# A Robust Traffic Information Management System Against Data Poisoning in Vehicular Networks

Carlos Pedroso, Thiago S. Gomides, Daniel L. Guidoni, Michele Nogueira, Aldri Santos

***Abstract*—Attacks against systems supported by vehicular networks, such as Traffic Information Systems, are more frequent and critical because of the real-time demand and high volume of data. Attacks that decrease data reliability, as data poisoning – DaP, are the most damaging because they severely risk data use. However, in general, vehicular network systems do not implement these features. Hence, this work presents MOVE, an efficient, secure, and VANET-based traffic management system against DaP attacks. MOVE relies on watchdog monitoring and relational consensus for attack detection, achieving efficient data authenticity and high availability. The performance evaluation of MOVE on OMNET++ has reached a detection rate of 90%, false-negative and false-positive rates of 4% and 10%, respectively. MOVE decreases vehicle travel time by up to 40%, and average time on traffic jams by 35%. It increases the average speed by 12% compared to ON-DEMAND.**

***Index Terms*—Robust traffic management system, VANETs security; Attack detection and prevention.**

## I. Introduction

The effective management of urban mobility is key to smart cities because it directly impacts the quality of life [1]. Inefficient urban mobility highlights social issues, such as difficulties in managing transportation time and stress, producing traffic jams and accidents. Inefficient urban mobility also contributes to the rise of air pollution and costs, with financial and environmental consequences [2]. Progress in road infrastructures reduces traffic jams in certain regions, but it commonly requires a high cost of design and operation. Therefore, information and communication technologies play a crucial role to improve safety, efficiency, and comfort in urban mobility [3], [4], being vehicular *ad hoc* networks (VANETs) a relevant technology to reduce the impact of congestion and improve traffic management [4], [5].

VANETs comprise vehicles that collaborate to monitor and disseminate traffic conditions assisting in estimating new routes without congestion. The secure and reliable exchange of information is fundamental to getting those benefits [6], [7]. However, VANETs face security vulnerabilities that allow attackers to exploit and violate data dissemination service's availability, integrity, and authenticity. Attacks undermine and impact the operation of traffic management system (TMS) [7]. Several studies have focused on handling attacks against data dissemination services [8], having as main goal to protect vehicles or data [6], [7] against attacks.

Among data-oriented attacks, Data Poisoning (DaP) is one of the most harmful. They generate inconsistency in data during dissemination [9], making it a challenge to detect *DaP* attacks because attackers benefit from devices already authenticated and operational in the network, performing data collection and dissemination [10]. Vehicle-oriented solutions assess data sources through the analysis of their behavior. Their results show that existing vehicle-oriented solutions have not been suitable for traffic management systems because they employ centralized entities, generating high consumption of resources, disregarding data verification, and they ignore collaborative detection between vehicles.

Hence, the literature urges alternatives to deal with threats in data dissemination services exploring the natural characteristics of VANETs. In this direction, collaborative detection stands out [11], where each device carries out its standard functions and plays as a collaborative attack detection agent. An effective way to achieve collaborative detection is to combine *watchdog* strategies and relational consensus strategies [12], which allow systems to work in a distributed way among vehicles and quickly detect, identify, and isolate suspicious vehicles that exhibit DaP behavior over time.

This article presents a robust traffic management system for vehicular networks, called MOVE (*Relational Consensus-Based Secure Traffic **M**anagement f**O**r **V**AN**E**TS*). MOVE reduces the damages of traffic jams, and its operation follows the collection and distribution of traffic information to support the identification of alternative routes that improve travel time. MOVE also supports a robust communication mechanism through *watchdog* monitoring and decision-making by relational consensus. MOVE promotes accurate data authenticity and availability to create a database distributed through vehicles on the road. A performance evaluation follows simulations and promotes a comparative analysis among the ON-DEMAND system [4], other traffic management system, and MOVE under different scenarios. MOVE has achieved 90% of DaP detection rate, with 4% of false negative, 10% of false positive, with 0.86 of accuracy. MOVE has promoted a reduction of average travel time by 40%. Additionally, it has decreased the average time lost caused by traffic jams by 35%, and has increased the average speed by 12% compared to ON-DEMAND.

This article proceeds as follows. Section II presents the related works. Section III details the MOVE system. Section IV

Carlos Pedroso, Wireless and Advanced Networks Laboratory (NR2) - UFPR, Brazil (capjunior@inf.ufpr.br)

Thiago S. Gomides, DCOMP - Federal University of São João Del Rei - UFSJ, Brazil (gomides@ufsj.edu.br)

Daniel L. Guidoni, DECOM - Federal University of Ouro Preto - UFOP - Brazil (guidoni@ufop.edu.br)

Michele Nogueira, Wireless and Advanced Networks Laboratory (NR2) - UFPR, Brazil (michele@inf.ufpr.br)

Aldri Santos, Wireless and Advanced Networks Laboratory (NR2) - UFPR, Brazil (aldri@inf.ufpr.br)

shows the performance evaluation. Finally, Section V presents conclusions and future works.

## II. Related Work

Recently several works have addressed traffic management system supported by distributed communication in vehicular networks [4], [5], [13]. In these solutions, vehicles are responsible for route monitoring and decision-making. Upon detecting a traffic jam, the vehicle forwards this information to others by a collaborative method, so that they can define a new route with fewer congestion points. Though, security issues emerge and disturb the TMSs [7], [14]. In [8], it is proposed an instant data evaluation scheme (IDES) for vehicle reputation management in VANETs against 3 types of intrusion attacks: bogus, collude, and secret. IDES collects the global reputation of vehicles for instantly recognizing untrusted data messages. It considers the global historical reputation records, requiring low processing time for message validation, but it ignores the collaboration between vehicles and disregards data validation. In [15], a vehicular misbehavior detection system based on Support Vector Machine (SVM) taking into account the data trust and the vehicle trust. The data trust emerges the SVM-based classifier to detect false messages based on message content and vehicle attributes, while the vehicle trust comes from a local vehicle trust module and a trusted authority (TA) vehicle trust module. In [16], a reputation management framework evaluates the trust in VANETs to identify denial of traffic service. The framework employs an integrated entity-centric and event-centric mechanism to establish trust. It uses the Roadside Unit (RSU) to manage the long-term reputation scores for most commuter vehicles with predefined daily trajectories. An event-centric reputation mechanism is adopted as a useful supplement. Though, computing trust between vehicles only is not enough to disseminate data safely.

In [17], the authors developed a framework for behavior detection by a V2X communication (vehicle-to-vehicle, vehicle-to-infrastructure and vehicle for any communication device). A central server acts as an authority for operation identification to achieve a global classification of the behavior of vehicles. Hence, this structure constantly overloads the network and hinders transmission between vehicles that always need to inform the central about misbehavior. In [18], an Intrusion Detection System (IDS) for detecting false data injection attack on a fixed industrial IoT network employs clusters similarity to handle with the devices density. It combines *watchdog* monitoring and collaborative consensus among objects to handle false data injection attacks. Despite effectiveness against attack, it ignores the device mobility issues. In [19], a system to detect distributed denial of service attacks (DDoS) attacks analyzes each data collection point using a Bayes classifier. The analysis happens redundantly, in parallel with the level of each data collection point, to avoid the single point of failure. Further, a distributed consensus favors collaborative decision-making. Though, the communication overhead among the participants diminishes the system's effectiveness. Therefore, as described in the works above, approaches against DaP attacks in vehicular networks usually employ a centralized and trusted entity for attack detection, limiting the growth of the network. In addition, they disregard data verification and the source of the attack, which increases network malfunction and allows vehicles to use corrupted information to make transit decisions. Thus, it is necessary to develop solutions capable of acting in a distributed way and capable of identifying and isolating malicious devices in VANETs to guarantee the exchange of traffic information between vehicles.

## III. Secure Traffic Management

This section describes the MOVE (*Relational Consensus-Based Secure Traffic **M**anagement f**O**r **V**AN**ET**S*) system for providing secure VANETs traffic management against data poisoning attacks. By periodically checking their neighboring roads' traffic conditions, vehicles with MOVE get to detect traffic jams and thus computing new routes. For that, vehicles check their neighboring roads' traffic conditions periodically and establish a new route whether traffic jams are detected. A communication protocol enables vehicles to request traffic information from other roads and store it in order to check it posteriorly. Applying the stored data, each vehicle gets to check for a new lower-cost route. Meantime, in this scenario, malicious vehicles/attackers, aiming network disturbing, can add false information through the data poisoning approach and affect the new routes decisions. Thus, MOVE is a TMS that takes care in identifying and excluding vehicles with misbehavior from the traffic information request-response process. Fig. **??** illustrates the vehicular setting of the MOVE operation that by using traffic information from neighboring roads, vehicles compute new routes with fewer traffic jams; in this way, each solid arrow on the road means a possibility for traffic jams avoidance. Though, face a DaP behavior by the red vehicle (red dotted arrow), the vehicle route changes; and the route in purple is no longer considered by the vehicle.

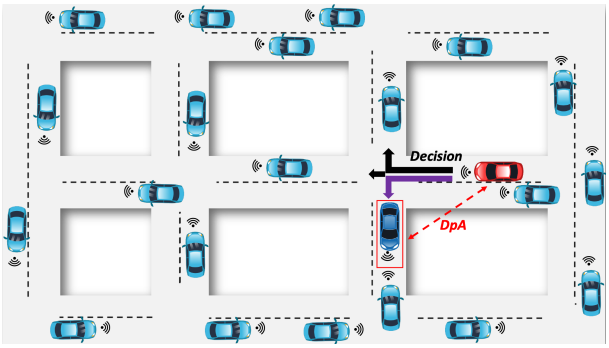



Fig. 1. A scenario of MOVE operation.

**Vehicular devices model:** Each vehicle in the VANET environment is capable of processing, analyzing, communicating and making distributed decisions. Vehicles own a set of technologies as On-Board Data Unit (OBU), IEEE 802.11p communication interface, and several sensors. Further, they operate in two main ways, as normal network members or cluster-heads, where a cluster means a group of vehicles traveling in the same lane with similar traffic views.

**Network model:** The VANET scenario is represented by a directed and weighted graph $G = (V, E)$, where $V$ (vertex)

and $E$ (edges) represent intersections and roads, respectively. The set $V$ is defined by $V = \{v_0, v_1, \ldots, v_n\}$ and the set $E = \forall(v_i, v_j)$, so that each edge $e_{ij} = (v_i, v_j)$ means the road segment that connects two intersections $v_i$ and $v_j$. The respective cost to travel the segment $e_{ij}$ is $w_{ij}$. $W$ describes the set of weight as $W = \{w_{ij},\ i \neq j\}$.

**Communication model:** Vehicles communicate each other over the wireless medium by mean of an asynchronous channel with packet loss due to noise and nodes position. Further, we applied two types of messages to help the MOVE manage and secure the network: $i$) alert/control messages for the control, management, the formation of clusters, and exclusion of attacking nodes; $ii$) Congestion Level (CL) messages, which provide information on traffic flow variations.

**Attack model:** The threat to the network consists of Data Poisoning (DaP) attacks, where attackers, once intruded into the network, initiate false dissemination of congestion information. There are three different variants of attacks: $i$) Inverse attack, whose attackers always send traffic data in reverse to the real data collected by the vehicles; $ii$) Max Level attack, where the attackers always send the highest measurement of traffic data to other vehicles; and $iii$) Random attack, where attackers always send traffic measurements with random values. We assume that DaP attacks happen by exploiting vulnerabilities resulting from other attacks, such as *Sybil*, or even network failures in which the attacker knows the data characteristics [10].

### A. MOVE Architecture

The MOVE architecture comprises four main modules, named **Displacement Analysis (DA)**, **Congestion Level Dissemination (CL-D)**, **Route Decision (RD)**, and **Security (SR)**, as shown in Fig. 2. Through the DA module, vehicles periodically monitor the displacement on their current road. DA plays an essential role in providing traffic information to CL-D and SR modules. CL-D then forwards the traffic information provided by DA to the neighbor's vehicles. Vehicles must store these received traffic data on their database, which contains the CLs of adjacent roads. RD makes use of the stored information for checking the existence of new routes, i.e. RD is a decision-making module. Note that, over the CL-D operation, malicious vehicles may provide poisoning traffic data about the conditions of their roads, and harming the decision of the module RD about the appropriate route.

The **SR** module addresses the security of traffic information disseminated among vehicles so that only legitimate data is available, i.e without poisoning. It comprises three components called Cluster Alert Dissemination **(CAD)**, Cluster Control **(CC)**, and DaP Detection **(DD)**. The CAD component holds the control messages exchanged among the vehicles with information about the cars' displacement analysis. CC coordinates the creation of a cluster of vehicles on a road and makes the monitoring and verification of vehicles that do not respect the similarity threshold between the traffic information disseminated by the vehicles. DD acts in the decision making and isolation of DaP attacks, applying jointly relational consensus and statistical analysis of standard deviation for detecting and excluding DaP vehicles. Thereby, when detecting an attacker, the vehicles alert the cluster leader about the attacker's existence and its traffic information.

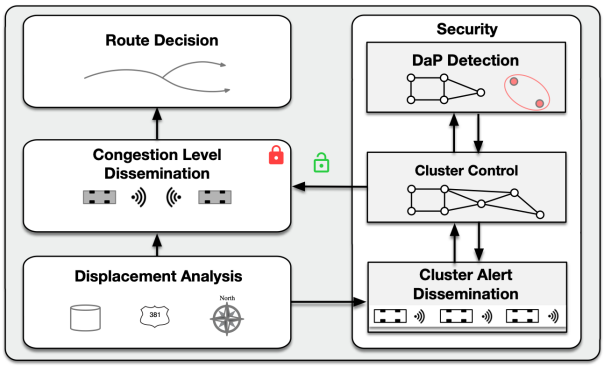



Fig. 2. MOVE architecture.

### B. Traffic Management

The MOVE's vehicular traffic management plays similarly to the model proposed by [4]. Therefore, during the displacement on city roads, vehicles monitor their current road traffic conditions. The **DA** module performs this role by analyzing vehicle travel conditions and correlating them with the road characteristics. Thus, each vehicle monitors its displacement based on the relationship between the Traveled Distance (TD) with the Verified Travel Time (VTT). Regarding the maximum allowed speed and VTT, we can calculate the Expected Distance (ED), which is the expected distance under free-flow conditions. In this sense, when MOVE checks that the vehicle has $TD = ED$, it considers the vehicle as traveling under free-flow conditions, which means a congestion level equals 1; and whether $TD < ED$ the vehicle is traffic jams. Note that the higher difference between TD and ED, the greater is the CL value; and the congestion levels range from 1 (free-flow) to 10 (severe traffic jam). In [4], the authors provide a detailed explanation about the CLs' calculation.

All vehicles maintain a database with a set of tuples $(road_{id}, value)$, where $road_{id}$ means a road identifier and $value$ corresponds to either *(i)* unknown value or *(ii)* congestion level value. The database tuples provide traffic information upgraded in $t$ predefined intervals. For instance, for each interval $t$ seconds, vehicles must check which adjacent roads have an unknown status in their database. For these nearby road segments with $unknown\ value$, a traffic $request$ message is then forward. To avoid network overload issues, only the longest vehicle traveling on a road with traffic request must reply the message, i.e, the cluster leader. Note that this vehicle has a more accurate perception of the CL, disregarding the impact of small traffic variations. Each request message aims to trigger a $response$ message. The $response$ message is addressed for all vehicles in the communication range, even the requesting one. Thus, vehicles update their respective databases with updated information $(road_{id}, CL_{id})$. For an unanswered request, an unknown value is assign in the database. Module **CL-D** is responsible for database making.

After receiving the requested information, through Module **RD**, vehicles apply the network model based on the

graph $G = (V, E)$ (described in Section III) to know the existence of a new lower-cost route. Therefore, each tuple $(road_{id}, CL_{id})$ added in the database is associated with an edge $e_{ij} = (v_i, v_j)$ and an weight $w_{ij}$. While roads with unknown values receive $CL = 1$ for keeping the consistent database, i.e. with all segments information. Vehicles get to update $G$ and calculate a route, and under the new route with fewer traffic jams interference than the previous one they will change their direction.

### C. Security Management

The security module (SR) takes into account the control messages exchanged between vehicles to identify traffic level values out of the predefined similarity threshold established by the traffic information on the road. It keeps data about neighbor vehicles seen as suspects or attackers in disseminating false information. Those data support SR to analyze the vehicles operating on the road, allowing an assertive detection of DaP attackers, and the maintenance of reliable vehicle clusters. The cluster leader vehicle, one that has been on the road the longest, acts to exclude DaP attacks based on the information sent by the other cluster members. Algorithm 1 describes the procedures for identifying and mitigating inconsistent traffic information created by DaP attacks. The detection starts along with the first exchanged control message since the vehicles need neighbor traffic information to compare with its information.

**Algorithm 1:** DaP Attack Security

```
1  procedure CLUSTERING CONTROL (msg)
2    if msg.EmitterVehicle ∈ NeighList then
3      NeighTraff ⟵ NeighTraff ∪ msg.TrafficMeasure
4
     NeighTrafTime ⟵ NeighTrafTime ∪ msg.TravelTime
5    else if msg.EmitVehicle ∉ AttkList, SuspList  then
6      if msg.TrafficMeasure ≤ Thresholdconsensus then
7        NeighList ⟵ NeighList ∪ msg.EmitVehicle
8        NeighTraff ⟵ NeighTraff ∪ msg.TraffMeasure
9      else
10       SuspList ⟵ SuspList ∪ msg.EmitVehicle
11 end procedure

12 procedure DAP ATTACK DETECTION (msg)
13   if msg.EmitVehicle ∈ SuspectList then
14     SuspList[msg.EmitVehicle] + +
15   if SuspList[msg.EmitVehicle] > thresholdattack then
16     AttkList ⟵ AttkList ∪ msg.EmitVehicle
17     BroadcastMessage("DetectedAttacker", Attk =
     msg.EmitterVehicle)
18 end procedure

19 procedure CONSENSUS EXCLUSION (msg)
20   if AmILeader then
21     if msg == "DetectedAttaker" then
22       AttkListLeader[msg.Attk] + +
23       if AttkListLeader[msg.Attk] > thresholdattack then
24
     BroadcastMessage("Attackerbyconsensus", msg.Attk)
25 end procedure
```

The CAD component periodically sends control messages by beacons for establishing clustering of vehicles as well as vehicles labeled by others as suspect, attacker, and honest. Thus, upon receiving a control message $(msg)$ via the **Clustering Control** procedure, the vehicle verifies the traffic information of the vehicle emitter to check whether it $(EmitVehicle)$ is belonging to its street neighborhood $(NeighList)$ (*l*.1-2). Case the emitter is already a cluster neighbor, i.e. it has previously sent information within the *Thresholdconsensus*, the vehicle then stores its traffic information $(NeighTraff)$ and travel time $(NeighTravTime)$ (*l*.3-4). Reminding that the longest travel time will indicate the cluster leader vehicle. Case the emitter is not yet a cluster neighbor, it is checked in the lists of $(SuspList)$ and $(AttkList)$ (*l*.5). Suppose the traffic congestion sent respect the *Thresholdconsensus*, i.e. the average level of congestion according to the traffic information sent by the vehicles in the cluster, the receptor updates its list of neighbors and the traffic information (*l*.7-8). Otherwise, only the $SuspectList$ list is updated.

The **DaP Attack Detection** procedure carries out also whenever a vehicle receives a control message $(msg)$ in order to check which vehicles are acting as attackers on the network. When the emitter vehicle belongs to the suspect list of the receptor vehicle (*l*.13), and whose the count exceeds the threshold $thresholdattack$ (*l*.15), the emitter is classified as an attacker, being then a "$DetectedAttacker$" message sent to the cluster vehicles (*l*.16-17). The **Consensus Exclusion** procedure performed by the cluster leader makes the removal of attacking vehicles in that clustering. As the role of cluster leader varies over time, the vehicle checks whether it is playing as one at that moment (*l*.20). The leader, upon receiving a "$DetectedAttacker$" message, updates how many times the attack was detected by cluster members (*l*.22) and when exceeding the $thresholdattack$, it must also notify all neighbors on the street the discovery of an attacker by consensus (*l*.24). Note that the list of $NeighTravTime$ initializes at each beacon interval. Thus, at the moment a vehicle changes the traveled street, it will send a control message on the new street and will leave the clustering of the previous one since each vehicle belongs to only one street clustering at any given time.

$$DP = \sqrt{\frac{\sum_{i=1}^{n} (X_i - M_A)^2}{N}} \leq Thresholdconsensus \quad (1)$$

Equation 1 computes the relational consensus based on the traffic congestion values measured by the vehicles. Thus, the reading values collected are used to form and assess the consensus between the vehicles in the cluster.The consensus formation takes into account the set of traffic congestion values $D = (d_i, d_{i+1}, ..., d_n)$, which represents the samples checked by the leader. The consensus calculation, indicated by $\sum_{i=1}^{n}$, comprises the sum of the values of all positions. The value of $X_i$ is referenced in position $i$ of the data set $D$. $M_A$ represents the arithmetic mean of the data. $N$ means the amount of data evaluated in the formation of the consensus. The consensual threshold $(Thresholdconsensus)$ expresses the predefined value, i.e. the traffic maximum allowed in the clustering, which can change according to the type of information evaluated. In this sense, the relational consensus encompasses the agreement, relationship and uniformity of opinions that vehicles establish through exchanging information between them. This information relates to the traffic congestion value available on each vehicle, being associated with the other vehicles to validate DaP attacker.

Equation 1 computes the relational consensus **considering/through/based on** the values measured by the vehicles. **In this way/Particularly**, the reading data collected by the vehicles participating in the network **are used to form a consensus and compare them between them. The consensus formation/deal takes into account...** Thus, a data set $D = (d_i, d_{i+1}, ..., d_n)$ is used, which represents the samples checked. The consensus calculation, indicated by $\sum_{i=1}^{n}$, comprises the sum of the values of set $D$, from the first position $(i = 1)$ to position $n \in N$. The value of $X_i$ is referenced in position $i$ of the data set $D$. $M_A$ represents the arithmetic mean of the data. $N$ means the amount of data evaluated in the formation of the consensus. The denominated consensual threshold ($Thresholdconsensus$) expresses the predefined value, which can change according to the type of data evaluated and the application where it operates. In this sense, the relational consensus encompasses the agreement, relationship and uniformity of opinions that vehicles establish through exchanging information between them. This information relates to the data available on each vehicle and is associated with the other vehicles to validate DaP attacker.

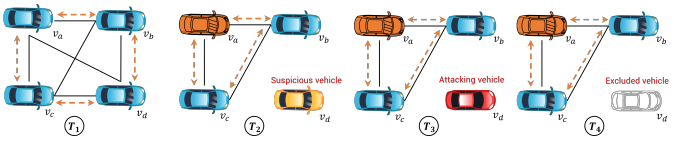



Fig. 3. Relational consensus among vehicles.

Fig. 3 illustrates an example of relational consensus among vehicles for detecting a DaP attacker. The dotted arrows mean the communication between the vehicles. In $T_1$, the vehicles, $v_a$, $v_b$, $v_c$, $v_d$ perform the exchange of control message and cluster formation and leader elections. In $T_2$, only the vehicles ($v_a$, $v_b$, $v_c$) integrate the cluster, because they respect the similarity threshold of the traffic information and $v_a$ is elected leader. However, the traffic information sent by $v_d$ does not respect the $Thresholdconsensus$, and $v_d$ is classified as a suspicious. In $T_3$, the vehicle $v_d$ sends control messages again to try to return to the cluster. The set formed by the vehicles ($v_a$, $v_b$, $v_c$) perform the calculation of Equation 1 and classify $v_d$ as an attacker since it again has divergent readings about the readings of the vehicles in the set. Finally, $T4$ illustrates the situation where the messages from the vehicle $v_d$ are disregarded since it will not participate in the clustering. Thus, the security is maintained in a distributed way by the participants themselves without the need for external entities. Vehicles executes **DA** during the entire trip. **DA** module, therefore, provides information to **CL-D** and **SR** modules. **CL-D** assess the congestion level requests by vehicles in adjacent roads. Meanwhile, *SR* allows vehicles to cluster to detect and exclude DaP attackers. After that, **CL-D** modules receives filtered data and forwards it to *RD* in which performs the decision.

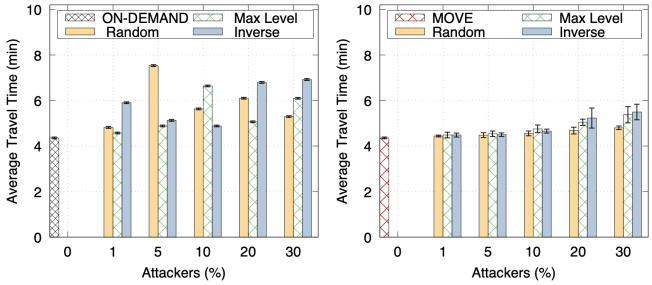
## IV. Evaluation

In this section, we present a performance evaluation by simulation of MOVE to support the service of traffic alert management. We implemented the MOVE by using tools that simulate vehicular communication, urban mobility, and the data poisoning (DaP) attacks model. In this way, we applied SUMO version 0.25 to coordinate and execute vehicular mobility and OMNET++ 5.1.1 and Veins 4.6 to ensure communication among vehicles. We evaluated both systems applying a Manhattan grid map under a region of 1 km$^2$ with the same parameters and specifications, described in [20], for instance, 120 road segments (two-way) with 200 meters each one, corresponding to 25 squares of the same size. For each simulation, we assessed a traffic density of 1000 vehicles/Km$^2$, where each vehicle travels a route composed of a random origin-destination pair. Further, MOVE sends the beacon dissemination at every two seconds and sets the value of three for both $thresholdattack$ and $thresholconsensus$ thresholds. We evaluated three types of data poisoning attacks: *Inverse, Max level, and Random*, as well as six different percentages of attackers: 1%, 5%, 10%, 20% and 30%. The following DaP attacks model are based on the work [10] and [18]. We look at metrics of traffic management performance and security when evaluating MOVE. For traffic management, we assess the **Average Travel Time (TT)**, **Average Time Lost in Traffic (TL)**, and **Average Speed (ASp)**, so that TT is related to the ability of the system to minimize the effects of traffic congestion; TL measures the extra time spent by vehicles to complete its trip under traffic jams, and ASp assesses how fast vehicles traveled in the network, where lower values mean a significant influence of traffic jams. For that, MOVE has been compared with the traffic management system named ON-DEMAND [4]. The security evaluation focuses on how MOVE identifies and excludes attackers. In the security evaluation, we analyzed only MOVE due the ON DEMAND ignore security issues, and applied the following metrics: **Attack Detection Rate (ATR)**, **Accuracy (AC)**, **False Positive (FP)**, **False Negative (FN)**, **Consensus Attacker Detection Rate (CADR)**, **Positive Consensus (PC)**, and **Negative Consensus (NC)**. In addition, we performed 33 simulations for each scenario (corresponding to the type of attack and a percentage of attackers), with a confidence interval of 95%. We carried out a comparative performance analysis between MOVE and ON-DAMAND, and we did not make a comparative security analysis because the closest work [16] focuses on mainly the trust among the devices against DDoS attacks and works on a centralized topology, being unfair to a comparative analysis.

### A. Results

The performance of MOVE and ON-DEMAND under traffic jams is shown in Fig. 4. As both systems are inspired by [4], they achieved similar behavior in a normal operation, but face to rising attacks is notable the difference of traffic information availability between them. The TT values seen in Fig. 4 indicate that MOVE gets to mitigate the poisoned traffic information through collaborative data validation. As ON-DEMAND does not employ a mechanism to detect and eliminate poisoned data, it takes into account these poisoned data in the traffic management. Further, the percentages of the DaP attackers in the network makes a substantial difference

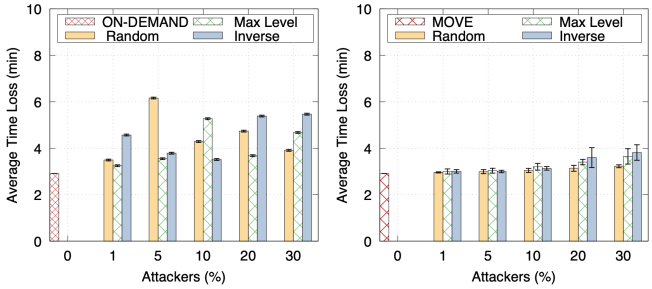
in relation to travel time provided by both systems. MOVE achieved a reduction of the average travel time by 40%, 7%, and 12% under attacks Random, Max level, and Inverse, respectively, compared to ON-DEMAND, for 5% of attackers. For 20% of attackers, MOVE decreases the travel time by 23% for Random, 1% for Max level, and 24% for Inverse.




(a) ON-DEMAND (b) MOVE

Fig. 4. Traffic management performance regarding travel time.

Reducing the time lost in transit can substantially mitigate the environmental impact caused by additional fuel consumption and $CO_2$ emissions. But, it is worth noticing that attacks make traffic decisions more challenging, with a higher chance of wrong decisions, and impacting the vehicle TL. Thus, the TL values in Fig. 5 show that even under a few attackers (1% and 5%) in all types of attacks, ON-DEMAND achieved a low performance, highlighting how complex the decision-making is. While MOVE slightly increased the TL up to 5% under 1%, 5%, and 10% of attackers. Further, with 20% of attackers, MOVE increased the TL up to 30% for all attacks.

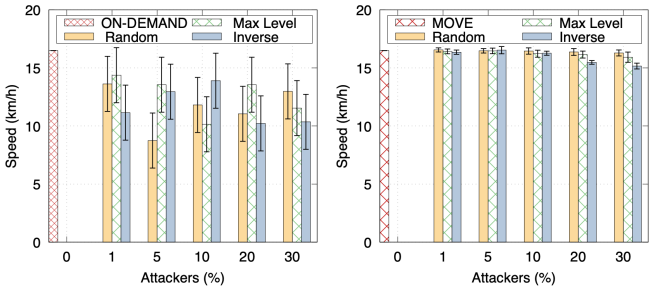



(a) ON-DEMAND (b) MOVE

Fig. 5. Average time lost due to traffic jams.

The values of average speed shown in Fig. 6 point out that attacks yielded high variability in the speed of vehicles with ON-DEMAND and a large confidence interval. Hence, it is essential the implementation of a mechanism for robust data poisoning detection and management. While MOVE achieved better results than ON-DEMAND in all evaluated attacks and densities, obtaining average speed around 17 $km/h$. We stand out that the attack Inverse is more dangerous to MOVE, generally decreasing the average speed around 3% compared to other attacks.

The graphics in Fig. 7 show the **ATR**, **AC**, and **FP**, and **FN** rates obtained by MOVE under the three types of DaP attacks. We note that MOVE reached an average of 87% detection for all scenarios and an average of 100% for scenarios with 1%,




(a) ON-DEMAND (b) MOVE

Fig. 6. Traffic management performance regarding average speed.

5%, and 10% of DaP attacks. These high ATRs corroborate with low FP and FN rates, where MOVE obtained an average of 4% for FPs and 10% for FNs. Further, MOVE reached AC values between 0.85 and 0.89, showing its capacity to correctly detect attackers. The wrong detection of an attack by MOVE is due to errors in calculating the consensus value among vehicles monitoring suspicious ones with low deviation from their readings. Firstly, those vehicles are classified as suspicious, but as new interactions and traffic information exchanges between vehicles take place, new calculations will allow the MOVE to identify the attack correctly. The attack detection effectiveness is due to the existence of the *watchdog* monitoring among all vehicles. Additionally, the suspect and attacker lists ensure the assertive detection of all DaP attacks. Further, with 20% of attackers, the MOVE rates remained stable for all security metrics.

TABLE I
EXCLUSION BY CONSENSUS

| Attacks | Metrics | Percentage of attackers 1% | 5% | 10% | 20% | 30% |
|---|---|---|---|---|---|---|
| Random | CADR | **76** | 63.6 | 40 | 18 | 11 |
| | PC | 1 | **0** | **0** | 0.7 | 0.3 |
| | NC | **0.02** | 1.9 | 6.66 | 20.3 | 38.1 |
| Max Level | CADR | **96** | 92 | 86 | 69 | 43 |
| | PC | **0** | **0** | **0** | **0** | 4 |
| | NC | **0** | 0.4 | 1.5 | 7.7 | 24.3 |
| Inverse | CADR | **97** | 93 | 81 | 63.7 | 39.9 |
| | PC | **0** | **0** | **0** | **0** | **0** |
| | NC | **0.03** | **0.03** | 2.1 | 9.5 | 25.7 |

Table I shows the effectiveness of relational consensus to exclude DaP attackers. This consensus takes into account attacker traffic information detected by at least three vehicles in the network ($thresholdconsensus$). We evaluated MOVE against DaP attacks in all scenarios by assessing the **CADR**, **NC** and **PC** rates. We noted that for attacks maximum and inverse level, MOVE remained stable with CADR rates varying between 75% and 80% and reaching 97% in some cases. It is noteworthy that MOVE under the random attack exhibited the worst results since the random attack behaviors are unpredictable in relation to other attacks. The **PC** and **NC** rates varied between 0.4% and 9%, meaning the exclusion of only legitimate attackers in the network. The effective exclusion of

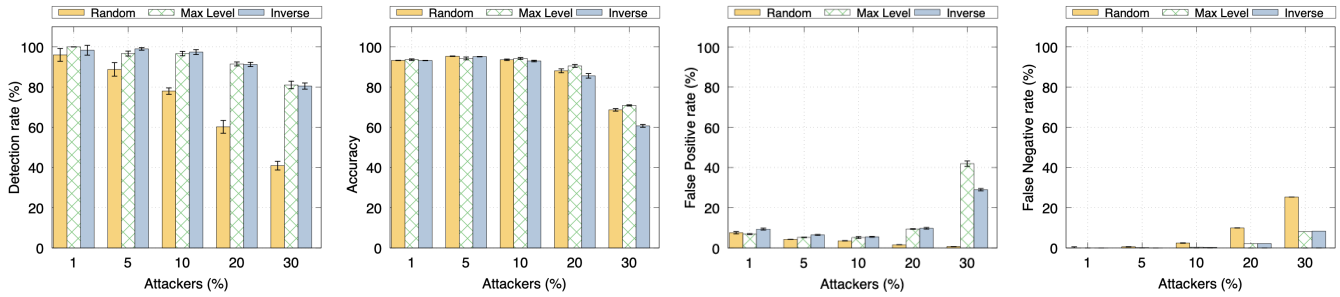



Fig. 7. Detection rate of DaP attacks,Accuracy, False Negative and False Positive Rates.

attackers is due to the relational consensus among vehicles and also the interactions among ones that have detected any type of DaP attacks.

## V. Conclusion

This work presented the MOVE system for robust traffic management against DaP attacks in VANETS. MOVE supports the collection and dissemination of traffic information to establish new routes with better travel times under congestion scenarios. To protect the system against data poisoning, MOVE relies on a watchdog technique to monitor the behavior of vehicles on the data information in the network and a relational consensus distributed by the similarity between neighboring vehicles to detect DaP attacks. Simulation results have shown MOVE's effectiveness greater than 80% in detecting, mitigating, and isolating three types of DaP attacks, thus ensuring that only authentic data traffic is available for the vehicles. As future works, we will evaluate the MOVE performance under the three types of DaP attacks acting simultaneously, as well as its performance by applying realistic vehicle tracks such as Cologne and Luxembourg. We will also compare the security of other TMSs under DAP attacks.

## References

[1] F. R. Soriano, J. J. Samper-Zapater, J. J. Martinez-Dura, R. V. Cirilo-Gimeno, and J. Martinez Plume, "Smart mobility trends: Open data and other tools," *IEEE Intelligent Trans. Systems Magazine*, no. 2, 2018.

[2] A. Thakur and R. Malekian, "Fog computing for detecting vehicular congestion, an internet of vehicles based approach: A review," *IEEE Intelligent Transportation Systems Magazine*, 2019.

[3] Y. V. Brandão, L. M. De Souza, T. S. Gomides, R. E. De Grande, F. S. H. Souza, and D. L. Guidoni, "A multi-layer and vanet-based approach to improve accident management in smart cities," in *2020 16th Int. Conf. on Distributed Comp. in Sensor Systems (DCOSS)*, pp. 165–172, 2020.

[4] T. S. Gomides, R. E. De Grande, A. M. de Souza, F. S. Souza, L. A. Villas, and D. L. Guidoni, "An adaptive and distributed traffic management system using vehicular ad-hoc networks," *Computer Communications*, vol. 159, pp. 317 – 330, 2020.

[5] D. L. Guidoni, G. Maia, F. S. H. Souza, L. A. Villas, and A. A. F. Loureiro, "Vehicular traffic management based on traffic engineering for vehicular ad hoc networks," *IEEE Access*, 2020.

[6] Z. Lu, G. Qu, and Z. Liu, "A survey on recent advances in vehicular network security, trust, and privacy," *IEEE Transactions on Intelligent Transportation Systems*, no. 2, 2019.

[7] M. Arif, G. Wang, M. Zakirul Alam Bhuiyan, T. Wang, and J. Chen, "A survey on sec. attacks in vanets: Comm., appli. and challenges," 2019.

[8] S. Su, Z. Tian, S. Liang, S. Li, S. Du, and N. Guizani, "A reputation management scheme for efficient malicious vehicle identification over 5g networks," *IEEE Wireless Comm.*, vol. 27, no. 3, pp. 46–52, 2020.

[9] A. Sen and S. Madria, "Risk assessment in a sensor cloud framework using attack graphs," *IEEE Transactions on Services Computing*, 2017.

[10] R. Deng, G. Xiao, R. Lu, H. Liang, and A. V. Vasilakos, "False data injection on state estimation in power systems—attacks, impacts, and defense: A survey," *IEEE Transactions on Industrial Informatics*, 2016.

[11] B. Li, R. Lu, W. Wang, and K.-K. R. Choo, "Distributed host-based collaborative detection for false data injection attacks in smart grid cyber-physical system," *Journal of Parallel and Dist. Computing*, 2017.

[12] A. L. Santos, C. A. Cervantes, M. Nogueira, and B. Kantarci, "Clustering and reliability-driven mitigation of routing attacks in massive iot systems," *JISA*, 2019.

[13] A. M. de Souza, N. L. S. da Fonseca, and L. A. Villas, "A fully-distributed advanced traffic management system based on opportunistic content sharing," *2017 IEEE Int. Conf. on Communications (ICC)*, 2017.

[14] M. A. Khan, M. S. Sheikh, and J. Liang, "A comprehensive survey on vanet security services in traffic management system," *Wireless Communications and Mobile Computing*, 2019.

[15] C. Zhang, K. Chen, X. Zeng, and X. Xue, "Misbehavior detection based on support vector machine and dempster-shafer theory of evidence in vanets," *IEEE Access*, 2018.

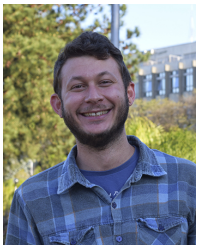
[16] Z. Tian, X. Gao, S. Su, and J. Qiu, "Vcash: A novel reputation framework for identifying denial of traffic service in internet of connected vehicles," *IEEE IOT Journal*, vol. 7, no. 5, pp. 3901–3909, 2019.

[17] J. Kamel, M. R. Ansari, J. Petit, A. Kaiser, I. B. Jemaa, and P. Urien, "Simulation framework for misbehavior detection in vehicular networks," *IEEE Transactions on Vehicular Technology*, 2020.

[18] C. Pedroso, A. dos Santos, and M. Nogueira, "Detecting fdi attack on dense iot network with distributed filtering collaboration and consensus," in *IEEE LATINCOM 2020*, nov 2020.

[19] M. Toulouse, B. Q. Minh, and P. Curtis, "A consensus based network intrusion detection system," in *2015 5th ICITCS*, IEEE, 2015.

[20] T. S. Gomides, "An adaptive and distributed traffic management system for vehicular ad-hoc networks," Master's thesis, Federal University of Sao Joao del Rei, 2020.

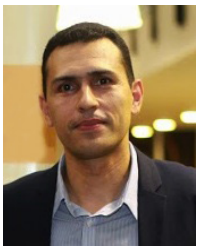


**Carlos Pedroso** is currently, a Ph.D. student at the Federal University of Paraná (UFPR). Bachelor in Computer Networks by the Faculty of Technology of São Paulo (FATEC) (2016). Master in Informatics from Federal University of Paraná (UFPR) (2019). Has experience in Computer Science, with emphasis on Hardware, Computer networks, Wireless Sensor Networks and Internet of Things, acting mainly on the following topics: IoT and Security.

**Thiago S. Gomides** is a visiting researcher in the Department of Computer Science at Brock University, Canada. He received his a Master degree in Computer Science (2020) from the Federal University of São João Del Rei, Brazil. His research topics include urban mobility, vehicular networks, and data communication. He was a visiting researcher through the Emerging Leaders in the Americas Program - ELAP (2019-2020) in the Department of Computer Science at Brock University, Canada.

**Daniel L. Guidoni** received his Ph.D. degree in computer science from the Federal University of Minas Gerais, Belo Horizonte, Brazil, in 2011. He is currently an Associate Professor with the Federal University of Ouro Preto, Minas Gerais, Brazil and former Associate Professor with the Federal University of São João del Rei, Minas Gerais, Brazil. His research interests include wireless networks, vehicular networks, IoT, Smart Cities, and communication protocols.

**Michele Nogueira** is professor of computer science at Federal University of Minas Gerais, where she has been since 2010. She received her doctorate in computer science from the University Pierre et Marie Curie – Sorbonne Universites, Laboratoire d'Informatique de Paris VI (LIP6) in 2009. She was a Visiting Researcher at Georgia Institute Technology (GeorgiaTech) and a Visiting Professor at University Paul Sabatier in 2009 and 2013, respectively.



**Aldri L. dos Santos** is professor of the Department of Computer Science at Federal University of Minas Gerais (UFMG). Aldri is PhD in Computer Science from the Federal University of Minas Gerais, Master in Informatics and Bachelor of Computer Science at UFPR. Aldri working in the following research areas: network management, fault tolerance, security, data dissemination, wireless ad hoc networks and sensor networks. He is leader of the research group (Wireless and Advanced Networks).