Work in Progress: Application of Secondary Information for Misbehavior Detection in VANETs[*]

Ashish Vulimiri[1], Arobinda Gupta[1], Pramit Roy[1], Skanda N. Muthaiah[2], and Arzad A. Kherani[2]

[1] Indian Institute of Technology, Kharagpur - 721302, India
`agupta@cse.iitkgp.ernet.in`
[2] GM India Science Lab, Bangalore, India

Abstract. Safety applications designed for Vehicular Ad Hoc Networks (VANETs) can be compromised by participating vehicles transmitting false or inaccurate information. Design of mechanisms that detect such misbehaving nodes is an important problem in VANETs. In this paper, we investigate the use of correlated information, called "*secondary* alerts", generated in response to another alert, called as the "*primary* alert" to verify the truth or falsity of the *primary* alert received by a vehicle. We first propose a framework to model how such correlated *secondary* information observed from more than one source can be integrated to generate a "degree of belief" for the *primary* alert. We then show an instantiation of the model proposed for the specific case of Post-Crash Notification as the *primary* alert and Slow/Stopped Vehicle Advisory as the *secondary* alerts. Finally, we present the design and evaluation of a misbehavior detection scheme (MDS) for PCN application using such correlated information to illustrate that such information can be used efficiently for MDS design.

Key words: VANET, Misbehavior detection, Primary alert, Secondary information.

1 Introduction

A vehicular ad hoc network (VANET) is an ad hoc wireless communication system setup between multiple vehicles in a neighborhood. The communication can either be between vehicle-to-vehicle (V2V) or may also involve some roadside infrastructures in which case it is termed as vehicle-to-infrastructure (V2I) communication. Several applications such as safety, traffic aid, infotainment, financial and navigational aid [1] etc., have been proposed for use in VANETs.

Typically, a V2X[3] based safety application triggers an alert in response to a specific event. For example, a crashed vehicle may trigger a

---

[3] A generic term used to refer to V2V or V2I.

Post Crash Notification (PCN) alert, a vehicle braking hard may trigger an Emergency Electronic Brake Light (EEBL) alert, or a deceleration beyond a certain threshold may trigger a Slow or Stopped Vehicle Advisory (SVA) alert. In the presence of misbehaving vehicles, such alerts can also be raised even if the corresponding event has not happened, or the information sent in the alert can be wrong. For example a misbehaving vehicle may raise a PCN alert even in the absence of a crash, or report false information about the position of the crash. Note that authentication schemes are not sufficient to handle this as even authenticated users can turn malicious or misbehave due to faulty modules. Hence, upon reception of an alert from another vehicle, the receiving vehicle needs to verify if the event corresponding to a particular alert is true or false. This process will be referred to as *misbehavior detection.* Misbehavior detection is particularly important since it is expected that if such alerts are shown to a driver, the driver will respond to these alerts by taking some necessary action. A false action taken can have serious implications on the safety of drivers. On detecting a misbehavior using some Misbehavior Detection Scheme (MDS), a vehicle's On Board Unit (OBU) reports the misbehavior to a backend *Certificate Authority (CA)* so that the misbehaving vehicle's certificate may be revoked and the vehicle evicted from the network by the CA. In the rest of this paper, the alert whose truth or falsity is to be verified is termed as the *primary alert.*

The event corresponding to a primary alert may have effects on its neighboring vehicles and can cause other alerts to be sent. As an example, if a PCN alert is true, i.e., a vehicle has indeed crashed, it may cause other vehicles nearby to slow down for some time, possibly causing a series of SVA or EEBL alerts to be transmitted. Hence, the receipt of an appropriate number of SVA alerts following the reception of an PCN alert may be useful to strengthen a node's belief in the truth of the PCN alert and vice-versa. Such correlated information is termed as *secondary* information in this paper.

This work investigates the use of such information correlated to or generated due to a *primary* alert in designing misbehavior detection schemes. Specifically, we propose a probabilistic framework that models the use of secondary information for misbehavior detection. We then instantiate the model with a specific example of PCN and SVA alerts, and discuss how some of the necessary probabilities can be estimated empirically. We finally present the design and evaluation of a Misbehavior Detection Scheme (MDS) for PCN alerts to show that the probability estimations can be used effectively for misbehavior detection. Note that these secondary information will be generated irrespective of whether a vehicle is doing misbehavior detection or not, and are not caused by the misbehavior detection process of any vehicle.

2 Related Works

The need for security in VANET applications has been well-established, and several works [2][3][4][5] investigate the requirements and challenges involved in securing V2V communications. The IEEE 1609.2 standard

[6] defines the functionalities of a security layer in V2V communication. However, though some of these works stress the need for misbehavior detection, no specific scheme for misbehavior detection is given for any application.

Among applications that can benefit from MDS, [7] and [8] discuss collision warning systems, both cooperative and autonomous. However, none of these papers propose any misbehavior detection scheme. Golle et al. [9] present a model to integrate information from different sensors and use it to identify malicious information and malicious nodes. However, the algorithm for actual detection is only sketched through examples. Also, the information used is local to within the vehicle and secondary information from other vehicles is not considered. A similar idea has been proposed by Schmidt et. al. [10]. They have given a trust calculation based scheme where a car calculates the trust in vehicles in its near vicinity from the various sensor values obtained. The paper lists only the scheme and its performance in various road scenarios have not been investigated. Ghosh et al. [11][12] have proposed misbehavior detection schemes for PCN application; however their schemes also do not consider any secondary information.

3   Model for Integrating Secondary Information

The event corresponding to a primary alert can cause other secondary events to occur, which can be detected by the same vehicle receiving the primary alert. Occurrence of such secondary events can act as supporting evidence in support of the primary alert. On the other hand, lack of such secondary events following a primary alert can indicate that the event corresponding to an alert may be false. The degree to which the available evidence supports the veracity of the primary alert (which we shall henceforth refer to as "degree of belief") is quantified by a numerical value β, $\{\beta \in [0, 1]\}$. The degree of belief is 1 when we are certain that the event corresponding to the primary alert has occurred, and is 0 when we are certain that it has not occurred.

In the rest of this section, we first define events and their attributes more formally and propose a model to define the probability of the primary alert being true given an individual secondary information. We then propose a combining rule to obtain a final degree of belief β for the primary alert given all the secondary information present.

Event Model. An "event" is any observation that provides some information about the likelihood of the primary alert. The set of all events is partitioned into a finite number of "event classes", each class being some category of events. Each event class is characterized by a set of attributes that define the information content of each event in the class. If $\{a_1, a_2, \ldots, a_n\}$ is the set of attributes defining an event class, any single event in that class is merely an assignment of values to these attributes. The following two attributes are common to all event classes: t, the time of occurrence of an event and (x, y), the position of the source of the event.

We now formally define event classes. An event class $\mathcal{E}$ is an ordered tuple $< D, E, W >$ consisting of:

$D = D_1 \times D_2 \times \cdots \times D_n$**:** The set of all possible events. The D_i are the domains of each attribute in the event class. In the model considered in this work, the attribute space is composed of the following:

    – $D1$: Time t: denotes the time at which the event happens.

    – $D2, D3$: Locations x, y: denote the x, y coordinates at which the event occurs (we consider a 2D road topology for ease of understanding).

    – u: $D_4, D_5, \ldots, D_n$: denotes the values of all other attributes

    We assume a model where the domains of t, x, y and u are discrete. The values of these attributes then define a 4-dimensional grid.

$R_E(t, x, y, u)$**:** A function that maps to each distinct tuple $\langle t, x, y, u \rangle$ a binary random variable that indicates if an event with attribute value u is generated by a car at time t at the position (x, y). The range of each random variable is $\{0, 1\}$. We denote by $GRID$ an observation of 0 or 1 values for each $R_E(t, x, y, u)$ in the event class. $GRID(t, x, y, u)$ is 1 if the event has been observed at the grid location $\langle t, x, y, u \rangle$, and 0 otherwise.

$W(t, x, y, u)$**:** A function that assigns a numeric weight indicating relative importance to each grid element. We impose the condition that

$$\left(\sum_t \sum_{(x,y)} \sum_u W(t, x, y, u) \right) = 1$$

Let $R_{prim}(t, x, y, v)$ denote a function that defines binary random variables indicating if the event corresponding to a primary alert with value v occurs at (x, y) at time t.

Probability of Events. Let t_a, x_a, y_a, v_a denote the time of occurrence, (x, y) coordinates, and the value of the primary alert. We wish to define a scheme that would allow the OBU to compute the posterior probability of the primary event being true, given the description of the secondary information (the set of alerts) received over the network. We compute this probability in several steps. First, we take each event class, and compute the probability of the event corresponding to the primary alert (we refer to this event as the *primary event* in the rest of this paper) being true given all observed events of this class. We then combine these probabilities computed for each class using a combining rule to get a final probability over all event classes. This final probability is the required "degree of belief" β.

Let the posterior probability of the primary event being true given all events of a single event class $\mathcal{E}$ be denoted by $\beta_{\mathcal{E}}$. By definition,

$$\beta_{\mathcal{E}}(t_a, x_a, y_a, v_a) = P(R_{prim}(t_a, x_a, y_a, v_a) = 1 | GRID) \qquad (1)$$

To compute this, we first compute the posterior probability of each individual event, and then combine them using a weight function W for

the event class to get a single posterior probability as follows.

$$P(R_{prim}(t_a, x_a, y_a, v_a) = 1 | GRID)$$
$$= \sum_t \sum_{(x,y)} \sum_u \{W(t, x, y, u) \times IND(t, x, y, u)\} \quad \text{where} \tag{2}$$
$$IND(t, x, y, u)$$
$$= P(R_{prim}(t_a, x_a, y_a, v_a) = 1 | R_E(t, x, y, u) = GRID(t, x, y, u))$$

The probability $P(R_{prim}(t_a, x_a, y_a, v_a) = 1 | R_E(t, x, y, u) = 1)$ is the probability of the primary event being true given the single random variable $R_E(t, x, y, u)$. Using Bayes' rule,

$$P(R_{prim}(t_a, x_a, y_a, v_a) = 1 | R_E(t, x, y, u) = 1) =$$
$$\frac{P(R_{prim}(t_a, x_a, y_a, v_a) = 1) \times P(R_E(t, x, y, u) = 1 | R_{prim}(t_a, x_a, y_a, v_a) = 1)}{P(R_E(t, x, y, u) = 1)}$$
$$\tag{3}$$

Thus, for each secondary event, a probability that the primary event has happened given the secondary event has happened can be calculated in terms of the probabilities on the right hand side. (We will later discuss and show an example of how all the probabilities on the right hand side can be estimated). The function W is an application specific weight assignment that reflects the relative importance assigned to each grid location. The highest weight would be assigned to the cells where the occurrence of an event provides the strongest evidence for or against the truth of the primary alert.

Computing β. In order to combine the $\beta_{\mathcal{E}}$ for all the event classes into a single estimate β of the value of the alert, a weighted combination is used.

$$\beta(t_a, x_a, y_a, v_a) = \sum_{\mathcal{E}} w_{\mathcal{E}} \beta_{\mathcal{E}}(t_a, x_a, y_a, v_a) \tag{4}$$

where $\sum_{\mathcal{E}} w_{\mathcal{E}} = 1$. The weight $w_{\mathcal{E}}$ assigned to a class $\mathcal{E}$ indicates the relative trust placed in that class with respect to all the other classes.

We thus have a method that allows us to compute the probability of truth or falsity of a primary alert given the secondary information observed. However, the probabilities defined above depend on several factors like the congestion model, the mobility models of the cars, the actual primary and secondary events considered and the nature of the correlation between them, and the chance of the safety condition occurring. Hence these probabilities may be hard to obtain analytically. In the next section, we show how this model may be developed for a specific application scenario and show how these probabilities may be estimated empirically through simulation.

4   An Example: PCN and SVA Alerts

We consider the PCN application and demonstrate how the required probabilities may be estimated in a system consisting of two classes of

events: (i) the class containing the PCN alert, and (ii) the class containing the SVA alerts. The PCN alert is the primary event, and the secondary information is comprised of zero or more SVA alerts raised.

The PCN event class consists of a single event, the first PCN notification. This event has no distinct attributes apart from t, the time at which the alert was raised, and (x, y), the location of the crash site. Let $CRASH(t, x, y)$ denote the random variable corresponding to the primary event $(R_{prim}(t, x, y, true))$.

The SVA event class also consists of a single event, an SVA alert. An SVA alert may originate from a vehicle at any distance to the crash site. This event class again has no distinct event attributes other than t and (x, y). Let the random variables R_E be denoted by SVA. The range of x is $\{1, 2, \ldots, D\}$ for some $D > 0$, and that of y is $\{1, 2, \ldots, l\}$, l being the number of lanes on the road.

The component probabilities needed to compute the final estimate $\beta(true)$ of the PCN alert being true are estimated from a combination of historical data of driver behavior and experimental results obtained by simulating different crash scenarios.

$P(CRASH(t_a, x_a, y_a) = 1)$:

> In general, this depends on the congestion on the road and on driving habits. This can be estimated for a given time duration and road segment based on historical data collected by the relevant authorities, such as state or national transportation authorities. Note that crashes are not very frequent in general, and the probability value can be taken to be low even in a congested scenario.

$P(SVA(t, x, y) = 1 | CRASH(t_a, x_a, y_a) = 1)$:

> These values may be obtained by simulating a crash at time t_a at location (x_a, y_a) using a vehicular traffic simulator, and then recording the average number of SVA alerts generated at each grid cell (t, x, y) per simulation run.

$P(SVA(t, x, y) = 1)$:

> Note that an SVA alert might be raised even when there is no crash due to road congestion. Applying Bayes' rule, the required probability is given by

$$
\begin{aligned}
P(SVA&(t, x, y) = 1) \\
&= P(SVA(t, x, y) = 1 | NOCRASH) \times P(NOCRASH) \\
&+ \sum_{t, x, y} P(SVA(t, x, y) = 1 | CRASH(t, x, y) = 1) \\
&\qquad\qquad \times P(CRASH(t, x, y) = 1)
\end{aligned}
\tag{5}
$$

> Thus, the probability value can be obtained by simulating the no-crash as well as various crash scenarios and observing the average number of SVA alerts at each grid cell.

$W(t, x, y)$:

> The weights may be obtained by simulating the crash and no-crash scenarios and computing the average number of extra alerts generated at each cell in the crash scenario. Multiple approaches might be taken to obtain the weight values from this information. One strategy might be to assign the weights in the proportion of the alerts

raised, with the highest weight assigned to the cell generating the most alerts. An alternate approach might be to fix a certain threshold based on the shape of the alert distribution across the grid, assign a weight 0 to all the cells having fewer alerts than this threshold, and distribute the total weight equally among the remaining cells. We emphasize again that this is just one possible choice of W and many other variations are possible.

5 A Misbehavior Detection Scheme for PCN Alerts

In this section, we show how an MDS for PCN alerts can be built from the estimated probabilities. In order to estimate the probabilities, we simulated a number of traffic scenarios using the mobility simulator *Vanet-MobiSim* [13]. The simulator was modified to allow cars to be stopped at any time, thus simulating a crash, and to collect traces of the points of time when the conditions for raising an alert were satisfied in a certain car. In order to simulate a crash at time t_a, a car was forced to stop at time t_a, and the behavior of the other vehicles on the road was observed for time duration T_{af}.

The traffic scenario simulated to estimate the probabilities involved three groups of 50 cars each, with speeds between 30-50, 40-60, and 60-100 km/hour respectively. The minimum inter-vehicle separation is set at 2m, and the *politeness factor*, which controls the aggressiveness of the drivers in enforcing a lane-change in VanetMobiSim, is set to 1.0, 0.7, and 0.3 respectively (higher speed drivers have lower politeness factors, and hence are more aggressive in their lane-changing behavior). The speed for raising the SVA alert was set to 20 km/hour, meaning that a car raised an SVA alert if its speed fell below 20 km/hour. A random car was made to stop in the middle of the simulation and the alerts generated in its immediate neighborhood were noted for the next $T_{af} = 25$ time slots, each of duration 10 seconds. The road segment observed was 25 slots of length 10 meters each from the crash site. We ignored the lane number, so that the observed grid is 2-dimensional representing the time from crash and distance from crash site $(t \times d)$ of size 25×25.

The number of alerts at each grid cell is recorded and these are averaged out over a large number of simulation runs to compute the average probabilities. The computed probabilities (of an SVA alert in a grid cell given that a crash has actually occurred) are shown in Figure 1.

The distributions of the average number of alerts over distance from the crash site and over time from the crash time are shown in Figure 2. As expected, it is seen that most of the SVA alerts happen close to the crash site. However, the number of SVA alerts very close to the crash time is low, increasing in number after some time as cars slow down, and then decreasing again if cars get time to change lanes or if number of cars decreases. The distribution of the alerts can be used to define the following simple weight function:

Weight Function W: *Let x_d denote the number of alerts at distance d from the crash site (cumulative over time) and x_t denote the number*

0.31	0.16	0.18	0.11	0.06	0.07	0.04	0.03	0.03	0.05	0.03	0.03	0.04	0.04	0.03	0.06	0.02	0.08	0.09	0.03	0.05	0.03	0.05	0.03	0.04
0.48	0.27	0.26	0.13	0.13	0.11	0.13	0.11	0.08	0.07	0.09	0.1	0.09	0.07	0.06	0.07	0.06	0.07	0.06	0.04	0.05	0.06	0.05	0.05	0.04
0.58	0.45	0.35	0.31	0.22	0.21	0.16	0.11	0.11	0.16	0.14	0.12	0.11	0.06	0.12	0.1	0.14	0.11	0.13	0.17	0.14	0.14	0.13	0.15	0.18
0.65	0.48	0.5	0.46	0.4	0.4	0.36	0.32	0.35	0.27	0.26	0.24	0.28	0.16	0.23	0.16	0.2	0.2	0.27	0.24	0.28	0.27	0.27	0.28	0.27
0.8	0.64	0.62	0.59	0.47	0.49	0.43	0.41	0.39	0.45	0.36	0.4	0.38	0.35	0.36	0.34	0.3	0.34	0.31	0.31	0.36	0.39	0.3	0.35	0.35
0.77	0.7	0.68	0.64	0.64	0.51	0.51	0.49	0.47	0.46	0.42	0.44	0.45	0.4	0.38	0.48	0.38	0.45	0.43	0.42	0.46	0.45	0.45	0.42	0.38
0.87	0.78	0.82	0.8	0.71	0.69	0.66	0.63	0.62	0.61	0.6	0.53	0.54	0.55	0.53	0.51	0.55	0.49	0.47	0.52	0.49	0.48	0.54	0.51	0.48
0.93	0.89	0.89	0.84	0.84	0.81	0.79	0.72	0.69	0.62	0.64	0.62	0.57	0.61	0.55	0.61	0.56	0.56	0.64	0.56	0.64	0.57	0.6	0.64	0.63
0.97	0.93	0.9	0.9	0.9	0.88	0.87	0.83	0.75	0.76	0.74	0.73	0.75	0.77	0.75	0.74	0.71	0.74	0.73	0.75	0.74	0.67	0.71	0.76	0.74
0.99	0.94	0.96	0.94	0.94	0.92	0.92	0.9	0.91	0.9	0.88	0.92	0.89	0.86	0.84	0.85	0.87	0.89	0.89	0.87	0.87	0.88	0.85	0.84	0.91
0.99	0.96	0.98	0.96	0.98	0.98	0.96	0.94	0.96	0.95	0.93	0.93	0.95	0.94	0.93	0.89	0.91	0.9	0.89	0.91	0.91	0.92	0.89	0.94	0.9
1	1	1	1	1	0.97	0.98	0.97	0.96	0.97	0.96	0.96	0.97	0.92	0.97	0.96	0.96	0.95	0.94	0.93	0.92	0.94	0.93	0.95	0.93
1	1	1	1	1	1	1	1	1	1	1	1	0.99	1	0.99	1	0.94	1	1	0.98	0.99	0.95	0.99	0.87	0.95
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.99	0.99	1	0.97	0.9	0.88
1	1	1	1	1	1	1	1	1	1	1	0.99	1	1	1	1	0.96	1	0.99	0.97	0.9	0.9	0.82	0.79	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.98	0.94	0.91	0.88	0.82	0.82	0.78	0.78		
1	1	1	1	1	1	1	1	1	1	1	1	1	0.98	0.96	0.91	0.86	0.82	0.81	0.8	0.78	0.76	0.73	0.68	
1	1	1	1	1	1	1	1	1	1	1	0.98	0.95	0.91	0.9	0.82	0.82	0.81	0.78	0.73	0.7	0.63	0.58	0.58	0.56
1	1	1	1	1	1	1	0.98	0.96	0.91	0.9	0.82	0.82	0.82	0.77	0.7	0.68	0.62	0.61	0.56	0.53	0.52	0.47	0.45	
1	1	1	1	1	0.99	0.96	0.91	0.85	0.82	0.82	0.82	0.74	0.71	0.67	0.66	0.59	0.56	0.55	0.5	0.48	0.44	0.35	0.28	0.25
1	1	1	0.95	0.91	0.89	0.82	0.82	0.81	0.74	0.7	0.7	0.63	0.59	0.56	0.55	0.53	0.48	0.47	0.35	0.3	0.23	0.21	0.18	0.14
1	0.94	0.91	0.82	0.82	0.79	0.74	0.74	0.7	0.69	0.59	0.57	0.55	0.54	0.51	0.45	0.35	0.3	0.3	0.19	0.16	0.12	0.09	0.09	0.08
0.91	0.82	0.82	0.77	0.74	0.73	0.69	0.57	0.57	0.55	0.54	0.52	0.45	0.42	0.35	0.28	0.23	0.16	0.12	0.08	0.04	0.08	0.04	0.01	0.01
0.82	0.74	0.74	0.7	0.65	0.57	0.55	0.55	0.51	0.48	0.42	0.33	0.28	0.25	0.19	0.12	0.09	0.08	0.04	0.04	0.03	0.01	0.01	0	0
0.74	0.68	0.65	0.57	0.55	0.53	0.49	0.45	0.36	0.28	0.25	0.19	0.12	0.11	0.1	0.08	0.04	0.04	0.01	0.01	0	0	0	0	0

Fig. 1. Probabilities of SVA alert in the presence of a crash

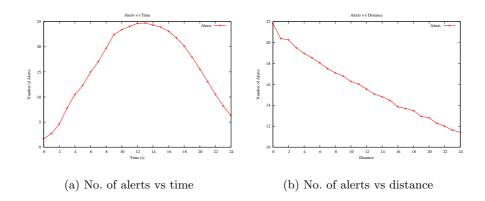(a) No. of alerts vs time   (b) No. of alerts vs distance

Fig. 2. Alert distributions over time and distance slots

of alerts at time t from the crash time (cumulative over distance). Let X^D be the maximum number of alerts in any distance slot (cumulative over time), and X^T be the maximum number of alerts in any time slot (cumulative over distance). First assign weight = 0 to any grid cell (d_i, t_i) if $x_{d_i} < \frac{2}{3}X^d$ or if $x_{t_i} < \frac{2}{3}X^t$. Let N be the remaining number of grid cells (which are not yet assigned weight 0). Assign weight $\frac{1}{N}$ to each of these grid cells.

Thus, the weight function chosen just assigns 0 weights to cells if the cell is at a larger distance from the crash site or if it is at time at which the total number of alerts is small. The rest of the cells are assigned equal weights. The weight function values for the individual grid cells are shown in Figure 3. We emphasize that this is just one possible specification of the weight function chosen for illustration, many other variations are possible.

Fig. 3. Weights assigned to grid elements using weight function W

Similarly, the same road segment is observed for the same duration in the absence of a crash. The probability of an SVA alert in each grid cell is then computed by averaging out the alerts received over a large number of simulation runs. It was noted that all the grid cells have a probability of 0, i.e., no SVA alerts were generated. Repeating the experiment for several other road segments gave similar results, some of which resulted in only a few SVA alerts.

The a-priori probability of a crash $P(CRASH(t_a, x_a, y_a) = 1)$ can only be estimated from observed data. We take it to be a small value, say, 0.001, indicating that a crash is reported in a particular place at particular time in around 1 in 1000 samples (for ex., a sample can be a day). Thus, we have now estimated $P(SVA(t, x, y) = 1 | CRASH(t_a, x_a, y_a) = 1)$ and $P(SVA(t, x, y) = 1 | NOCRASH)$. The a-priori probability of a SVA alert, $P(SVA(t, x, y) = 1)$, can now be easily computed noting that only one alert is considered and hence all but one of the terms in the summation in Equation 5 will be 0.

Hence, given any $GRID$ (i.e., an actual observation of 0-1 values for each of the random variables R_E corresponding to each grid cell), $P(R_{prim}(t_a, x_a, y_a, v_a) = 1 | R_E(t, x, y, u) = 1)$ for each grid cell can be computed by Equation 3, and the weight function can be fixed to allow aggregating them into a single value. This final value gives β, the belief in the truth or falsity of the PCN alert, with a value closer to 1 indicating a higher belief in the occurrence of the crash.

To validate the correctness of the estimation process and show that such estimation can actually be used in an MDS, we use the estimated probabilities and the weight function for detecting both a true alert and a false alert. We first simulate a crash in VanetMobiSim, and then observe the alert pattern over the grid. We then use the individual probabilities of each of the grid cells and the weights to compute a final degree of belief in the alert being true. This is repeated for 10 runs (i.e., 10 alert

patterns observed for 10 independent crashes) and the average degree of belief of these 10 runs is computed. This final degree of belief comes out to be 0.93. Note that the final degree of belief is high, indicating a high degree of belief that the crash actually occurred. This validates the correctness of the estimation process when a crash actually happens.

We repeated the same experiment with no crash in VanetMobiSim. The computed degree of belief comes out to be 0. This indicates that the MDS detects that the crash has not occurred (hence the alert is false), thereby validating the correctness of the estimation process when no crash actually happens.

Thus, it is seen that secondary information can be effectively used to design an MDS for PCN alerts. The false positive rate is seen to be 0 and the false negative rate is seen to be very low $(1 - 0.93 = 0.07)$ for the cases simulated.

5.1 Robustness of Estimation

In the above section, we have estimated the probabilities for a particular traffic model, and then applied it for misbehavior detection for scenarios following the same traffic model. However, in practice, the probabilities will be estimated once and stored in the OBU a-priori. The actual secondary information observed may come from a traffic scenario that is not exactly the same as the traffic model with which the probabilities were estimated. To evaluate the robustness of the proposed scheme over multiple traffic models, we perform the following experiments: We choose 4 traffic scenarios as shown in Figure 4.

Scenario 1	3 node groups each with 50 cars, with speeds 30-50 km/hr, 40-60 km/hr, 60-100 km/hr and politeness factors 1.0, 0.7, and 0.3 respectively
Scenario 2	1 node group with 100 cars, speed 30-50 km/hr, politeness factor 1.0
Scenario 3	1 node group with 100 cars, speed 40-60 km/hr, politeness factor 0.7
Scenario 4	1 node group with 100 cars, speed 60-100 km/hr, politeness factor 0.3

Fig. 4. Traffic scenarios simulated

Thus, Scenarios 2, 3, and 4 represent homogeneous drivers with progressively more aggressiveness, while Scenario 1 (which is the same as the scenario simulated earlier) is a mixture of the three. In the first set of experiments, we estimate the probabilities from simulating traffic from Scenario 1 only, and then apply it to detect misbehavior, if any, for traffic generated for all four scenarios. The final value reported is once again the average of 10 runs. The final value obtained for the 4 cases for both the cases of when there is a crash and when there is no crash is shown in Figure 5.

It is seen that the false positive rate is still 0 for all cases. However the false negative rate increases with increase in the aggressiveness of the

	When there is crash	When there is no crash
Scenario 1	0.93	0.0
Scenario 2	0.83	0.0
Scenario 3	0.59	0.0
Scenario 4	0.43	0.0

Fig. 5. Belief values obtained with probabilities estimated from Scenario 1

driver. This is expected as with higher politeness factor, drivers tend to change lane less often and hence, may have to slow down more and thus cause more SVA alerts. Thus, a majority of the SVA alerts in Scenario 1 are generated by cars with higher politeness factor, and hence the nature of alerts generated in Scenario 1 and Scenario 2 are similar. As the politeness factor decreases, the number of SVA alerts also decreases. This makes the scenario with which the probabilities are estimated very different than the scenario in which it is used for misbehavior detection, causing an increase in the false negatives.

The above arguments suggest that it may be better to estimate the probabilities from a scenario that represents a traffic model that is not very dissimilar with the traffic models for which misbehavior detection is needed. Hence, we next estimated the probabilities from Scenario 3, which intuitively represents a traffic model in between Scenario 2 and 4, and then applied it to detect misbehavior for all the four models. The final value obtained for the four cases for both the cases of when there is a crash and when there is no crash is shown in Figure 6.

	When there is crash	When there is no crash
Scenario 1	0.82	0.0
Scenario 2	0.94	0.0
Scenario 3	0.98	0.0
Scenario 4	0.98	0.0

Fig. 6. Belief values obtained with probabilities estimated from Scenario 3

It is seen that estimating the probabilities with Scenario 3 (a median scenario among the four) gives very good results for the MDS for detecting misbehaviors for all the scenarios. It is thus important to chose the right traffic model to estimate the probabilities.

6   Conclusion and Future Work

In this paper, an MDS based on a probabilistic framework of using correlated or secondary information to verify an alert in VANETs has been presented. An application of the proposed framework in identifying misbehavior for the specific case of PCN by using secondary information based on SVA alerts has been detailed and its performance evaluated.

This work can be extended in several directions. The PCN alert is a periodic alert and is sent periodically until the crash is cleared. Hence, the PCN alert that other vehicles receive may not be the first such alert

sent. Hence the grid that vehicles observe may be shifted in time from the grid that is considered in this framework. Extending our model to incorporate these observations is an interesting problem. Also, characterizing and designing MDS for other applications using secondary information is another important activity.

References

1. Bai, F., Krishnan, H., Sadekar, V., Holland, G., ElBatt, T.: Towards characterizing and classifying communication-based automotive applications from a wireless networking perspective. In: 1st IEEE Workshop on Automotive Networking and Applications. (2006)
2. Hubaux, J., Čapkun, S., Luo, J.: The security and privacy of smart vehicles. IEEE Security and Privacy **2**(3) (2004)
3. Raya, M., Papadimitratos, P., Hubaux, J.: Securing Vehicular Communications. IEEE Wireless Communications Magazine, Special Issue on Inter-Vehicular Communications **13**(5) (2006)
4. Torrent-Moreno, M., Killat, M., Hartenstein, H.: The challenges of robust inter-vehicle communications. In: IEEE 62nd Vehicular Technology Conference, Fall 2005. Volume 1. (2005)
5. Gerlach, M., Festag, A., Leinmuller, T., Goldacker, G., Harsch, C.: Security architecture for vehicular communication. In: 5th International Workshop On Intelligent Transportation. (2007)
6. In: IEEE Trial-Use Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages, IEEE Std 1609.2-2006. (2006)
7. Tan, H.S., Huang, J.: DGPS-based vehicle-to-vehicle cooperative collision warning: Engineering feasibility viewpoints. IEEE Transactions on Intelligent Transportation Systems **7**(4) (Dec. 2006)
8. ElBatt, T., Goel, S.K., Holland, G., Krishnan, H., Parikh, J.: Cooperative collision warning using dedicated short range wireless communications. In: VANET '06: 3rd ACM International Workshop on Vehicular Ad Hoc Networks. (2006)
9. Golle, P., Greene, D., Staddon, J.: Detecting and correcting malicious data in vanets. In: VANET '04: 1st ACM International Workshop on Vehicular Ad Hoc Networks. (2004)
10. Schmidt, R.K., Leinmüller, T., Schoch, E., Held, A., Schäfer, G.: Vehicle behavior analysis to enhance security in vanets. In: 4th Workshop on Vehicle to Vehicle Communications (V2VCOM 2008). (2008)
11. Ghosh, M., Varghese, A., Kherani, A., Gupta, A.: Distributed misbehavior detection in VANET. In: IEEE Wireless Communication and Networking Conference. (2006)
12. Ghosh, M., Varghese, A., Gupta, A., Kherani, A., Muthaiah, S.: Misbehavior detection scheme with integrated root cause detection in VANET. In: VANET'09, 6th ACM International Workshop on Vehicular Internetworking. (2009)
13. Härri, J., Filali, F., Bonnet, C., Fiore, M.: VanetMobiSim: generating realistic mobility patterns for VANETs. In: VANET'06, 3rd ACM International Workshop on Vehicular Ad Hoc Networks. (2006)