Modified Self-Organizing Feature Maps for Detection Abnormal Behaviors of Connected Vehicles

Sami Albouq*, Khalid Alghamdi
Department of Computer Science and Engineering
Oakland University
Rochester, MI USA
*Email: ssalbouq [AT] oakland.edu

Mohamed Zohdy
Department of Electrical and Computer Engineering
Oakland University
Rochester Hills, MI USA

Abstract— **Connected vehicles form a self-organized network without a priori fixed infrastructure. However, due to the lack of centralization, they are vulnerable to security attacks, and in order provide security against malicious attacks, Intrusion Detection Systems (IDSs) are being developed for major protection. In this paper, we propose a new scheme for IDSs based on neural networks, which is the Self Organizing Features Maps (SOFM) specifically. We modified this algorithm to improve the performance in detecting anomalies and spot outliers accurately (MSOFM). The privilege of our scheme is that we have no constraints on our data, and thus no need for preprocessing data. The simulations and results demonstrate the capabilities of our scheme in detecting attacks in various scenarios.**

Keywords-Vehicle; Security; Anomalies; Detection; Neural Network;Adaptive; MSOM, Fake Messages, Communications.

I. INTRODUCTION

Connected Vehicles (CVs) comprise a special form of ad-hoc network [1]. It is similar to mobile ad hoc networks (MANET) in creation, but differ in some characteristics such as high mobility and dynamic network typology. The CV nodes branch into two different categories: Roadside Unit (RSU) or Vehicle. Every vehicle is equipped with several devices, such as GPS and Wi-Fi, that enable communication between nodes. This means that every node can join and leave the network dynamically, and the topology of the network can change frequently [2].Communication between nodes works in single or multi-hop, which means that every node serves as both router and data terminal. The CVs communications are divided into two different categories [1]: vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I), also known as vehicle to roadside unit (V2R), as shown in Figure 1. V2V communication is used to propagate alert information while V2R communication is needed to report some events to the centralized authority (CA) such as traffic manager and/or emergency response team.

Security is an important issue in CV applications that provide safety-related information and traffic management [3]. These applications will improve traffic safety with timely provided information about traffic jams, car accidents or road conditions. Also, they require real-time information, and this conveyed information could affect life or death decisions [4]. However, these applications are vulnerable to numerous attacks. A malicious node with access to the open medium can threaten the information security. Every transmitted message carries the status of the sender like its identity and time of sending the message, in addition to safety information. A misbehaving node may tamper with the content of the message or inject a false one into the network to cause serious problems. The main common objective of the malicious node is either out of a selfish desire or maximizing the damage to the network while avoiding being caught. For instance, an attacker may transmit a false message announcing "Heavy traffic conditions" to others in order to make its movement easier on the road as shown in Figure 2. To address this, nodes should minimize the impact of malicious attacks by protecting themselves and distinguishing false information.

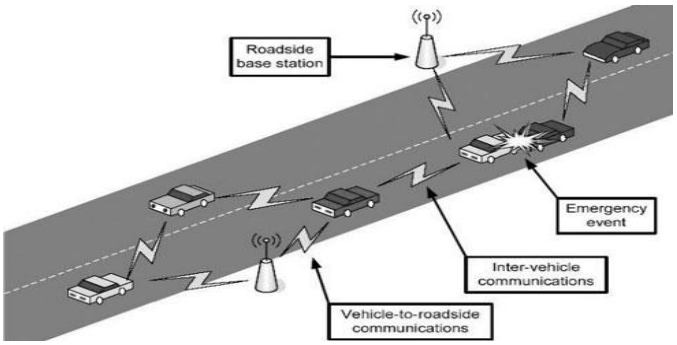

Figure 1. Vehicle communications.

In order to decrease the severity of attacks from malicious nodes and simulate cooperation, regular nodes need to monitor and frequently evaluate received events from their neighbors. Nodes should obtain a certain trust level toward others. They need to cooperate with trusted neighbors, decline any activities

from suspicious neighbors, and report if a neighbor is found to be malicious.

An effective way to identify abnormal behaviors and secure the vital network is an intrusion detection system (IDS) that is coupled with the self-organizing feature maps algorithm (SOFM) [6]. This is a mechanism to identify suspicious activities without affecting the network efficiency. It attempts to recognize anomalous or abnormal activities that deviate from the normal behavior patterns [5]. Our proposed IDS makes an assumption about the data, which motivates the general approach. The assumptions is that the number of normal instances is larger than the number of intrusions because the intrusions are rare and different from the normal data, and they will appear as outliers in the data, which can be detected. However, our goal is to build an intrusion detection system that is able to monitor the network activities and detect such intrusion attacks automatically. Once the attack is detected, the neighboring vehicles should be informed. Therefore, our proposed work is composed of four phases: a phase of feature selection, followed by a modification of the SOM to improve its performance accuracy, then an analysis phase, and finally a response phase to prevent or minimize the impact of the attack on the CVs.

The rest of this paper will be structured as follows. We begin in Section 2 with related work. Here, we discuss existing detection systems, while in section 3, we present our scheme and show the simulation and discussion. Finally, Section 4 concludes this work by successfully detecting tampered information with the source sender.

II. RELATED WORK

One of the first existing solutions in misbehaving detection systems was introduced by Golle et al. [7]. He created a model that verifies and corrects malicious data and detects misbehaving nodes. This researcher used general approach for validating data based on local sensors data collection. Every node verifies received data based on the VANET model. If malicious data is detected, an adversarial model is used to search for best explanations for the errors and correct them.

Raya et al. [8] proposed protocols as components of a framework for a misbehavior detection system that detects fault nodes or misbehavior activities. This framework compares a node's behaviors with the average behaviors of other nodes to build data models on the fly. A technique is called entropy is used to represent the anomalous and normal behaviors of nodes. In addition, K-mean is used for clustering to find attackers and exclude them from the communication system

Ghosh et al. [9] presented a misbehavior detection system for false alert messages in post-crash notification. The proposed model monitors sender's behaviors by checking the validity of received alert messages. If the alert message is correct, the driver can safely take an action based on the situation; otherwise, if the message is incorrect, the driver can ignore that alert.

Kim et al. [10] proposed a message filtering mechanism that leverages multiple complementary sources of information. He built a system with a multi-source detection model. In this system, an observed event is tested with multiple sources to prove its existence. After that, driver is alerted.

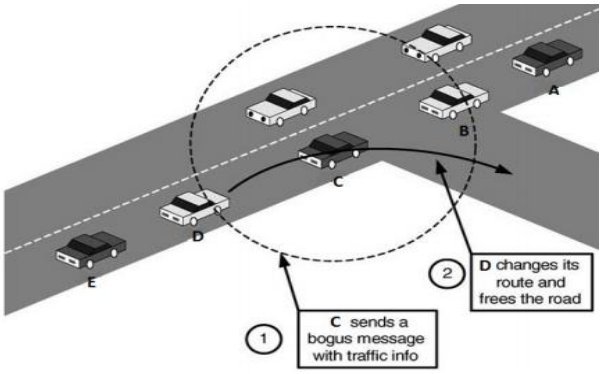

Figure 2. Message tampering.

Yang et al. [11] Introduced MisDis, which is a method that is used for identifying misbehaving vehicles. It is implemented by state automata and supervision in addition to a special security log that records the target vehicle's behavior. The security log idea was exploited from PeerReview system that described in [12]. PeerReview is a system providing accounting for the behavior of every peer and identifying those peers that break the protocol.

Haddadi and Sarram [13] combined two mechanisms anomaly and signature techniques, to introduce an intrusion detection system. The system works in parallel in two-tier analysis function, signature and anomaly detection. If the first stage, which is the signature and anomaly detection cannot classify the data as an attack and normal, it forwards the audit data to a second stage a probable attack detection module for review.

III. PROPOSED SOLUTION

A. *Intrusion Detection System*

The proposed IDS observes attacks as data outliers and deployed clusters. In this work, we used our modified version of the SOFM algorithm to detect all possible attack (anomalies) clusters. There are two methods for discovering attack data in the system:

- **Detecting outlying data** (DOD): every input vector calculates the Quantization Error (QE) with every winning neuron (the best matching unites) to obtain the smallest result, which specifies the vector's cluster

zone or category [14].

$$QE = 1/N \sum_{i=1}^{n} |X_i - m_c| \quad (1)$$

N: denotes number of input vectors.
X_i: denotes the inpute vector .
m_c: denotes the BMUs.

- **Detecting outlying clusters** (DOC): every node calculates its average distance from the rest of the nodes or closest neighbors to determine the similarity.

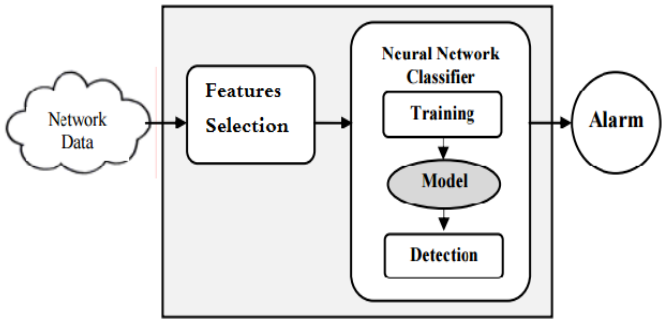

Figure 3. IDS architecture.

The advantage of the previous methods is to eliminate all restrictions on the training data, which means we avoid time consuming preprocessing data. Also, improve detection attack in the situation we do or do not have the traces of attacks during the training phase [14]. However, if clean data is trained using the MSOFM, only the first method will be used. On the other hand, if an attack exists, we may need to use one or both detecting methods for tracing the attack.

1) *Clustering*

In the first step, we trained data to create clusters from received messages. The MSOFM starts with empty clusters and generates clusters based on the similarities and differences of the single read vector data. For every new data vector (message), the MSOFM computes the distance between to every weight vectors. The weight with smallest distance is selected and updated it with its neighbors to form a cluster (more details in section C). At this point, we had either one normal cluster or multiple clusters that contain some suspicious data.

2) *Detection*

In the second step, the objective was to identify suspicious data with their origins. We examined all vectors data in order to find the inconsistencies, having in mind that attacks will often results in deviation from normal ones. Thus, when we compute DOD and DOC distances, they should be similar for all messages in order to be normal. Thereafter, we calculated either mean, median or mode values of all (DOD and DOC) to find a deviation messages. The reason behind this is that the majority of the neighbor nodes send correct messages, which produce higher deviation from each of the above values in the cases of the wrong messages.

B. Feature-Selection

Feature selection is an important step in our proposed IDS. The objective of feature selection is to choose a subset of features from the complete input features to predict the output accurately and reduces computational cost. In CV messages, there are numerous features that can be monitored by the IDS. However, we selected the most important ones to improve the classification and speed up the computation. The reason for eliminating some features is to improve the overall performance of the IDS without affecting the accuracy of detection. The feature that we selected included*:*

- **Message weight**: Defines category and importance of the event.
- **Original confidence**: Defines occurrence accuracy level of the first original message.
- **Calculated Confidence**: Indicates newly calculated confidence of every node.
- **Location**: The coordinates x and y of the original event.
- **Time**: The occurrence time of the original event.

C. Self Organizng Feature Maps

SOFM is a special type of neural network. It is also known as the Kohonen Neural Network algorithm [6], which used to classify and visualize data. It takes higher resolution input data and maps them to multidimensional or single output [20]. The algorithm is considered as an unsupervised training technique that uses competitive learning. Figure 4 illustrates mapping of high dimensional input data space onto a regular two-dimensional array of neurons. The objective of the SOFM is to transform an input data set of arbitrary dimension to a one- or two-dimensional topological map [23].

Before the start of the clustering phase, the algorithm takes in ten parameters including the rate decay, neighborhood radius, history limit, neighborhood decay, learning rate and history factor. The default values are always good for certain situations, but in our case, we modified some parameters to obtain more effective clustering.

D. Learning algorithm

The map is initialized with one of these distributions (Gaussian or Gamma), which generates weights value for each neuron. The goal is to produce small weights that reduces computational time. After that, the algorithm chooses a random vector from the input data set per iteration to present it

to each neuron. The goal is to find which weight most closely matches the input by calculating the p-norms distance [22]:

$$L_p = \left(\sum_{i=1}^{n} |q_i - p_i|^2 \right)^p \quad p - norm\ 1,2 \dots \infty \qquad (2)$$

In the above equation, the variable q represents the input vector, while the variable p represents the weight neuron. The aim of this step is to measure how different each neuron weight is from the input vector. The algorithm tracks the calculation of each distance and selects the shortest one which called the Best Matching Units (BMU). The BMU is the neuron that will receive more learning than neighbors. Since we have the BMU, the algorithm will loop over all of the neuron weights in the array and update every one based on the equation (3):

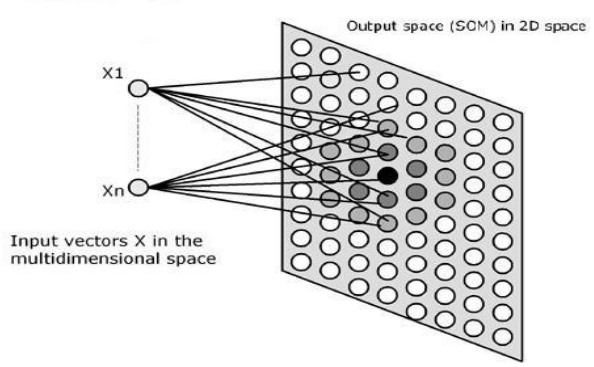

Figure 4. Self-Organizing Feature Maps layers

$$W_v(t+1) = W_v(t) + \alpha(t) * \beta(t) * [dist(t) - W_v(t)] \qquad (3)$$

- **t** denotes current iteration
- **λ** is the limit on time iteration
- $\mathbf{W_v}$ is the current weight vector
- **dist** is the target input
- **β(*t*)** is the neighborhood function, and
- **α(t)**, is learning restraint due to time.

When vector one is finished learning, the second vector is presented and start from equation #2.

E. Simulation and Experiments Discussion

To trust the result of our proposed IDS performance, we created a graphical simulation for CVs that simulates a realistic vehicular scenarios using real road maps from all over the world as figure 5 shown. The simulation integrates both features, network communication and real vehicle movement to provide a close look to the real world, which gives approximate realistic results. It supports traffic models that simulate driving decisions of individual vehicle and road networks. However, to create a real map in our simulation, we took advantage of other technologies such as OpenStreetMap [15, 16], which generates an XML map file. The map file contains high quality information about each city's topology and traffic services that helped us in visualizing the map in our simulator. When the map is created, a traffic scenario was implemented to simulate the real traffic activities as shown in Figure 5.

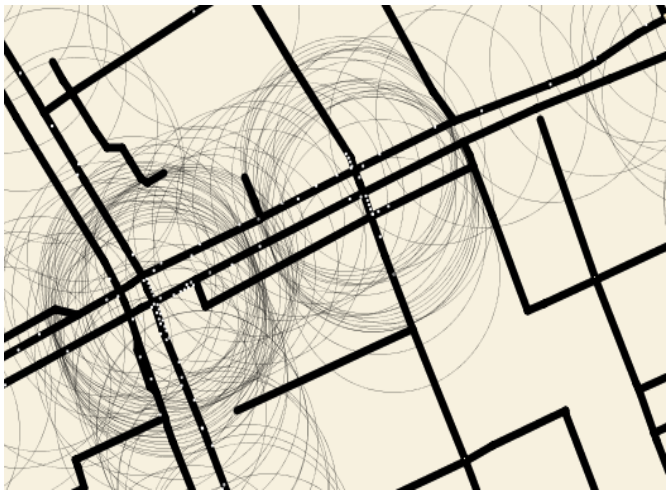

Figure 5. Simulation of vehicle traffic in NY City.

There are two important components used in the simulation: vehicles and RSUs [21]. RSUs is comprised a fixed infrastructure that can be placed in the map on any road to enhance the robustness of the communication between nodes. For example, if there is a gap between nodes, RSU may close this gap by connecting them together. On the other hand, each vehicle is a self-dependent and movable node that determines its decision of the navigation and communication individually.

Communication between nodes is an essential method that is supported in the simulation. Nodes have the ability to communicate with others within a short range distance. In other words, vehicles can disseminate messages to neighbors within the range of 300m. This means, every node maintains a list of the neighbors which enter its communication range. Thus, the techniques that are used for transmitting messages included [17] Beacon, which used heavily in safety applications [18]. Every transmitted message contains several important fields of information such as ID, speed, location, confidence, and others. These fields are used differently based on the type of events.

To examine our IDS, we instantiate a traffic model scenario where vehicles moves around and exchanges messages when an event occurred. Every vehicle is supposed to use our proposed IDS. However, several other events could take place in the scenario such as stopped or slow vehicle advisor, road hazard condition notification, emergency brake and emergency vehicle Approaching [19]. If an event occurs, a

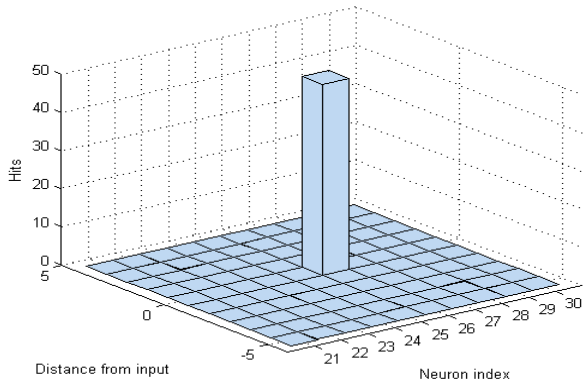

Figure 6. IDS result with clean data

message will be sent to all neighboring vehicles warning them to respond quickly. Neighbors will forward the received message to their neighbors with additional information, for example a new calculated confidence, which determines the event occurrence level beside the initial data that is sent from the original vehicle. Because we assume the first-had message is trusted, the attack may initiate here by sending a number of false messages to some of the neighbors at random fashion. Therefore, when the message is received from neighbors, vehicles should run the IDS to examine the validity of the messages and respond accordingly.

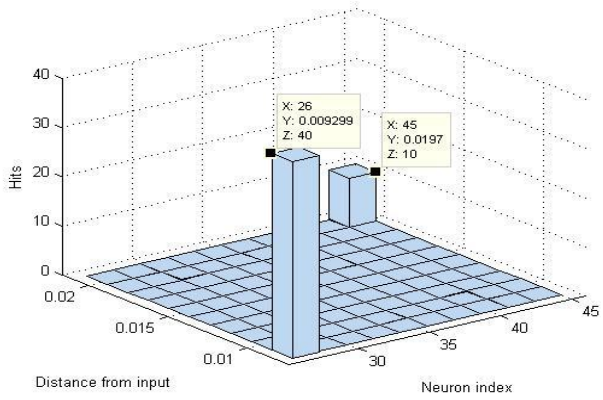

Figure 7. IDS result with 10 new received fake messages after training

There are two situations for attack in our experiment. The attack may appear after or before the training phase. In other words, the first case clean data is trained and later a fake message is received, but the second case both malicious and clean data are trained together. The aim of this is to show that no constrains on the training data, and the IDS is able to detect 100% malicious data.

Figure 6 shows the IDS result with normal messages that were received by a vehicle. The vehicle received 40 valid messages. These messages were presented to the IDS and no outliers were detected because all messages were mapped to the same cluster (BMU #26). After that, 10 new fake messages were received. So, instead of retrain the entire data, we used the DOD and DOC for checking the validity of these messages. The results from both methods revealed that they were anomalies and did not belong to the normal cluster as depicted in Figure 7.

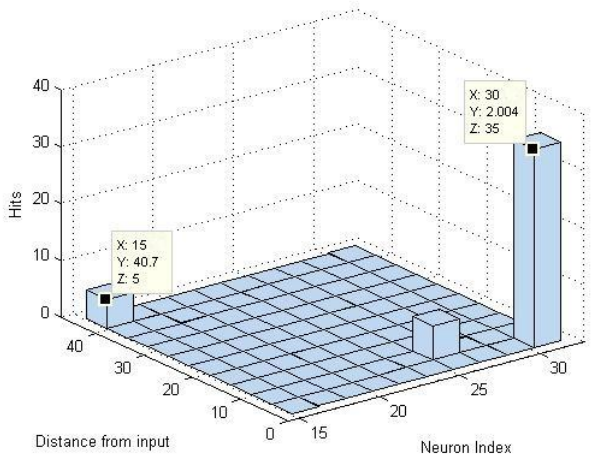

Figure 8. IDS result with both normal and outliers detection

In Figure 8, the IDS examined 50 messages (90% normal and 10% fake). The resulting outcome showed that the most hit of the normal data was received by the BMU #30, while the fake message was clustered into two clusters with 5 total hits for each.

Overall, after further investigations with more fake messages, we concluded that the proposed IDS was able to detect tampered messages when they were less than 49% of the entire data.

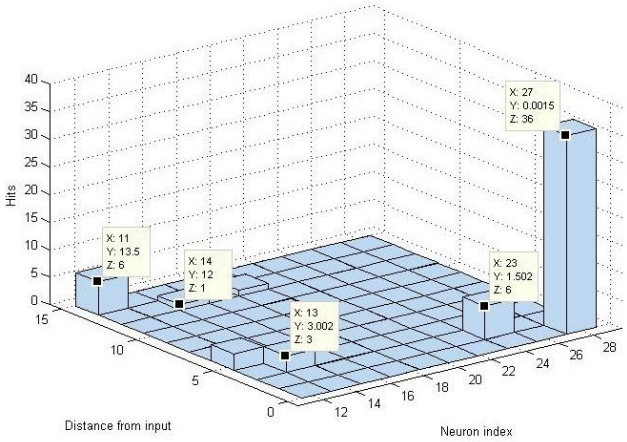

Figure 9. IDS result with 35% outliers.

IV. CONCLUSION

In this paper, we presented a new framework for intrusion detection scheme using our modified version of the Self Organizing Feature Maps algorithm. The approach incorporated two methods (DOC and DOD) that reduced computational cost. They allowed data presented directly to the MSOFM with no preprocessing step and determined attacks after training phase. The MSOFM structure was discussed beside the classification technique. At the end, several experiments and simulations were shown how the IDS were able to classify simulated attack graphically as opposed to normal data.

REFERENCES

[1] Y.Qian, K.Lu and N.Moayeri, 2008, ―A secure vanet Mac protocol for DSRC applications", Gobal Telecommunications Conference, IEEE, pp.1-5

[2] Pathan, Al-Sakib Khan. Security of Self-Organizing Networks : MANET, WSN, WMN, VANET. Chapter 9 Classification of Attacks on Wireless Mobile Ad Hoc Networks and Vehicular Ad Hoc Networks A Survey. Hoboken: CRC Press, 2010. Ebook Library. Web. 13 Aug. 2015.

[3] Design and Analysis - of Realistic Mobility Models forWireless Mesh Networks Master Thesis Philipp Sommer

[4] Pathan, Al-Sakib Khan. Security of Self-Organizing Networks : MANET, WSN, WMN, VANET. Chapter 10 Security in Vehicular Ad Hoc Networks Vikas Singh Yadav, Sudip Misra, and Mozaffar Afaque

[5] . Ebook Library. Web. 13 Aug. 2015.

[6] VOKOROKOS,, Liberios, and Anton BALÁŽ,. "INTRUSION DETECTION SYSTEM USING SELF ORGANIZING MAP." Acta Electrotechnica Et Informatica, 1 June 2006. Web.http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.399.1118&rep=rep1&type=pdf

[7] Kohonen, T. 1995. Self-Organizing Maps, volume 30 of Springer Series in Information Sciences. Berlin, Heidelberg: Springer. (Second Extended Edition 1997).

[8] Golle, Philippe, Greene, Dan, and Staddon, Jessica. (2004). Detecting and correcting malicious data in VANETs. Paper presented at the Proceedings of the 1st ACM in- ternational workshop on Vehicular ad hoc networks, Philadelphia, PA, USA. 29-37.

[9] Raya, M., Papadimitratos, P., Aad, I., Jungels, D., and Hubaux, J. P. (2007). Evic- tion of Misbehaving and Faulty Nodes in Vehicular Networks. IEEE Journal on Se- lected Areas in Communications, 25(8), 1557-1568.

[10] Ghosh, Mainak, Varghese, Anitha, Gupta, Arobinda, Kherani, Arzad A., and Muthaiah, Skanda. (2009). Misbehavior detection scheme with integrated root cause detection in VANET. Paper presented at the Proceedings of the sixth ACM interna- tional workshop on VehiculAr InterNETworking, Beijing, 123-124.

[11] Kim, T.H., Studer, H., Dubey, R., Zhang, X., Perrig, A., Bai, F., Bellur, B., Iyer, A.: VANET Alert Endorsement Using Multi-Source Filters. In: Proceedings of the Seventh ACM Inter- nationalWorkshop on VehiculAr InterNETworking, pp. 51–60. ACM, New York (2010)

[12] Yang, Tao, Xin, Wei, Yu, Liangwen, Yang, Yong, Hu, Jianbin, & Chen, Zhong. (2013). MisDis: An Efficent Misbehavior Discovering Method Based on Accounta- bility and State Machine in VANET. In Y. Ishikawa, J. Li, W. Wang, R. Zhang & W. Zhang (Eds.), Web Technologies and Applications, 7808, 583- 594

[13] Haeberlen, A., Kouznetsov, P., & Druschel, P. (2007, October). PeerReview: Practi- cal accountability for distributed systems. In ACM SIGOPS Operating Systems Re- view (Vol. 41, No. 6, pp. 175-188). ACM.

[14] F. Haddadi, M. Sarram, Wireless intrusion detection system using a lightweight agent, in: Second International Conference on Computer and Network Technology, Bangkok, Thailand, 2010, pp. 84–87.

[15] Banković, Z., Moya, J.M., Araujo, A., Fraga, D., Vallejo, J.C., de Goyeneche, J.M.: Distributed Intrusion Detection System for WSNs based on a Reputation System coupled with Kernel Self-Organizing Maps. Int. Comp. Aided Design 17(2), 87–102 (2010)

[16] M. M. Haklay and P. Weber, "OpenStreetMap: User-Generated Street Maps," IEEE Pervasive Computing, vol. 7, no. 4, pp. 12–18, 2008.

[17] "OpenStreetMap," mayo 2014. [Online]. Available: http://wiki.openstreetmap.org/wiki/Main Page

[18] L. Buttyan, T. Holczer, and I. Vajda, "On the effectiveness of changing ´ pseudonyms to provide location privacy in vanets," in Proceedings of the Fourth European Workshop on Security and Privacy in Ad hoc and Sensor Networks (ESAS 2007), Juli 2007.

[19] M. Raya and J.-P. Hubaux, "The security of vehicular ad hoc networks," in Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks (SASN). ACM, November 2005, pp. 11–21.

[20] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," Physical Review E, vol. 62, pp. 1805–1824, August 2000

[21] E. Cavalcante, A. Aquino, G. Pappa, and A. Loureiro, "Roadside unit deployment for information dissemination in a VANET: An evolutionary approach," in Proc. 14th Int. Conf. Genetic Evol. Comput. Conf. Companion, 2012, pp. 27–34. [Online]. Available: http://dx.doi.org/10.1145/ 2330784.2330789

[22] Bryant, Thomas and Dr. M. Zohdy. "Noise Signal Identification by Modified SelfOrganizing Maps." International Journal of Computer and Information Technology, 1 Jan. 2014. Web. 1 Jan. 2014.

[23] Patole, Vivek. "Self Organizing Maps to Build Intrusion Detection System." International Journal of Computer Applications, 1 Jan. 2010. Web.