Intelligent Transportation Systems: When is Safety Information Relevant?

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

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

Abstract— **In this paper, we compare two methods of estimating relevance for the emergency electronic brake light application. One uses an analytically derived formula based on the minimal safety gap required to avoid a collision. The other method uses a machine learning approach. The application works by disseminating reports about vehicles that are performing emergency deceleration in effort to warn drivers about the need to perform emergency braking. Vehicles which receive such reports have to decide whether the information contained in the report is relevant to the driver, and warn the driver if that is the case. Common ways to determine relevance are based on the lane or direction information, but using only these attributes can still lead to many false warnings, which can desensitize the driver. Desensitized drivers may ignore warnings or turn off the system completely, thus eliminating any safety benefits of the application. We show that the machine learning method, in comparison to the analytically derived formula, is able to significantly reduce the number of false warnings by learning from the actions drivers take after receiving a report. The methods were compared using simulated experiments with a range of traffic and communication parameters.**

Keywords- machine learning, vehicle safety

I. INTRODUCTION

In 2005, the National Highway Traffic Safety Administration (NHTSA) released a document identifying eight potential safety applications which utilize Dedicated Short Range Communication (DSRC) technology [1]. The applications were selected based on potential safety benefits they provide. Among them, the Emergency Electronic Brake Light (EEBL) application was determined to be one of three applications to possess high benefit potential. EEBL was defined as an application that alerts drivers of any hard braking done by vehicles in front of them. The idea was to extend drivers' visibility through the emergency brake notifications. This was described as most helpful in situations where visibility is limited, such as in adverse weather conditions.

EEBL works by vehicles disseminating a report each time they perform emergency braking, which occurs when the deceleration rate exceeds a predefined threshold. When a report arrives at a vehicle, the system checks whether the information contained in the report is relevant to the driver. Based on this, a warning would be given to the driver. In [1], authors suggested that a report could be considered based on the vehicle's lane. However, using lane information as the only factor, all drivers of the following vehicles in the same lane would see the warning. This could cause many unnecessary warnings to be shown, which might result in drivers ignoring the EEBL system. False warnings could be eliminated by restricting the EEBL warnings to those vehicles for which emergency braking is necessary. This could be done by calculating required deceleration forces, given certain report attributes, such as distance, vehicle speeds, or vehicle density. However, such equations may not take into account normal driver behavior and might result in drivers to consider the warnings as unnecessary. It may also be difficult to come up with the proper equation that would take into account all the relevant factors.

In this paper, we compare two ways of determining the relevance. One approach uses an analytically derived formula that tries to estimate the minimal safety gap a vehicle would need to avoid a collision. The other is our proposed method of using machine learning techniques for learning reports' relevance. In this method, vehicles check whether emergency braking was done within a fixed amount of time after receiving each report. Based on this, training examples are created for a machine learning process which learns a report relevance model. The learned model can then be used to determine the likelihood of an arbitrary report being relevant. The decision to warn the driver is then based on this likelihood, which depends on several factors, such as the distance between the reporting and receiving vehicle, the density of vehicles on the road, and the velocity of the vehicles. The advantage of our proposed method is the ability to easily combine these individual factors. Moreover, through simulations, we show that the method is able to significantly reduce the number of warnings that drivers would perceive to be unnecessary.

II. RELEVANT WORK

The impact of false warnings on driving behavior is studied in [2], based on a collision avoidance system. It is discovered that when the false warning rate increases over 60%, drivers begin to decrease their headways, implying they start to distrust the system.

Current simple strategies for determining relevance are based on lane and direction [3]. A more sophisticated method is used in [4], where authors propose that warnings be evaluated based on a time-space relevance factor. However, the details of the exact form of the relevance factor are not provided. In [5], warnings are given for collisions at intersections when the difference between the estimated time-to-collision and time-to-avoidance values are within a certain threshold. Although such a method could be employed in an EEBL application, the formulation uses a fixed number of factors and cannot be easily extended to include additional information (such as weather). In [6], authors propose to give warnings based on calculation of required deceleration forces to avoid a collision. However, authors assume full knowledge of every vehicle's position, velocity, acceleration, and length, which is not available in the EEBL application.

Use of machine learning for relevance estimation has been previously used in [7, 8] for the purposes of ranking reports in travel time and parking information dissemination applications. Handling of false warnings through machine learning has been studied in [9], where authors experiment with a smartphone application alerting drivers of slowdowns in traffic on the road ahead (see [10] for application details). While the work presented in this paper also uses machine learning techniques for avoiding false warnings, the key difference is the feedback mechanism. In contrast to the procedure described in [9], our method for feedback is automatic and therefore does not involve users inputting their preferences manually. The advantage of this, aside from eliminating burden on the users, is that the feedback is based on user action and not stated preference, which may be imprecise.

One application related to the EEBL is a PreCrash system. A PreCrash system is designed to detect a possibility of a collision and is typically used to prepare the vehicle for an imminent impact by pre-tensioning seat belts or taking other actions [11, 12]. While this system may also be used to warn drivers, similarly to the EEBL application, since it typically only relies on knowledge of the vehicle ahead, it may not provide warnings in sufficient time for driver to take action. Other related applications include the Adaptive Cruise Control [13] and Cooperative Adaptive Cruise Control [14]. The goal of these systems is to set vehicle's acceleration or deceleration in order to maintain a safe following distance, typically through a defined formula. Unlike EEBL, these systems often rely on vehicles being equipped with lidar or similar device, and are not designed for warning drivers.

III. MODEL ENVIRONMENT

The environment consists of a set of vehicles, controlled by human drivers. A subset of these vehicles is equipped with the following:

- On-board computer, with storage capability
- Positioning device (e.g. GPS)
- Short-range wireless communication device (e.g. DSRC or Wi-Fi)
- Accelerometer
- Warning indicator
- Digital map

We assume that the storage of each vehicle has an unlimited capacity and that the positioning device is accurate at all times. The accelerometer on the vehicle senses the deceleration rate. The warning indicator is used to warn drivers about the need to brake. The digital map contains information about the road on which the vehicle is driving.

We call the subset of vehicles that are equipped as *participating vehicles*. Otherwise, they are called *non-participating vehicles*. Non-participating vehicles do not have a warning indicator (no EEBL) or any communication capabilities. The ratio between participating vehicles and the total number of vehicles is called the *participation rate*.

IV. THE EMERGENCY ELECTRONIC BRAKE LIGHT APPLICATION

Our system design for the EEBL application consists of three elements: report creation, report dissemination, and report relevance estimation. The system works with vehicles creating reports based on their deceleration. These reports are then disseminated to nearby vehicles using the short-range communication device. Once a report arrives at a vehicle, the relevance of the report is estimated. Based on this estimate, the warning indicator light is turned on to alert the drivers about the need to perform emergency braking. We now discuss the details of each of the design elements individually.

A. Report Creation

A vehicle *emergency decelerates* when its deceleration force exceeds or equals the threshold *Tsevere_brake*. Naturally, the *Tsevere_brake* threshold corresponds to hard braking, so we will assume that it is set to the maximum deceleration force of the vehicle. When a vehicle initiates an emergency deceleration, they generate an *emergency brake report* which includes the current location and speed of the vehicle, and the time when the emergency deceleration was performed. Additional reports will be generated continuously, every 0.1 seconds, while the emergency deceleration is sustained.

B. Report Dissemination

Reports are disseminated using the short-range communication device through *broadcast communication*. In broadcast communication, the reports sent can be received by all *neighboring vehicles*. Neighboring vehicles are those within a fixed distance (i.e. the transmission range) of the sender. New reports received by a vehicle are immediately rebroadcast to its neighbors.

C. Report Relevance Estimation

When a report is received, we estimate its relevance to the vehicle through one of the methods presented in the next section. The methods also define whether a warning will be shown to the driver for the given estimate of relevance.

V. THE RELEVANCE ESTIMATION METHODS

In this section, we will present two methods for estimating relevance of emergency brake reports. Both methods utilize the same information for calculating relevance, which is information that would be expected to be available in an EEBL system. This includes information about the vehicle that generated the emergency brake report, the vehicle that received the report. However, specific information about other vehicles is not assumed to be known. The first method, discussed in the next subsection, uses an analytically derived formula. The second, discussed in the following subsection, is our proposed method that utilizes machine learning techniques.

A. Analytic Method

This method determines relevance based on an estimate of the minimum safety gap. The *gap* is defined in terms of the distance between the receiving vehicle and the vehicle in front of it. We define the *minimum safety gap* as the minimum gap, which the receiving vehicle requires, to avoid a collision with the vehicle in front of it. This is based on the assumption that at the time the report is received by a vehicle, the vehicle in front of it will immediately emergency decelerate until stopped. Given this assumption, if the gap between the two vehicles is less than the minimum safety gap, the report will be labeled as relevant. Otherwise, the report will be labeled as irrelevant.

To find whether the gap exceeds the minimal safety gap, we consider the scenario shown in fig. 1. There are vehicles driving on a road with a single lane. One of the vehicles (*A*), then receives a report from a downstream vehicle (*C*).

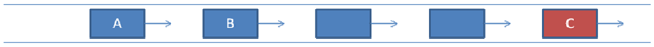

Figure 1. Assumed scneario for deriving the formula. Shows vehicle C sending report to vehicle A, which has vehicle B in front of it.

Assume that the vehicle directly in front of *A*, vehicle *B*, will emergency decelerate immediately after *A* receives the report. When this happens, we want to estimate whether the gap between *A* and *B* is less than or equal to the minimum safety gap. To do this, we have to know the position and velocity of *B*. However, in the EEBL system, a report would contain information only about *C*. Therefore, we have to estimate the initial (i.e. at the time of report generation) position, velocity, and acceleration of *B* using available information. For velocity and acceleration, we will use the values of *C* that are available from the report. For position, we will assume we have knowledge of the density of vehicles on the road. The density can either be calculated from knowledge of current travel times or it can be estimated locally from wireless communication device signals and knowledge of the participation rate (see [15]). Let $s_x(t)$, $v_x(t)$, and $a_x(t)$ be the position, velocity, and acceleration of vehicle *X* at time *t*. Then, assuming equally spaced vehicles in between *A* and *C*, the initial position of B can estimated from density as follows:

$$s_B(0) = \min\left\{\frac{s_C(0) - s_A(0)}{[s_C(0) - s_A(0)] \cdot density + 1} + s_A(0). \qquad s_C(0)\right\} \tag{1}$$

Note that the minimum is taken between the estimate and the initial position of *C*, because the estimate should never be greater than the position of *C*.

Once we have the information about B, we can find the minimal safety gap formula. We assume a driver reaction time of 1.0 seconds, using the study in [16]. With this assumption, we then find the formula for the minimal safety gap, based on equations derived in [6]. We have omitted the formula in this paper due to the space limitations.

B. Machine Learning Approach

The general idea behind our machine learning approach is to use the received reports as an input to a supervised machine learning algorithm. The objective is to learn whether emergency deceleration by the reporting vehicle will necessitate emergency deceleration by the receiving vehicle. The method works in two stages: learning and usage. 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. Then, for every report received by a vehicle, we calculate values for a set of report attributes and determine whether the report was relevant. We will discuss the selection and calculation of the attributes in subsection 1. We check whether the report was relevant by monitoring the vehicle's behavior after the report was received for a specified amount of time, called the *Reaction-delay seconds (*explained below). We then use the following definition to determine whether the report was *relevant*: a report *r* received by a vehicle *v* is relevant, if the driver of *v* performs an emergency deceleration within *Reaction-delay* seconds after *r* was generated.

The relevance definition stems from the intuition that a temporal correlation (occurrence within *Reaction-delay*) between receiving a report and undertaking emergency deceleration implies that the event indicated by the report necessitated a reaction from the driver and is hence relevant. *Reaction-delay*, is the time interval starting when the reported event occurs, and ending when the driver starts reacting. This involves the time it takes for the reported event to cause a chain reaction that becomes visible to the driver, as well as the driver reaction time. This is the reason for waiting the *Reaction-delay* seconds after the report is received before determining its relevance. For our experiments we found that a *Reaction-delay* value of 5 seconds achieved the best performance.

Once the attribute 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 attribute values and a label indicating its relevance. Labels can be either positive (relevant) or negative (not relevant). The training examples that are generated are added to a training example set stored by each vehicle. Once a sufficient number of training examples are generated (we used thousands in our experiments), the set of all stored examples is input to a machine learning algorithm, which uses the training examples to learn a relevance model (see subsection 2). After learning, the method proceeds to the 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 report attribute values are calculated and the learned relevance model is used to decide when the warning should be turned on. Note that it is possible to continue to learn in the usage stage. However, because the EEBL warnings will affect drivers' behavior, this may introduce a bias in the learning. We will therefore leave this issue as part of future work. We will now discuss the attribute selection in the next subsection and then provide an overview of the machine learning methods we utilized.

1) Attributes

The attributes used for the reports are determined based on factors that affect the drivers' decision to initiate emergency deceleration. In this paper, we identify three such attributes: distance (d), difference in velocities (vd), and density (ρ). We define distance as the time needed for the receiving vehicle, traveling at its current velocity, to reach the point at which the report was generated. Difference in velocities is the difference between the velocities of the receiving and the reporting vehicles. Density is the number of vehicles on the road, divided by the road length. Note that additional attributes, such as those related to weather or road conditions, may also be used to augment the learned relevance model.

2) Machine Learning Methods

In this subsection, we discuss two methods that we have used for learning.

a) Naïve Bayes

The Naïve Bayes method is a simple method for learning based on probabilities. It assumes conditional independence among the attributes and uses the Bayes' rule for calculating the probabilities. The algorithm finds a mapping of the attributes to the probability that the driver will perform emergency deceleration within *Reaction-delay* seconds after report is received. We label the conditional probability that a vehicle v performs emergency braking, given a report with attributes ρ, d, and vd, as $P(v_{br}|\rho,d,vd)$ and the unconditional probability that a vehicle v performs emergency deceleration as $P(v_{br})$. Using Bayes' rule, the conditional independence assumption, and the law of total probability, the probability of relevance is then calculated as:

$$P(v_{br}|\rho,d,vd) = \frac{P(\rho|v_{br})\cdot P(d|v_{br})\cdot P(vd|v_{br})\cdot P(v_{br})}{\begin{gathered}(P(\rho|v_{br})\cdot P(d|v_{br})\cdot P(vd|v_{br})\cdot P(v_{br})\\ +P(\rho|v_{br'})\cdot P(d|v_{br'})\cdot P(vd|v_{br'})\cdot P(v_{br'}))\end{gathered}} \tag{2}$$

In this equation, $P(v_{br'})$ is the unconditional probability of not emergency braking, and can be calculated as $(1\text{-}P(v_{br}))$. $P(v_{br})$ can be calculated by counting all instances of emergency braking over time. We additionally have to find the conditional probabilities, $P(\rho|v_{br})$, $P(d|v_{br})$, $P(vd|v_{br})$, $P(\rho|v_{br'})$, $P(vd|v_{br'})$, and $P(d|v_{br'})$. We make the assumption that these follow a normal distribution and hence estimate the probabilities by maintaining sample mean and standard deviations for each, using the training examples as samples. We thus have to maintain two parameters for each of the six distributions: μ, the mean, and σ, the standard deviation.

These values can be calculated as follows. After receiving a report, the vehicle waits for *Reaction-delay* seconds since report was received. After this time, if the vehicle was, at any time, emergency decelerating, we update $P(\rho|v_{br})$, $P(d|v_{br})$, $P(vd|v_{br})$, otherwise, we update $P(\rho|v_{br'})$, $P(vd|v_{br'})$, and $P(d|v_{br'})$. The updates are performed by calculating a new mean and standard deviation.

b) Logistic Regression

This method assumes that the emergency braking probability fits a logistic function, $f(z)=1/(1+exp(-z))$, where z is a linear combination of the attributes ρ, d, vd: $z=\beta_0+\beta_1\rho+\beta_2 d+\beta_3 vd$. The parameters β_1, β_2, and β_3, are the coefficients of the attributes, and β_0 is the intercept value. The values of these parameters are found based on the training examples, typically using maximum likelihood methods (see [17] for details).

VI. Evaluation

The main goal of the evaluation was to measure the effectiveness of the EEBL system in preventing vehicular collisions and to measure the potential for driver desensitization to warnings. The evaluation was done through simulations using the MITSIM simulator [18], modified to enable vehicle collisions and to simulate the EEBL system. The evaluation compared the two relevance estimation methods: the analytic method (Analytic), and the machine learning method using Naïve Bayes (ML-NB) and logistic regression (ML-LR). Two baseline methods were also used: noEEBL, which never showed warnings, and simpleEEBL, which showed warnings for every received report. The next subsection will describe our modifications to MITSIM. Subsection B explains our simulation environment and subsection C provides the evaluation procedure. Results are shown in subsection D.

A. Modifications to MITSIM

Since the original software focused mainly on evaluating traffic flow, it did not permit vehicle collisions. To prevent collisions, MITSIM allowed vehicles, in certain cases, to exceed their specified maximum deceleration. We therefore modified MITSIM by making sure maximum deceleration cannot be exceeded at any time during car following. As a result, vehicles may not have enough time to stop and collisions can occur. We also implemented the ability to stop one of the vehicles immediately, which allowed us to simulate a worst case emergency braking situation.

The EEBL system was implemented by checking, during every time step, the deceleration rate of all vehicles. If one of the vehicles reached the maximum deceleration rate, the EEBL report was then sent to all following vehicles in the same time step. All vehicles which received the report would resent it in the same time step and the relevance estimator would decide whether a warning would be given. If a warning was given, the vehicle, after some driver reaction time, decelerated with maximum force until stopped.

B. Simulation Environment

The environment consisted of a 3 mile road with a single lane. Vehicles entered the road at a rate specified by the mean departure rate, which is a random number from a normal distribution and is fixed for the duration of a single simulation run. The inter-vehicle spacing is Poisson distributed with mean rate equal to the mean departure rate. The speed limit and free-flow speeds on the road are both set to 55mph. The driver reaction times were set according to values in [16]. *Tsevere_brake* was set to the MITSIM specified maximum deceleration. *TWarning* was set to 0.5. All other parameters were set to their MITSIM defaults.

Each run simulated a single incident on the road, in which a particular vehicle, we call the *incident vehicle*, stopped immediately after it traversed 99% of the road length. The vehicle selected to stop was always the 100th vehicle entered onto the road. This was done to allow the simulation to initialize. However, we have tested using the 1st, 10th, 50th, and 200th vehicle instead of the 100th as the incident vehicle, but the results did not change. The incident initiated a possible sequence of collisions. For each run, we recorded the number of collisions in the last 2 miles of the road. The first mile is not considered, because collisions at the road entrance are typically unrealistic due to the method MITSIM uses for loading vehicles onto the road. In order to focus on the collisions that happen after the incident, the EEBL system was initially turned off and then turned on after the incident occurred.

C. *Evaluation Procedure*

The evaluation of our relevance estimation method was done in several steps. First, we ran 2000 simulation runs without EEBL to gather training examples. For these runs, the departure rate was set to a mean of 800 vehicles per hour, and the standard deviation to 400. Examples from the first 1000 runs were used for learning, done through the Weka Learning Toolkit [19], using the NaiveBayesSimple and Logistic implementations with default parameter values. The remaining examples were used for testing the false warning rate with the different relevance estimation methods. We defined the *false warning rate* as the number of false warnings, divided by the total number of warnings shown. A false warning was counted every time the relevance estimation method would have given a warning for the particular set of attribute values, while the driver would not have performed emergency deceleration within the *Reaction-delay* seconds of report generation. We used the false warning rate as a means of testing the potential driver desensitization that can occur. A high false warnings rate will increase the possibility of driver desensitization, because when a warning is shown in cases where the driver did not feel it was needed, he or she will be less likely to react to the warning the next time it is shown.

After learning the relevance models, we then ran 1000 simulation runs with EEBL enabled, and repeated the runs using varying departure rates, participation rates, and transmission ranges. For every combination of parameter values, the average number of collisions per run was then calculated.

D. *Results*

1) *False Warning Rate*

The results of the false warning rate tests are shown in fig. 2. Both the simpleEEBL and analytic methods had similar false warning rates of 67% and 65%, respectively. Intuitively, such high rates of false warnings will eventually desensitize drivers and reduce the safety benefit of EEBL. In contrast, ML-NB achieved a much lower false warning rate of 39%. ML-LR had an even lower rate of 24%. The reason that the analytic method had a high false warning rate was because the formulas used to calculate the minimal safety gap are based on assumptions about driver behavior and approximations of vehicle positions that may not be valid.

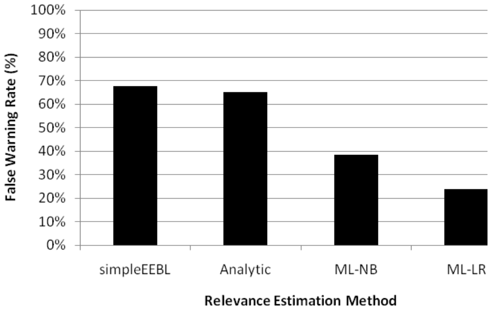

Figure 2. False warning rate for different relevance estimation methods.

2) *Number of Collisions*

The results showing the average number of collisions per for different parameter values are shown in fig. 3 and 4. For results shown in fig. 3, we varied the departure rate, while keeping the transmission range set to infinity, and the participation rate set to 100%. With the noEEBL method, the average number of collisions decreased considerably with increasing departure rates. This behavior can be attributed to the fact that at low densities, vehicles travel at much higher velocities, which makes it harder for the vehicle to stop and avoid a collision with a stopped vehicle. Averaging over all departure rates, noEEBL averaged 28.16 collisions per run. Through the use of EEBL, vehicular collisions were nearly avoided at all departure rates, with an average of 0.260 collisions per run for the simpleEEBL method. The reason for the large difference between the number of collisions using noEEBL and simpleEEBL is that the tested scenario creates a chain of vehicular collisions which is avoided using EEBL. On average the machine learning method using naïve Bayes (ML-NB) achieved the best performance, with 0.256 average collisions, but this was not much lower than the simpleEEBL, analytic (0.291), or the ML-LR (0.301) methods. Note that in some cases, the machine learning or analytic methods may prevent more collisions than simpleEEBL, because the simple method may cause unnecessary emergency deceleration. This unnecessary deceleration can then cause additional collisions that would not otherwise happen without the EEBL warning.

In real life, the wireless communication technologies such as DSRC typically have a limited transmission range, so therefore only a subset of vehicles (those within range) can receive the reports. Additionally, not all vehicles may be equipped with EEBL. In fig. 4, we show what happens when the transmission range (TR) is limited and not all vehicles are equipped with EEBL, using ML-NB and simpleEEBL methods. For these tests, the departure rate was fixed to 600 veh/hr. Results showed no significant difference between any of the EEBL methods as the participation rate or transmission range varied. Also, the results indicate a considerable decrease in the number of collisions even with a low participation rate. The implication is that obtaining a safety benefit from the EEBL system may not require a large vehicle participation rate.

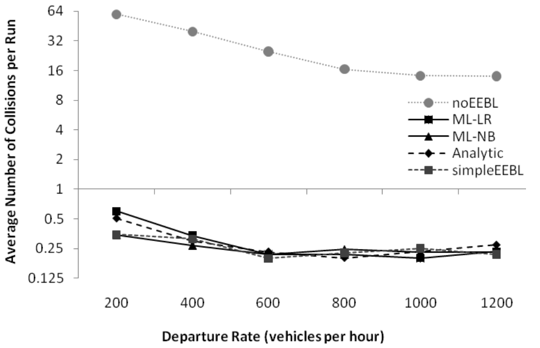

Figure 3. Average number of collisions for different relevance estimation methods and vehicle departure rates.

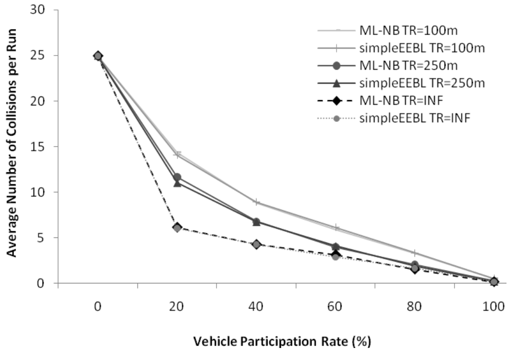

Figure 4. Average number of collisions vs. vehicle participation rate at limited transmission range (TR).

VII. Conclusion

In this paper we compared two methods for estimating relevance of emergency brake reports in an EEBL system. One used an analytic approach based on the minimal safety gap to avoid a collision. The other method used a machine learning approach. The two methods were evaluated in simulations using the MITSIM software. Tests showed that the simpleEEBL method, which always displayed a warning to drivers when a report arrives, significantly lowered the number of collisions on the road. The results also showed that both the analytic and the machine learning methods achieved a similar number of collisions as the simpleEEBL. However, the machine learning method significantly reduced the number of false warnings given to drivers in comparison to all other methods, which reduced the possibility of driver desensitization to warnings. Additional tests showed the impacts of participation rate and limited transmission range on the average number of collisions.

Acknowledgment

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

References

[1] A. Carter, "The status of vehicle-to-vehicle communication as a means of improving crash prevention performance," Tech. Rep. 05-0264, NHTSA, 2005.

[2] 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, 39, pp. 216-229.

[3] Y. Zang, L. Stibor, H. J. Reumerman, and H. Chen, "Wireless local danger warning using inter-vehicle communications in highway scenarios," Proc. *14th European Wireless Conference,* June 2008, pp.1-7, 22-25.

[4] B. van Arem, C. M. J. Tampère, and K. M. Malone, "Modeling traffic flows with intelligent cars and intelligent roads," *Proc. IEEE Intell. Vehicles Symp.*, Columbus, OH, Jun. 2003, pp. 456–461.

[5] R. Miller and H. Qingfeng, "An adaptive peer-to-peer collision warning system," Vehicular Technology Conference, 2002. VTC Spring 2002. IEEE 55th , vol.1, pp. 317- 321.

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

[7] P. Szczurek, B. Xu, J. Lin, and O. Wolfson, "Machine Learning Approach to Report Prioritization with an Application to Travel Time Dissemination," *Proc. of The Second International Workshop on Computational Transportation Science*, 2009, pp.31-36.

[8] P. Szczurek, B. Xu, J. Lin, and O. Wolfson, "Spatio-temporal Information Ranking in VANET Applications," International Journal of Next-Generation Computing. 1, 1 (2010).

[9] C. Manasseh, Y. P. Fallah, R. Sengupta, and J. Misener, "Learning User Perception To Traveler Situation Awareness Alerts on Mobile Devices", unpublished.

[10] C. Manasseh, Y. P. Fallah, R. Sengupta, and J. Misener, "Using Smartphones to Enable Situation Awareness on Highways", Proc. of Intelligent Transportation Society of America annual conf. ITSA 2010, in press.

[11] M. M. Muntzinger, S. Zuther, and K. Dietmayer, "Probability estimation for an automotive Pre-Crash application with short filter settling times," *Proc. IEEE Intelligent Vehicles Symposium,* 3-5 June 2009, pp.411-416.

[12] Z. Sun, R. Miller, G. Bebis, and D. DiMeo, "A real-time precrash vehicle detection system," Proc. *Sixth IEEE Workshop on Applications of Computer Vision (WACV), 2002*, pp. 171- 176.

[13] S. Kato, S. Tsugawa, K. Tokuda, T. Matsui, and H. Fujii, "Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications," *IEEE Transactions on Intelligent Transportation Systems,* vol.3, no.3, pp. 155- 161, Sep 2002.

[14] G. Naus, R. Vugts, J. Ploeg, R. Vd Molengraft, and M. Steinbuch, "Towards On-The-Road Implementation Of Cooperative Adaptive Cruise Control," 16th ITS World Congress, 2009. Sep. 21-25th.

[15] S. Panichpapiboon and W. Pattara-atikom "Evaluation of a Neighbor-based Vehicle Density Estimation Scheme," *IEEE International Conference on ITS Telecommunications (ITST'08)*, Phuket, Thailand, Oct. 2008, pp. 294-298.

[16] G. Johansson and K. Rummer, "Drivers' brake reaction time," Human Factors, 13(1):23, February 1971.

[17] S. Le Cessie and J. C. van Houwelingen, J.C., "Ridge Estimators in Logistic Regression", Applied Statistics, 1992, 41(1):191-201.

[18] 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, issue 3, June 1996, pgs. 113-129.

[19] I. H. Witten and E. Frank. Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.