RD^4: Role-Differentiated Cooperative Deceptive Data Detection and Filtering in VANETs

Kewei Sha, *Member, IEEE*, Shinan Wang, and Weisong Shi, *Senior Member, IEEE*

***Abstract*—The data quality of collected sensing data, which determines the practical value of sensing systems, has been studied in several previous efforts; however, we argue that vehicular ad hoc networks (VANETs), which are a particular application of highly dynamic sensing systems, requires specific treatment to guarantee data quality. In this paper, we design a mechanism, i.e., RD^4, which is a role-differentiated cooperative deceptive data-detection and filtering mechanism, to detect the false data in VANETs. RD^4 is evaluated using an extended traffic simulator. Three scenarios, i.e., freeway, road construction on a highway, and a traffic light on a local street, are deployed in general. Evaluation results show that the proposed mechanism can achieve more than 90.00% recall and precision rate in most cases.**

***Index Terms*—Data quality, false detection, vehicle communication, vehicular ad hoc network (VANET), wireless sensor network.**

I. Introduction

With an increase in real deployments of wireless sensing systems [1], [2], we envision that the success of these systems is decided by the quality of the collected data [3]. Data quality is mainly affected by the deceptive data, including redundant and false data. Thus, the major concern of improving data quality is to detect and filter out deceptive data.

On the other side of spectrum, in the particular case of vehicular ad hoc networks (VANETs) [4], the conventional approaches of maintaining the data quality are inappropriate because two factors should simultaneously be satisfied. First, the mechanism must adapt to a frequently changing network topology. Second, sensor networks on vehicles underline the real-time requirement, which forces outlier detection, trust-based detection, and similar offline mechanisms to be insufficient. Although several methods have been proposed emphasizing the real-time functionality of the system [5], most of them assume a particular data model, e.g., Gaussian and linear. Others provide a real-time-fashion abnormal data-detection mechanism [6] but fail to be applied in VANETs since their approaches mainly rely on the network hierarchy.

In this paper, we propose a role-differentiated cooperative deceptive data-detection mechanism, i.e., RD^4, to detect and filter false data in VANETs. In RD^4, when a sensor is deployed, it picks up a role from the role set based on several sensing features. We will evaluate the efficiency and efficacy of our mechanism in three specific vehicular scenarios.

The rest of this paper is organized as follows. Section II denotes the motivation and describes the deceptive data-detection problem in general, as well as in VANETs, which is followed by the RD^4 mechanism in the particular scenario of VANETs in Section III. We design a VANET simulator and evaluate the performance of the RD^4 mechanism in Section IV. Related work can be found in Section V. Finally, conclusions and future work are discussed in Section VI.

II. Motivation and Problem Statement

VANET data suffer from two disadvantages. First, unreliable components generate unreliable data. Second, in a highly distributed system, data coming from all possible sources make the collected data even more untrustworthy since each individual sensor is easily being compromised. Based on the common features of collected sensing data, we summarize the sensor data by defining deceptive data and analyzing the specific scenario of VANETs.

A. Deceptive Data Definition

In this paper, we classify the deceptive data into two categories: *redundant data* and *false data*. Redundant data are defined as the data that share exactly the same or very similar information with the data reported previously or by other sensors. Another type of deceptive data is false data, which may result from the malfunction of the sensor board, unreliable wireless communication, and compromised sensors.

In the particular setting of VANETs, we are more interested in false data than in redundant data. First, the energy issue of sensors in vehicles is less affected or could even be ignored since sufficient power is able to be generated from the battery in each vehicle. Second, redundant data could be helpful in detecting false data since it provides additional information of the whole sensor network.

B. Insufficiency of Previous Approaches

Security technologies using traditional cryptography mechanisms such as encryption for confidentiality and hashing digestion for message integrity are employed. However, we argue that these technologies are necessary in detecting and filtering out deceptive data but are insufficient. Following are several

Manuscript received June 11, 2009; revised October 2, 2009. First published January 15, 2010; current version published March 19, 2010. This work was supported in part by the National Science Foundation under Grant CNS-0721456. The review of this paper was coordinated by Dr. P. Lin.

K. Sha is with Oklahoma City University, Oklahoma, OK 73106-1493 USA.

S. Wang and W. Shi are with the Department of Computer Science, Wayne State University, Detroit, MI 48202 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2010.2040400

reasons. First, rather than checking the data, most security-based approaches try to prevent attacks. To be specific, they try to validate the legitimacy of the reporting nodes instead of validating the legitimacy of the data. Thus, if the attacker is from legitimate but compromised vehicles, it is very difficult to detect and distinguish the attacker from a normal reporter. Second, in mobile sensing systems with high mobility, there is no permanent relationship between any two sensor nodes, therefore, verifying one another by using traditional security strategies like mutual authentication becomes challenging. Moreover, the large scale of the system and high mobility set up an obstacle in key distribution if a security-based approach is adopted. Finally, in such a totally distributed environment, all decisions should be made locally, including detecting deceptive data without the help of central servers, which makes the problem even worse when decisions should be made in a real-time fashion.

In addition to security technologies, reputation-based approaches, which usually require a strong identity, cannot work in this case because of the possible large scale, high mobility, and inadequate support from central servers. Several other previous efforts have also been made in deceptive data detection; however, most of them assume a specific distribution of the monitored parameter, which they use as a model to predict the missing values and to check the reported values. These methods can be useful techniques to detect deceptive data, but they rely too much on the distribution; therefore, they cannot generally be extended or applied to event-based sensor networks.

III. RD^4 in Vehicular Networks

In this section, we describe the RD^4 mechanism in the context of VANETs.

A. System Assumptions

To verify our proposed mechanism, the following assumptions are held, based on existing theoretical or practical products.

With the technology of short-range radio communication, such as dedicated short-range communication (DSRC) [7], vehicles are able to communicate with each other with various types of information, such as road conditions. We assume that each vehicle broadcasts information to nearby neighbors. For example, as mentioned in [8], every node could exchange information of vehicle speed and location with others. In addition, each vehicle could have a *received buffer*, which is in charge of verifying the received data from other vehicles based on RD^4.

In addition, we assume that every vehicle in the vehicular network holds a unique identification. Although Sybil attacks [9] are commonly encountered in VANETs, we argue that this type of attack is beyond the scope of our discussion. Sybil attacks could basically be described as a node that illegitimately fakes multiple identifications. There are existing security technologies [10] that handle Sybil attacks, while our approach mainly focuses on detecting deceptive data coming from unreliable system components, attacks from compromised nodes, and communication errors.

B. Role Definition in Vehicular Networks

The first step is defining a set of different roles in vehicular network applications. Considering various types of function components such as vehicles and roadside units (RSUs) in vehicular networks, we classify those function components into four roles: RSUs; public vehicles such as police cars, school buses, and so on; regular vehicles like personally owned cars; and the vehicle itself. Thus, the role set in the vehicular networks is defined as $R = \{R_{\mathrm{rsu}}, R_{\mathrm{pub}}, R_{\mathrm{reg}}, R_{\mathrm{self}}\}$.

For each event that the sensor senses or receives, it issues a confidence score to the event, which is denoted as $csr(E,T)_{ji}$, indicating the truth level of this event, where E specifies the event, and j and i are the identity of the role and the identity of the sensor, while T means the score will be valid for T time slots. We define the maximum confidence score that a sensor with ID i and role R_j can issue to a piece of data or an event report as $CSR(E,T)_{ji}$, which should satisfy $csr(E,T)_{ji} \leq CSR(E,T)_{ji}$. In our design, the definition of maximum confidence score (CSR) is closely related to the trustable level of each role in the system. For example, RSUs are mostly controlled by public organizations such as the Department of Transportation. Thus, we define the CSR_{ji} following the order of $CSR_{\mathrm{self},i} > CSR_{\mathrm{rsu},i} > CSR_{\mathrm{pub},i} > CSR_{\mathrm{reg},i}$.

C. False Accident Report Detection

In this paper, we assume that the detection of a true accident is handled by the accident sources, which can be either the vehicles involved in the accident or the police cars taking care of the accident. Except for those two types of vehicles, other vehicles are not supposed to report an accident. Malicious vehicles may insert false accident reports by acting like the vehicles involved in the accident. Thus, our goal is to remove false accident reports from malicious vehicles. In this paper, we assume that each vehicle is equipped with a tamper-proof component; therefore, even if a vehicle is compromised, it still cannot generate multiple identities.

When an accident report is received by a vehicle on the road, the vehicle will make a judgement about the truth of this report, based on the signal strength from its own observation and the signal strength reporting the same event from others. In our design, based on the reality that traffic will be blocked so that the vehicles will slow down when an accident happens, we use the vehicle velocity deceleration as the signal strength defined in the model: $p(t) = a(t) = dv/dt$, where $a(t)$ is the acceleration rate, and v is the velocity of the vehicle. Then, the accumulated signal strength observed by the vehicle j, i.e., $ASS(E,T_0)_{ji}$, can be defined as $ASS(E,T_0)_{ji} = \int_0^T a(t) = v_T - v_0 = \Delta v$, where T_0 is the timestamp. To emphasize the roles of vehicles, we assign to this important role a special ability to send out accident reports with a stronger signal strength; thus, after we calculate the accumulated signal strength observed by the vehicle itself, we will adjust the accumulated signal strength by considering the role of the vehicle. The new accumulated signal strength we get is

$$ASS(E,T_0)_{ji} = \frac{CSR(E,T)_{ji}}{CSR(E,T)_{\mathrm{reg},i}} \Delta v. \tag{1}$$

Except for the signal strength gained from the observation of the vehicle i, vehicle i will receive signal strength for the event E from other vehicles. If we integrate both types of signal strength of the event E, we get an integrated value of the accumulated signal strength of event E at vehicle i with timestamp T_0 as follows:

$$ASS(E,T_0)_{ji} = \sum_{n=0}^{N} W_{jn} * csr(E,T_0)_{jn} + \frac{CSR(E,T)_{ji}}{CSR(E,T)_{\text{reg},i}} \Delta v. \quad (2)$$

In the aforementioned equation, W_{jn} is the weight of the event report from sensor n of role j. It can be defined as the reverse of the distance between the vehicle reporting the event to the location of the event. The confidence scores from other vehicles $csr(E,T_0)_{jn}$ are based on their decision toward this event E. If they confirm this event, they will report this event by transforming the accumulated signal strength into the confidence score. The transformation will be demonstrated later in this section.

In general, if the accumulated signal strength exceeds the preset bound to confirm the event, the event is confirmed, and a new report about this event is forwarded. Otherwise, it will wait another T time slots to check the accumulated signal strength and recalculate the new confidence score. Considering the timeliness of events, in our design, the signal strength will degrade with the passage of time, with a fading rate of α. Hence, we can calculate the integrated value of the accumulated signal strength after a period of T time slots, which is specified as follows:

$$\begin{aligned} &ASS(E,T_0+T)_{ji} \\ &\quad = \alpha ASS(E,T_0)_{ji} + \sum_{n=0}^{N} W_{jn} * ASS(E,T_0+T)_{jn} \\ &\qquad + \frac{CSR(E,T)_{ji}}{CSR(E,T)_{\text{reg},i}} \left(v_{T_0+T} - v_{T_0}\right). \end{aligned} \quad (3)$$

Having the integrated value of the accumulated signal strength, we can map it to get a confidence score for the event E at the vehicle i as follows:

$$\begin{aligned} &csr(E,T)_{ji} \\ &\quad = f\left(ASS(E,T_0)_{ji}\right) \\ &\quad = \begin{cases} ASS(E,T_0)_{ji}, & ASS(E,T_0)_{ji} \le CSR(E,T)_{ji} \\ CSR(E,T)_{ji}, & ASS(E,T_0)_{ji} > CSR(E,T)_{ji}. \end{cases} \end{aligned} \quad (4)$$

Based on the confidence score, a final judgment on the truth of the event is generated as follows:

$$\text{Valid}(E) = \begin{cases} \text{True}, & csr(E,T)_{ji} > \theta \| csr(E,2T)_{ji} > \theta \\ \text{False}, & \text{otherwise}. \end{cases} \quad (5)$$

This formula means that the event is confirmed as true when the confidence score exceeds a preset threshold; otherwise, the event is confirmed as false, and the propagation of the event report will be terminated.

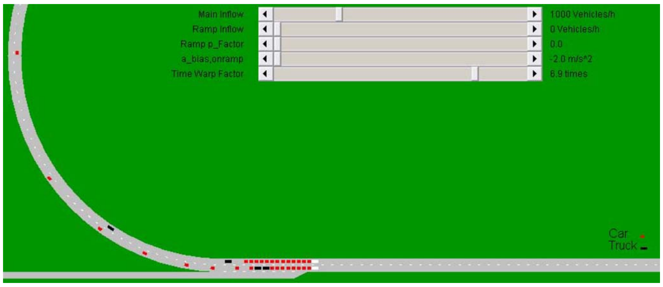

Main Inflow 1000 Vehicles/h
Ramp Inflow 0 Vehicles/h
Ramp p_Factor 0.0
a_bias,onramp -2.0 m/s^2
Time Warp Factor 6.9 times

Fig. 1. Snapshot of the vehicular network simulator.

IV. Performance Evaluation

We are in a position to evaluate the effectiveness and efficiency of the proposed mechanism. First, we will describe a VANET simulator, followed by an evaluation of our protocol in terms of several performance metrics, namely, recall, accident report propagation range, accident report confirmation time, detail on the freeway, road construction, and traffic light scenarios.

A. Freeway Scenario

First, in this section, we describe the simulator, implementation, simulation settings, and several performance metrics.

1) Simulation Setup: We design a simulator for VANETs by extending a traffic simulator designed in [11] and [12] that simulates the movements of the vehicles, such as acceleration, deceleration, and lane changing. The simulation of the RD^4 mechanism for VANETs is based on a road segment of a two-lane one-way highway scenario with an on-ramp. We implement a communication subsystem for the VANETs. In addition, we simulate the scenario of accidents, as well as malicious vehicles. Last but not the least, we fulfill the RD^4 mechanism to record and classify the report based on the mechanism in Section III. A snapshot of the simulator is shown in Fig. 1, where the red dots depict the regular vehicles, the black dots denote the public vehicles, and the white dots specify an accident.

In our experiments, the communication between the vehicles follows the DSRC with a maximum communication range of 200 m. For each vehicle, the speed limitation is 120 km/h. The road segment consists of a U-shape road with the length of 6575 m. When an accident appears, the road will be blocked for several minutes. The malicious vehicles will periodically broadcast a fake accident event if it does not detect a true event. Otherwise, it keeps silent. The percentage of malicious vehicles is around 10%, and there is a 10% chance that each one of them broadcasts a false message every 0.25 s. Meanwhile, the lifetime of each message is set to 2 s. We use 5.83 for the threshold in this situation, although there could be other values that provide an optimal solution for the mechanism.

2) Effectiveness of RD^4 in the Freeway Scenario: In this section, we show how effectively the RD^4 mechanism detects the false accident reports inserted by malicious vehicles and confirms the true accident reports generated by the vehicles involved in the accident. This property is evaluated by recall, which is defined as the fraction of the amount of a certain event reported that our mechanism classifies the reports as this event.

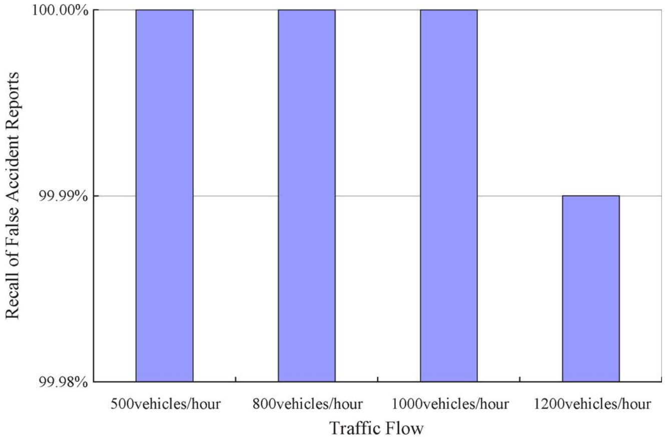


Fig. 2. Recall of false accident reports.

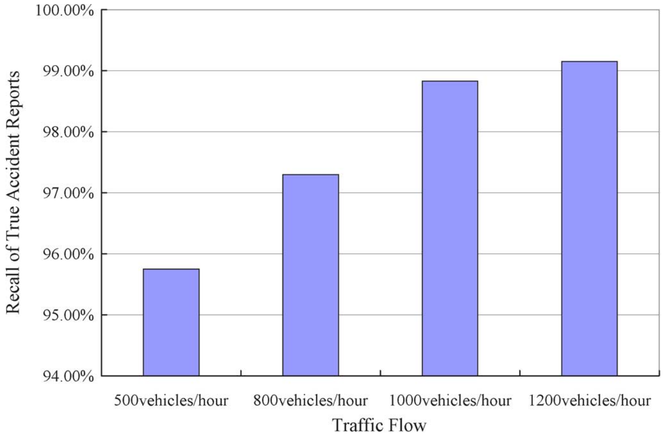


Fig. 3. Recall of true accident reports.

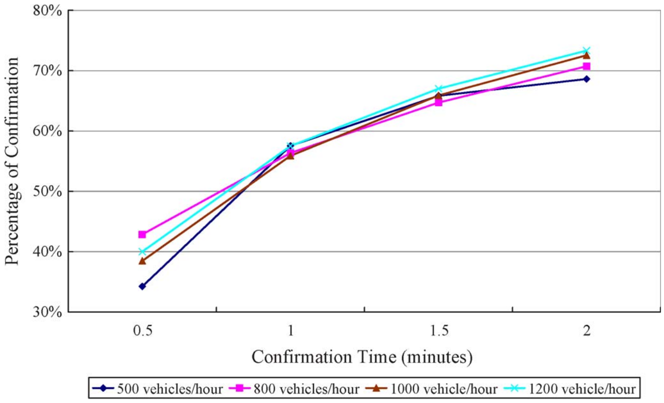


Fig. 4. Confirmation time of true accident reports.

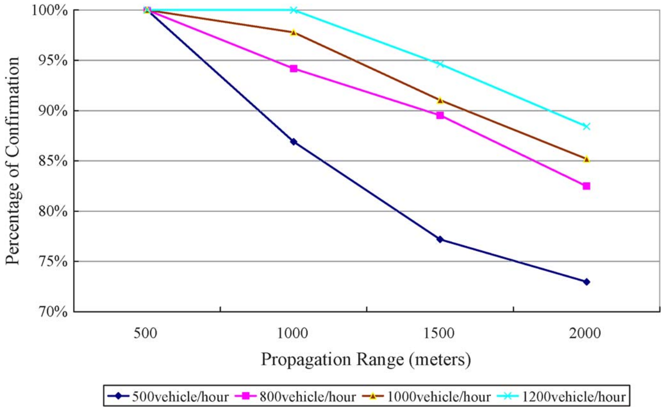


Fig. 5. Propagation range of true accident reports.

To be specific, in this application, the recall of false accident reports is defined as the fraction of the false accident reports that is detected by the RD^4 mechanism, which is shown in Fig. 2, while the recall of true accident reports is defined as the fraction of true accident reports confirmed by the RD^4 mechanism, which is shown in Fig. 3.

In Figs. 2 and 3, the x-axis is the traffic flow on the road, and the y-axis shows the recall. We can easily observe that the RD^4 mechanism detects 99.90% of false accident reports in most cases, and more than 95.70% of real accident reports are confirmed. We also find that the recall of false accident reports drops a little when the traffic flow grows, while the recall of true accident reports increases as the traffic flow increases because more traffic likely brings a lower average speed and strong signal strength, which is helpful when confirming a true accident. However, on the other hand, a lower speed also ruins the accuracy of detecting a false accident report. Fortunately, based on experiments, we find that the effect is big only if the speed of vehicles is very low, which depicts a heavy traffic jam.

3) Efficiency of RD^4 in the Freeway Scenario: Next, we show the efficiency of RD^4. Basically, we evaluate how fast a true accident report can be confirmed and how far it can be propagated.

Fig. 4 shows the percentage of nearby vehicles (within 2 km) that confirm the true accident report. The x-axis depicts the time, and the y-axis denotes the percentage of confirmation. Four lines of different colors show different scenarios of different traffic flows. It is easy to see that, after 2 min, nearly 80.00% of the vehicles confirm the accident report within 2 km. As the traffic flow increases from 500 to 800 vehicles/h, more vehicles detect the accident in the first half minute. With the passage of time, the confirmation percentage almost linearly increases approximately 10.00% every half minute because RD^4 needs to collect sufficient signal strength to confirm the accident report.

The propagation of a true accident confirmation is depicted in Fig. 5, where the x-axis records the distance from the vehicle to the accident location, and the y-axis shows the percentage of vehicles of the corresponding distance that confirm the accident report (in 5 min). Similar to the aforementioned experiment, four scenarios with different traffic flows are reported in the figure. With the increase of flow, for the same range, a larger percentage of vehicles detect the accident because a high flow will usually result in a low velocity and more confirmation messages about the accident reports. We can also see that a larger percentage of vehicles located close to the accident confirm the true accident report than vehicles located far away from the accident location. For example, when the flow is more than 1000 vehicles/h, more than 97.00% of the vehicles within the range of 1 km confirm the accident report, and the confirmation rate reduces to between 85.00% and 90.00% when the vehicles are within the range of 2 km. This shows the degrading of the

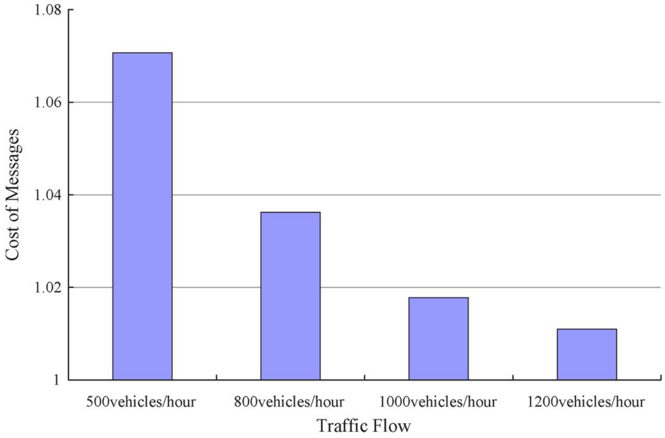


Fig. 6. Message complexity of RD^4.

signal strength of the accident with the increase of the distance between the vehicles and the accident.

4) Message Complexity in the Freeway Scenario: Having seen the effectiveness and efficiency of RD^4, we evaluate the overhead of this mechanism.

Fig. 6 shows the number of messages delivered to confirm a true accident report in terms of different traffic flows. We calculate the message complexity as the ratio of the number of messages that all vehicles received to the number of messages that all the vehicles confirmed as a true accident report. We observe that the average number of messages to confirm a vehicle is very small. On the average, one message is enough. Actually, the cost only accumulates at the beginning of an accident; with more vehicles broadcasting alarm reports, this cost barely increases. Furthermore, message complexity decreases as the flow grows, because more vehicles mean a stronger signal from the other vehicles. Furthermore, vehicles that send messages several times after confirming it contribute to a low message complexity.

B. Road-Construction Scenario

We also implement RD^4 in another scenario to evaluate its performance. The setting for this scenario is basically described as follows. On a state highway or a freeway, roads are under construction, while the other lanes are still open. A typical situation is shown in Fig. 7.

We assume that on a two-lane highway that is partially under construction where one of the lanes is closed while the other one is still open, due to sensor malfunction or malicious attacks, some vehicles try to distribute messages indicating that the entire road is closed. As a result, they try to redirect the vehicle flow to some other highways or local passes. However, the partially closed highway can still accommodate relatively small traffic flow, which makes the redirect unnecessary from a resource-saving point of view. We are trying to deploy the proposed mechanism to avoid such a situation.

We assume that there are 10.00% malfunctioning sensors among all vehicles that will broadcast false messages to their neighborhood every 1 s. The parameter settings are exactly the same as in the previous simulation, which, in this case, indicate that CSR_{self}, CSR_{rsu}, CSR_{pub}, and CSR_{reg} are set to 5, 3, 2, and 1, respectively, except that we increased the threshold since traffic may slow down because it is heavy. We have tested more than seven different traffic flows, and the results are demonstrated in Fig. 8.

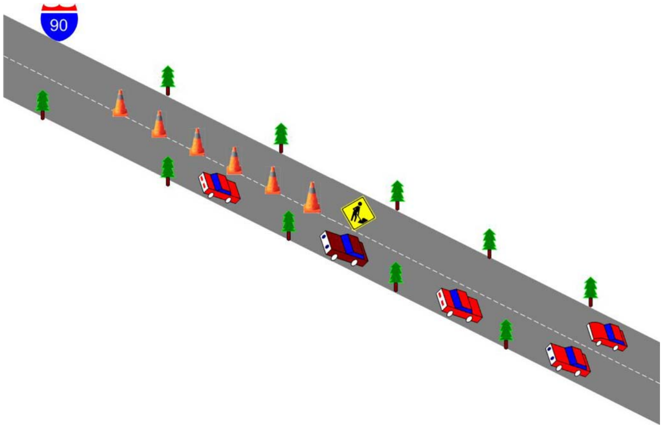


Fig. 7. Road under construction. One lane is closed, and the other is still open.

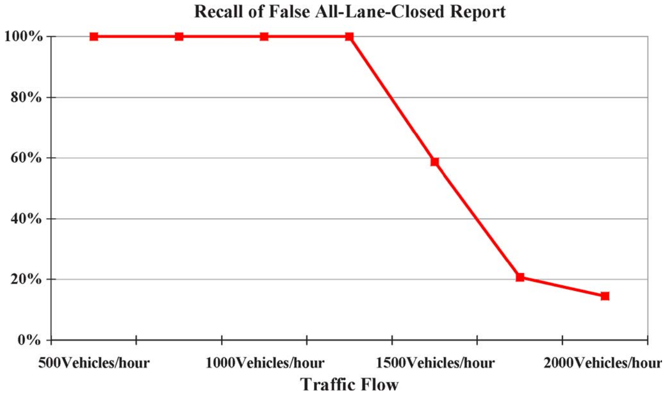


Fig. 8. Recall of false reports in the one-lane-closed scenario.

First, the recalls reach almost 100.00% if traffic flows are 500, 800, 1000, and 1200 vehicles/h, respectively. However, the recall dramatically decreases when the traffic flow is beyond 1200 vehicles/h. The reason is that the traffic flow is so heavy that one lane can hardly let all vehicles pass through; as a result, a traffic jam is formed during the simulation. It is obvious that the recall drops along with increases in the number of vehicles since a jam is more likely to form with a heavy traffic flow, and messages are easier to broadcast with a high vehicle density. To further understand the performance of the proposed mechanism, we pick the test with a 1500-vehicle/h traffic flow for a detailed analysis.

Fig. 9 shows that before the traffic jam is formed, our mechanism achieves a very high recall ratio, despite the drop after the traffic jam has stopped most vehicles. As the figure shows, before 450 s when a traffic jam is about to happen, the recall of false reports arrives at almost 100.00%. After around 1 min, it gradually decreases to 98.69% and further down to 58.70% about 100 s later.

Elapsed Time	Recall of false all-lane-closed report
Before 450 seconds	99.98%
Around 450 second Traffic jam formed	--
After 515 seconds	98.69%
After 525 seconds	86.30%
After 535 seconds	75.02%
After 545 seconds	65.63%
After 555 seconds	58.70%

Fig. 9. Recall of false reports in the one-lane-closed scenario with a specific traffic flow of 1500 vehicles/h.

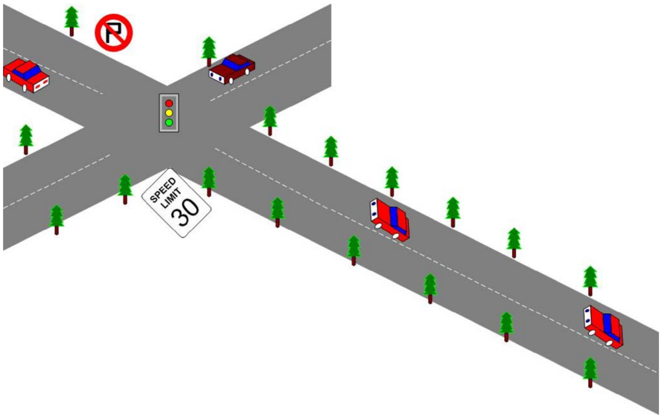


Fig. 10. Scenario of local traffic with traffic light.

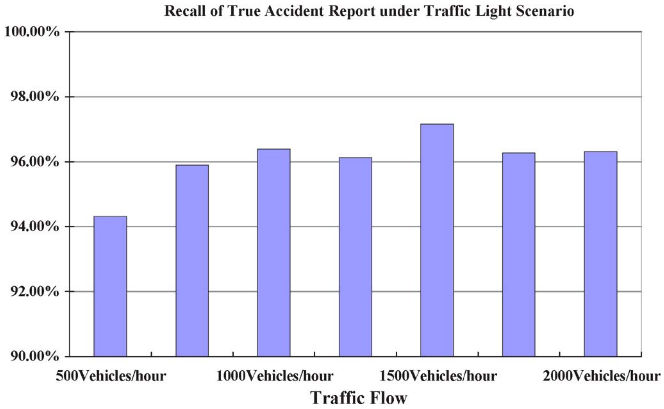


Fig. 11. Recall of true accident reports in the traffic light scenario.

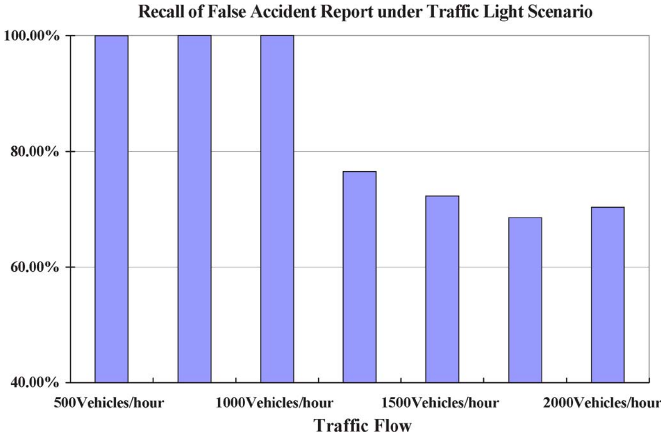


Fig. 12. Recall of false accident reports in the traffic light scenario.

C. Local Traffic Light Scenario

Furthermore, we have tested our mechanism under another circumstance that evaluates the performance of our system locally with a traffic light, as shown in Fig. 10. In reality, accidents while driving locally are less frequent than those that occur on highways in terms of numbers. As a result, it is worth studying. On the other side of the spectrum, all automobiles have to stop for a while if there is an accident ahead, even though the traffic light goes green, which makes the true accident report distribution necessary so that other vehicles can predict the accident and take alternatives, avoiding unnecessary delays.

Obviously, malicious vehicles are willing to distribute false accident reports if the traffic light goes red but without any real accident occurring ahead since both events share a similar observation, i.e., stopped and lower speed vehicles. In such case, we add another new role, i.e., *the traffic light*. Because traffic lights are always managed by authoritative departments, they can contribute as a more reliable component in the system. In addition, traffic lights are always within the sight of drivers who are close enough; it is reasonable to assign a higher CSR to them. In our simulation, the sensor attached to the traffic light will broadcast a negative CSR to nearby vehicles while the traffic light appears to be red. This way, other vehicles are aware of the current road situation. On the other hand, if a real accident happens, regardless of whether the light is red or green, most vehicles will likely remain stopped at the current position. We can imagine that the true message would quickly be distributed because even if the traffic light changes to green, none of the vehicles move.

The red light lasts 30 s in our simulation. The CSR assigned to the traffic light is -3 in this situation. The parameter settings are the same as those in the first simulation. We increase the threshold to 6.02, which, although it may not be the best choice, provides promising results.

The recall of true accident reports is demonstrated in Fig. 11. As we can see, if the accident event report is true, there is a very high probability of confirming it among all the different traffic flows from 500 to 2000 vehicles/h, which is a very high probability of approximately 95.00%. There is a trend to slightly increase as the traffic flow becomes heavy since the proposed mechanism relies on the signal strength received from other vehicles.

Fig. 12 shows the recall of false accident reports after introducing the traffic light. Basically, there are two different categories, as illustrated in this figure. On one hand, because we add another role, i.e., traffic light, the recall reaches almost 100.00% if the traffic flow is up to 1000 vehicles/h. On the other hand, the recall drops as the traffic flow increases. The reason for this is given as follows. If the number of vehicles is large, a traffic jam occurs even after the light turns green. In this case, a considerable number of vehicles jammed could help a false message to be confirmed, although it did not truly happen.

In summary, we find that the RD^4 mechanism effectively and efficiently works in detecting and filtering false accident reports due to malicious attacks and sensor malfunctions, as well as confirming true accident reports at a low cost. A tradeoff should be made in assigning the maximum confidence score for each role, and the way to assign these CSRs could be an interesting direction in future work.

V. Related Work

A previous effort that is close to our idea is presented to detect and diagnose data inconsistency failure in wireless sensor networks [13]. The basic idea of their approach is to build a node-disjoint path and use majority voting to detect inconsistencies among collected data; however, their approach could hardly be applied when the node is compromised. In addition, building several disjoint paths is not applicable in vehicular networks because of dynamically changed neighbors.

In traditional systems, outlier detection problems have been explored. Three fundamental approaches have been proposed to detect outliers [14]. Unsupervised clustering techniques are used to determine the outliers without prior knowledge of the data. They assume that errors or faults are separated from normal data and will thus appear as outliers. The basic idea is to classify data into different clusters and detect the data outside of the cluster as outliers. The other approach is supervised classification, in which they model both normality and abnormality. If new data are classified to the abnormal area, it will be an outlier. Another approach is a mixture of the previous two. Three of them could not fulfill the request that a decision should be made in a real-time fashion and locally since all of them are classified in an offline fashion.

A spatial outlier-detection approach is proposed in [15]. They first formally model the spatial outlier-detection problem, and then, they use a set of neighborhood-aggregation functions and distributive-aggregation functions to detect the outliers. Adam *et al.* design an algorithm to detect anomalies based on neighborhood information [16]. They explore both spatial and semantic relationships among the objects. The Bayesian network is a commonly used approach to classify sensor nodes according to the spatial-temporal correlations between those sensor nodes. For example, Bayesian belief networks are used in outlier detection in [17].

A set of sensing data-cleaning approaches has been proposed. Elnahrawy and Nath propose an approach for modeling and online learning of spatiotemporal correlations in sensing systems to detect and filter noise data [18]. A weighted-moving-average-based approach is proposed to clean sensor data [19]. The basic idea is to average the temporal-spatial data in an efficient way and detect noisy data based on the calculated average. Ye *et al.* propose a mechanism to statistically en-route filter injected false data [20]. They assume that an event will be detected by multiple sensors and rely on the collective decisions of multiple sensors for false report detection. However, their approach is based on medium-access-control-layer authentication without consideration of unreliable communication and sensing components. Declarative support for sensor data cleaning [21] is proposed by Jeffery *et al.* In their work, the authors utilize the temporal and spatial nature of sensor data to drive many of its cleaning process. As a result, they propose a framework named extensible sensor stream processing (ESP) to segment the cleaning process into five programmable stages to clean sensing data. Again, their approach fails to detect deceptive data in a real-time fashion.

Several error-correction algorithms have been presented in the literature. A distributed algorithm to detect measurement errors and infer missing readings in environmental applications is presented in [22]. In their work, a data-distribution model is built based on the history data. An error is detected when there is a mismatch between the model prediction and the sensor reading. Sensor self-diagnostics or sensing-reading calibration has been studied in [23], in which preliminary data checking, analysis, and sensor diagnosis are performed on board. They build a set of rules based on the data collected by different types of sensors and use this rule set to analyze sensing data.

There are also deceptive data-detection algorithms in vehicular networks. Probabilistic validation of aggregated data [24] shows a way to probabilistically detect malicious cars that generate false aggregated information. In particular, they focus on validating speed and location information. Golle *et al.* propose a mechanism [25] to detect and correct malicious data in VANETs. They define a model of VANETs, which specifies what events or sets of events are possible. A function maps a set of events to two values: valid and invalid. Only when the set of events is consistent is it classified as valid. Otherwise, malicious data are detected. How to check the consistency among the events is defined by a set of rules. However, this approach fails if the number of malicious vehicles is greater than the number of honest vehicles.

A mechanism using stronger identities [10] to prevent Sybil attacks in VANETs has been proposed by Raya and Hubaux. Basically, the mechanism requires building the identification of the message sender when necessary. We argue that our mechanism could be built on top of such systems to take advantage of existing security solutions and focus on deceptive data coming from unreliable components and compromised nodes.

VI. Conclusion

To improve the quality of collected data in vehicular networks, in this paper, we have first proposed a role-differentiated cooperative deceptive data-detection and filtering mechanism called RD^4. Other than security-based approaches, RD^4 focuses on the data. Based on a comprehensive evaluation, we show that the proposed mechanism is very efficient and effective; it can confirm over 95.70% of true reports very quickly and filter over 99.90% of false accident reports if traffic flows are within a reasonable range. In addition, the RD^4 mechanism could be applied to different scenarios, as we have evaluated it on a road-construction scenario on a highway and a traffic light scenario while driving locally.

References

[1] M. Batalin, M. Rahimi, Y. Yu, D. Liu, A. Kansal, G. S. Sukhatme, W. J. Kaiser, M. Hansen, G. J. Pottie, M. Srivastava, and D. Estrin, "Call and responses: Experiments in sampling the environment," in *Proc. ACM SenSys*, Nov. 2004, pp. 25–38.

[2] N. Shrivastava, R. Mudumbai, U. Madhow, and S. Suri, "Target tracking with binary proximity sensors: Fundamental limits, minimal descriptions, and algorithms," in *Proc. ACM SenSys*, Nov. 2006, pp. 251–264.

[3] K. Sha and W. Shi, “Consistency-driven data quality management in wireless sensor networks,” *J. Parallel Distrib. Comput.*, vol. 68, no. 9, pp. 1207–1221, Sep. 2008.

[4] A. K. Saha and D. B. Johnson, “Modeling mobility for vehicular ad hoc networks,” in *Proc. Int. Conf. Mobile Comput. Netw., Proc. 1st ACM Int. Workshop Veh. Ad Hoc Netw.*, 2004, pp. 91–92.

[5] M. A. Osborne, S. J. Roberts, A. Rogers, S. D. Ramchurn, and N. R. Jennings, “Towards real-time information processing of sensor network data using computationally efficient multi-output Gaussian processes,” in *Proc. 7th Int. Conf. Inf. Process. Sensor Netw.*, 2008, pp. 109–120.

[6] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos, “Online outlier detection in sensor data using non-parametric models,” in *Proc. 32nd Int. Conf. Very Large Data Bases*, 2006, pp. 187–198.

[7] Dedicated Short Range Communications (DSRC) Home. [Online]. Available: http://grouper.ieee.org/groups/scc32/dsrc/

[8] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode, “Trafficview: Traffic data dissemination using car-to-car communication,” *ACM Sigmobile Mobile Comput. Commun. Rev.*, vol. 8, no. 3, pp. 6–19, Jul. 2004.

[9] J. R. Douceur, “The sybil attack,” in *Proc. IPTPS*, Mar. 2002, pp. 251–260.

[10] M. Raya and J.-P. Hubaux, “The security of vehicular ad hoc networks,” in *Proc. ACM Workshop SASN*, 2005, pp. 11–21.

[11] C. Thiemann, M. Treiber, and A. Kesting, “Estimating acceleration and lane-changing dynamics based on ngsim trajectory data,” *Transp. Res. Rec.*, vol. 2088, pp. 90–101, 2008. DOI: 10.3141/2088-10.

[12] Microsimulator of Road Traffic. [Online]. Available: http://www.traffic-simulation.de

[13] K. Ssu, C.-H. Chou, H. C. Jiau, and W.-T. Hu, “Detection and diagnosis of data inconsistency failures in wireless sensor networks,” *Comput. Netw.*, vol. 50, no. 9, pp. 1247–1260, Jun. 2006.

[14] V. Hodge and J. Austin, “A survey of outlier detection methodologies,” *Artif. Intell. Rev.*, vol. 22, no. 2, pp. 85–126, Oct. 2004.

[15] S. Shekhar, C. Lu, and P. Zhang, “A unified approach to spatial outliers detection,” *Geoinformatica*, vol. 7, no. 2, pp. 139–166, Jun. 2003.

[16] N. Adam, V. P. Janeja, and V. Atluri, “Neighborhood based detection of anomalies in high dimensional spatio-temporal sensor datasets,” in *Proc. ACM Symp. Appl. Comput.*, Mar. 2004, pp. 576–583.

[17] D. Janakiram, A. Reddy, and A. Kumar, “Outlier detection in wireless sensor networks using Bayesian belief networks,” in *Proc. 1st Int. Conf. Commun. Syst. Softw. Middleware*, Jan. 2006, pp. 1–6.

[18] E. Elnahrawy and B. Nath, “Cleaning and querying noisy sensors,” in *Proc. 2nd ACM Workshop WSNA*, Sep. 2003, pp. 78–87.

[19] Y. Zhuang, L. Chen, X. S. Wang, and J. Lian, “A weighted moving average-based approach for cleaning sensor data,” in *Proc. 27th Int. Conf. Distrib. Comput. Syst.*, Jun. 2007, p. 38.

[20] F. Ye, H. Luo, S. Lu, and L. Zhang, “Statistical en-route filtering of injected false data in sensor networks,” in *Proc. IEEE INFOCOM*, Mar. 2004, pp. 839–850.

[21] S. Jeffery, G. Alonso, M. J. Franklin, W. Hong, and J. Widom, “Declarative support for sensor data cleaning,” in *Proc. 4th Int. Conf. Pervasive Comput.*, May 2006, pp. 83–100.

[22] S. Mukhopadhyay, D. Panigrahi, and S. Dey, “Model based error correction for wireless sensor networks,” in *Proc. 1st Annu. IEEE Commun. Soc. Conf. Sensor Ad Hoc Commun. Netw.*, Oct. 2004, pp. 575–584.

[23] H. Li, M. C. Price, J. Stott, and I. W. Marshall, “The development of a wireless sensor network sensing node utilising adaptive self-diagnostics,” in *Proc. 2nd Int. Workshop Self-Organizing Syst.*, Aug. 2007, pp. 30–43.

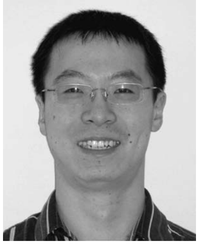
[24] F. Picconi, N. Ravi, M. Gruteser, and L. Iftode, “Probabilistic validation of aggregated data for V2V traffic information systems,” in *Proc. 3rd ACM Int. Workshop VANET*, Sep. 2006, pp. 76–85.

[25] P. Golle, D. Greene, and J. Staddon, “Detecting and correcting malicious data in VANETs,” in *Proc. 1st ACM Int. Workshop Veh. Ad Hoc Netw.*, Oct. 2004, pp. 29–37.

Kewei Sha (M’10) received the B.S. degree from East China University of Science Technology, Shanghai, China, in 2001 and the Ph.D. degree from Wayne State University, Detroit, MI, in 2008.

He is currently an Assistant Professor with Oklahoma City University, Oklahoma City, OK. He is a coauthor of a number of peer-reviewed journal and conference papers. His current research interests include cyber–physical systems, distributed systems, and wireless sensor networks, particularly in data quality management, security and privacy, and system protocol design.

Dr. Sha has served as a program committee member and reviewer for many international journals and conferences.



Shinan Wang received the B.E. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2008. He is currently working toward the Ph.D. degree with the Department of Computer Science, Wayne State University, Detroit, MI.

He is a coauthor of two international conference papers. His current research interests include cyber–physical systems, mobile computing, and wireless sensor networks, particularly in energy-saving and data-mining issues.

Weisong Shi (M’99–SM’09) received the B.S. degree in computer engineering from Xidian University, Xi’an, China, in 1995 and the Ph.D. degree in computer engineering from the Chinese Academy of Sciences, Beijing, China, in 2000.

He is currently an Associate Professor of computer science with the Department of Computer Science, Wayne State University, Detroit, MI. He is the author of more than 100 peer-reviewed journal and conference papers. He is the author of the book *Performance Optimization of Software Distributed Shared Memory Systems* (High Education, 2004). His current research interests include computer systems and mobile computing.

Dr. Shi has served on the technical program committees of numerous conferences. He received a Microsoft Fellowship in 1999, the President’s Outstanding Award from the Chinese Academy of Sciences in 2000, the “Faculty Research Award” from Wayne State University in 2004 and 2005, the “Career Development Chair Award” from Wayne State University in 2009, and the “Best Paper Award” from the International Conference on Web Engineering in 2004 and the International Parallel and Distributed Processing Symposium in 2005. His work was recognized one of 100 outstanding Ph.D. dissertations (China) in 2002. He is a recipient of the National Science Foundation CAREER award.