked three features that are related to the vehicle dynamics. The selection follows the intuition, since the lower the distance between the two vehicles, the higher the chance that the receiving vehicle may be affected by the emergency braking. The difference in accelerations is also important, because a report may be more relevant if the receiving vehicle is currently accelerating, while the sending vehicle is emergency braking, and vice versa. Similarly, if the receiving vehicle is travelling faster than the sending vehicle, the report may also be more relevant.

For the HMW application, three features were chosen. Two features, difference-in-accelerations and the difference-in-speeds, were the same as for EEBL. These features are important for the same reasons. However, in HMW the difference in time-to-collision feature was also chosen. Intuitively this feature makes sense for an application like HMW, because it indicates whether two vehicles are on a collision course.

Lastly, in the case of the CLW application, the algorithm chose the difference in acceleration and speeds, but did not choose the distance, which was chosen for EEBL. This may be appropriate, because whenever a vehicle losses control, it

may end up in any place on the road. However, do note that the tested scenarios were limited in our evaluation, so more features may have been selected if a greater number of scenarios were used.

### 2) False Warning Rates

The results of the false warning rate evaluations are shown in fig. 3. The results show that both machine learning methods were able to significantly decrease the number of false warnings in comparison to the baseline, AlwaysWarn, method. In all applications, the percentage of false warnings for the AlwaysWarn baseline exceeded 80%, which would most likely lead to driver desensitization. Conversely, the machine learning methods achieved much lower false warning rate in all applications.

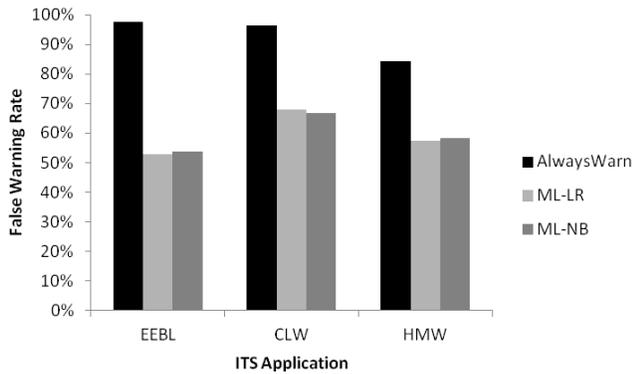


Figure 4. Comparison of the false warning rates for each application.

### 3) Average Number of Collisions

The results of the safety benefit evaluation are shown in fig. 4. Both machine learning methods were close in performance to the AlwaysWarn method and significantly better than the NeverWarn default method. While AlwaysWarn achieved fewer collisions than the machine learning methods, the differences were not significant. Additionally, with false warning rate over 80%, the use of this method would quickly lead to driver desensitization. Therefore over time, the safety benefit of this method cannot be sustained.

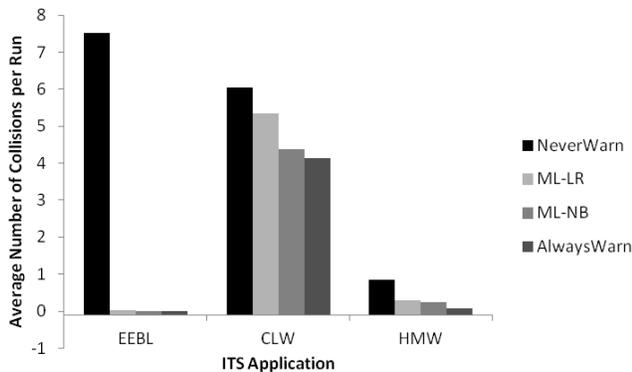


Figure 5. Comparison of the average number of collisions per run for each application.

## V. CONCLUSION

This paper introduced a platform for developing and evaluating relevance estimators in novel ITS safety applications. The platform provides a set of tools based on the Observe-Driver-and-Learn principle for learning and testing relevance estimation models using machine learning techniques. The evaluation proved the feasibility of using the platform for three different applications. The results showed that the learned models achieved significant decreases in the number of false warnings, while maintaining the safety benefits of the applications.

## REFERENCES

- [1] A. Carter and J. Chang, "Using Dedicated Short Range Communications for Vehicle Safety Applications – the Next Generation of Collision Avoidance," in *Proc. 21<sup>st</sup> ESV International Technical Conference on the Enhanced Safety of Vehicles*, June 2009, Paper-09-0330.
- [2] A. Carter, "The status of vehicle-to-vehicle communication as a means of improving crash prevention performance," Tech. Rep. 05-0264, NHTSA, 2005.
- [3] F. Ahmed-Zaid, et al., "Vehicle Safety Communications – Applications (VSC-A) Final Report," NHTSA Pub. DOT HS 811 492A, Sept. 2011.
- [4] F. Ahmed-Zaid, et al., "Vehicle Safety Communications - Applications: System Design & Objective Testing Results", *SAE Int. J. of Passeng. Cars – Mech. Syst.*, vol. 4, no. 1, pp. 417-434, June 2011.
- [5] T. A. Dingus, D. V. McGehee, N. Mankal, S. K. Jahns, C. Camey, and J. M. Hankey, "Human factors field evaluation of automotive headway maintenance/collision warning devices," *Human Factors*, vol. 39, pp. 216-229, 1997.
- [6] P. Szczurek, B. Xu, J. Lin, and O. Wolfson, "Intelligent Transportation Systems: When is Safety Information Relevant?," in *Proc. 12th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2011*, Lucca, Italy, June 20-23, 2011.
- [7] P. Szczurek, B. Xu, J. Lin, and O. Wolfson, "Spatio-temporal Information Ranking in VANET Applications," *International Journal of Next-Generation Computing*, vol. 1, no. 1, 2010.
- [8] Y. Zang, L. Stibor, H. J. Reumerman, and H. Chen, "Wireless local danger warning using inter-vehicle communications in highway scenarios," in *Proc. 14th European Wireless Conference*, June 2008, pp.1-7, 22-25.
- [9] B. van Arem, C. M. J. Tampère, and K. M. Malone, "Modeling traffic flows with intelligent cars and intelligent roads," in *Proc. IEEE Intell. Vehicles Symp.*, Columbus, OH, June 2003, pp. 456–461.
- [10] B. A. Galler and H. Asher, "Vehicle-to-Vehicle Communication for Collision Avoidance and Improved Traffic Flow," Technical Report, Transportation Research Board, May 1995.
- [11] R. Miller and H. Qingfeng, "An adaptive peer-to-peer collision warning system," in *Proc. Vehicular Technology Conference*, pp. 317-321, 2002.
- [12] L. Tu and C. Huang, "Forwards: A Map-Free Intersection Collision-Warning System for All Road Patterns," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 7, Sept. 2010
- [13] Q. Yang, and H. N. Koutsopoulos, "A Microscopic Traffic Simulator for Evaluation of Dynamic Traffic Management Systems," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 3, pp. 113-129, June 1996.
- [14] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update," *SIGKDD Explorations*, vol. 11, no. 1, 2009.
- [15] M. A. Hall (1998). *Correlation-based Feature Subset Selection for Machine Learning*. Hamilton, New Zealand.
- [16] S. Le Cessie and J. C. van Houwelingen, "Ridge Estimators in Logistic Regression", *Applied Statistics*, vol. 41, no. 1, pp. 191-201, 1992.