A Methodology for the Development of Novel VANET Safety Applications[1]

Piotr Szczurek, Bo Xu, Ouri Wolfson
Department of Computer Science
University of Illinois at Chicago
Chicago, IL, U.S.A.
pszczu1@uic.edu, {boxu,wolfson}@cs.uic.edu

Jie Lin
Department of Civil and Materials Engineering
University of Illinois at Chicago
Chicago, IL, U.S.A.
janelin@uic.edu

ABSTRACT
We present a methodology for the development of passive ITS safety applications that aim to disseminate reports about dangerous events on the road. Examples of such applications include the emergency electronic brake light or the highway merge warning. A major issue with such applications is the decision of when a warning should be shown. Since the recipient vehicle may be far away from where the dangerous event occurred, a large number of false warnings may be shown to the drivers. This leads to driver desensitization which may reduce the safety benefits. While previous research has provided a way of handling false warnings by estimating the relevance of the reports, these methods do not take into all the important factors and are not easily adaptable for novel applications. In this paper, we propose a simulation platform for developing and evaluating relevance estimators for passive ITS safety applications that can be utilized for developing novel applications. The paper provides examples of the effectiveness of this platform on three previously proposed applications.

Categories and Subject Descriptors
H.4.3 [**Information Systems Applications**]: Communications Applications; I.6.5 [**Simulation and Modeling**]: Model Development

General Terms
Performance, Design, Experimentation.

Keywords
VANET, machine learning, safety, DSRC.

1. INTRODUCTION
The use of VANETs has been frequently proposed for use in vehicular safety applications. A major driver of these applications is the Crash Avoidance Metrics Partnership Consortium, which created a project that resulted in the standardization of the SAE J2735 Basic Safety Message (BSM) that contains the vehicle state information (e.g. position, speed, and acceleration) important for safety applications [1]. Examples of the proposed applications include the Electronic Emergency Brake Lights (EEBL), the Highway Merge Warning (HMW), and the Control Loss Warning (CLW) [2]. The EEBL application disseminates a report whenever a vehicle performs emergency deceleration. 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. Lastly, CLW is used to warn drivers whenever a vehicle on the road loses control. All of these are examples of passive safety applications which 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.

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 report about emergency deceleration is unlikely to affect a vehicle that is far away from where this event took place. 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 [3]. 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. However, the determination of relevance can be uncertain and may depend on many factors, such as the positions, speeds, and accelerations of vehicles, the road characteristics, or weather. With such a large quantity of factors, it can therefore be difficult to develop an appropriate relevance estimator for a novel application. Determining whether the developed estimator is effective will also be difficult, because in real life, events (e.g. emergency deceleration) about which the application will notify the vehicles may be rare.

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 [6],[7] or relied on analytically derived formulas [8],[9]. 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. 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

[1] This work was supported in part by the National Science Foundation IGERT program under Grant DGE-0549489 and IIS-0957394.

VANET'12, June 25, 2012, Low Wood Bay, Lake District, UK.
ACM 978-1-4503-1317-9/12/06.

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.

2. 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.

2.1 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 the following: computing device with storage, communication device (e.g. DSRC), warning indicator, digital map, and sensors (accelerometer, GPS, and gyroscope). Conversely, a *non-participating vehicle* lacks the equipment carried by the 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. We assume that the set of sensors is the same among all participating vehicles. Every *T* ms (we have used 100ms), called the reporting period, the sensors are used to generate a set of values, called the vehicle state vector, describing the state of the vehicle. 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 (e.g. vehicle or tire type) and vehicle dynamics is information generated from the sensors (e.g. position, speed, acceleration). In emergency situations, vehicles may perform certain evasive actions to try to avoid a collision. Typically, the immediate driver reaction 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*.

2.2 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 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.

3. THE PLATFORM

Our relevance estimator development and evaluation platform provides a set of tools for generating relevance estimation models for arbitrary ITS safety warning applications. The tools use the MITSIM simulator [10], extended to enable the simulation of ITS safety applications and the Weka Learning Toolkit [11] for machine learning. The tools provide an ability to evaluate the safety benefit of the application for the learned relevance estimation model. The platform is based on the Observe-Driver-and-Learn (ODaLe) principle, which is a method for learning the relevance of information that has previously been used for safety [4] and non-safety applications [5]. 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 (irrelevant) and 1 (relevant). The method works in two stages: the learning stage and the usage stage. In the learning stage the relevance function is instantiated using machine learning techniques. This is done by turning off the warning indicator and observing how a driver normally reacts after a report is received. We then label each report as either positive (i.e. relevant) or negative (i.e. not relevant) by monitoring the vehicle's behavior after the report was received for a specified fixed amount of time, called the *Reaction-delay* (5s was found to be optimal experimentally). A relevant report is one in which received the driver of the receiving vehicle performed an emergency deceleration within *Reaction-delay* seconds after the report was received. We then calculate values for a set of features for the report, such as the distance between the vehicles or the vehicle density. We have identified a large set of these features and make them available in our system. We also provide an automatic means of feature selection using the algorithm described in [12]. The features along with the label form a training example which is stored in an example set. Once a sufficient number of examples are generated, they are used as input to a machine learning algorithm that finds the relevance function. Two algorithms that are used in our platform include the Naïve Bayes and the logistic regression methods.

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, 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 therefore set the value of *TWarning* 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.

4. EVALUATION

To evaluate the platform, we have tested the different relevance estimation methods for three applications in terms of the number of collisions and the false warning rate. Four methods were compared: 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. AlwaysWarn and NeverWarn are baseline methods that assume reports are respectively either always or never relevant. For each application, we set up typical scenarios and specified the report trigger. Each scenario had a varied number of lanes (1-3), speed limits (25-65mph), departure rates (400-1800 veh/h/lane). We then used our developed tools for running the simulations and generating the relevance estimation models. The results (see figs. 1 and 2) 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. In terms of the number of collisions, 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.

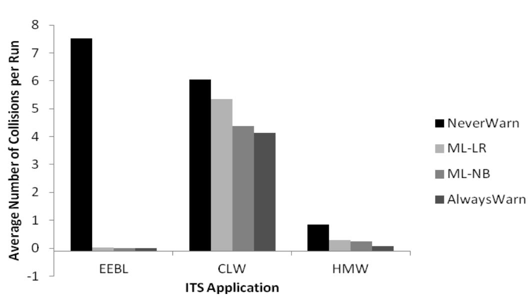

Figure 1. Comparison of average number of collisions.

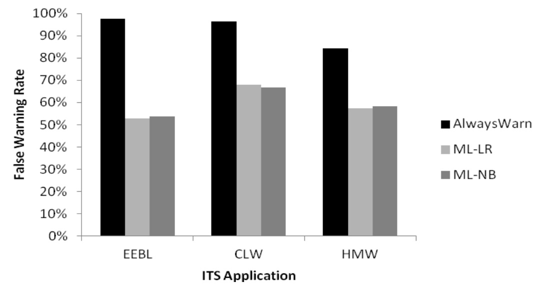

Figure 2. Comparison of the false warning rates.

5. REFERENCES

[1] F. Ahmed-Zaid, et al., "Vehicle Safety Communications – Applications (VSC-A) Final Report," NHTSA Pub. DOT HS 811 492A, Sept. 2011.

[2] 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.

[3] T. A. Dingus, D. V. McGehee, N. Mankkal, 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] B. A. Galler and H. Asher, "Vehicle-to-Vehicle Communication for Collision Avoidance and Improved Traffic Flow," Technical Report, Transportation Research Board, May 1995.

[9] R. Miller and H. Qingfeng, "An adaptive peer-to-peer collision warning system," in Proc. Vehicular Technology Conference, pp. 317-321, 2002.

[10] 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.

[11] 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.

[12] M. A. Hall (1998). Correlation-based Feature Subset Selection for Machine Learning. Hamilton, New Zealand