Credibility Enhanced Temporal Graph Convolutional Network Based Sybil Attack Detection On Edge Computing Servers

Baiting Luo, Xiangguo Liu, Qi Zhu

Abstract— **The emerging vehicular edge computing (VEC) technology has the potential to bring revolutionary development to vehicular ad hoc network (VANET). However, the edge computing servers (ECSs) are subjected to a variety of security threats. One of the most dangerous types of security attacks is the Sybil attack, which can create fabricated virtual vehicles (called Sybil vehicles) to significantly overload ECSs' limited computation resources and thus disrupt legitimate vehicles' edge computing applications. In this paper, we present a novel Sybil attack detection system on ECSs that is based on the design of a credibility enhanced temporal graph convolutional network. Our approach can identify the malicious vehicles in a dynamic traffic environment while preserving the legitimate vehicles' privacy, particularly their local position information. We evaluate our proposed approach in the SUMO simulator. The results demonstrate that our proposed detection system can accurately identify most Sybil vehicles while maintaining a low error rate.**

I. Introduction

The development of edge computing has the potential to bring revolutionary changes to intelligent transportation systems. Compared with vehicular cloud computing, the servers for vehicle edge computing (VEC) are deployed on the roadside units (RSUs) in proximity to the vehicles, and the processing and analysis of data collected by vehicles' onboard units (OBUs) or RSUs themselves take place on those edge computing servers (ECSs). Therefore, VEC has the advantages of low latency and high context awareness, which can substantially support the vehicular ad hoc network (VANET) to enhance the driving safety applications [1].

However, ECSs are subjected to a variety of security threats. The work in [2] summarizes the vehicular edge security issues into five categories, which are sensor security, operating system security, control system security, vehicle to everything (V2X) security, and security for edge network and platforms. In V2X security, the Sybil attack is one of the most severe types of security attacks because the attacker can create multiple fabricated vehicles, called *Sybil vehicles*, with stolen IDs to launch various attacks such as information forgery attacks. Furthermore, the attacker can use the Sybil vehicles to maliciously overload RSUs' computing and storage capacity, which may disrupt the legitimate vehicles from using ECSs for their applications. This may consequently disrupt the traffic flow and increase the chance of accidents, especially when the traffic is dense.

Due to the hazard of Sybil attacks in vehicular networks and applications, various Sybil detection schemes have been proposed to defend against them. Received signal strength indicator (RSSI) based scheme utilizes the RSSI value obtained from Basic Safety Messages (BSMs) to identify malicious nodes [3], [4]. Trajectory-based scheme captures sequential characteristics of vehicular trajectories to filter out malicious nodes [5]. Cryptography-based scheme assigns vehicles with pseudonyms and leverages signatures or secrete keys to defend against Sybil attacks while preserving privacy [6], [7]. However, one of the most significant schemes in general Sybil detection, the graph-based schemes [8], [9] are rarely discussed in the context of VANET. Graph-based schemes construct a graph based on nodes' social relationship, and then use both topological and node embedding information to detect attacks. Furthermore, to achieve computationally efficient graph analysis, Graph Convolutional Network (GCN) [10] is developed. It has shown its powerful capability in generating meaningful node embedding by fusing the structural information aggregated from a node's neighborhood with the node's self features to detect spammers in social networks [11], [12]. When applied to VANET, graph-based schemes can leverage connected autonomous vehicles (CAVs)' inherent connectivity to create the preliminary model but face the challenge of CAVs' highly dynamic mobility, which may lead to volatile graph structures. Therefore, to fully leverage GCN's information aggregation capability while capturing and utilizing the dynamic information brought by vehicle s' mobility, complementing the GCN with Gated Recurrent Unit (GRU) [13] is a promising idea.

In this paper, we propose a credibility enhanced Sybil attack detection system based on temporal graph convolutional network (TGCN) to secure ECSs while preserving the CAVs' local position information for privacy. As shown in Fig. 1, the detection system uses the TGCN classifier to make the real-time classification based on nearby traffic information and physical characteristics reported in V2X messages. Therefore, the commonly implemented features such as trajectories, driving patterns are substituted with topological information extracted from CAVs' connectivity, which reserves CAVs' position information for privacy and enables the proposed Sybil detection system to defend against a more complicated attack model. The classification results are then used for building up the CAVs' local credibility. We also introduce Bayesian inference to integrate a global credibility, which is the vehicle's accumulative credibility in previous ECSs, with the vehicle's local credibility. The updated final

Baiting Luo, Xiangguo Liu, and Qi Zhu are with the Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL 60201, USA. baitingluo2019@u.northwestern.edu, xg.liu@u.northwestern.edu, qzhu@northwestern.edu.

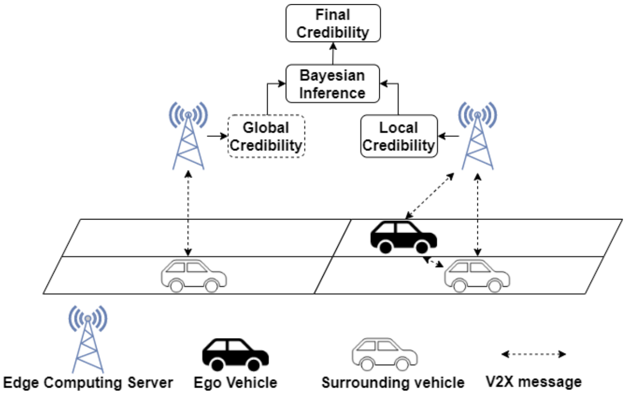

Fig. 1: Overview of our Sybil attack detection system design. The ego vehicle travels in the region monitored by the current ECS and sends the surrounding traffic information to it. TGCN based classifier will use the reported information to classify the vehicle's identity continuously; the results will be used to build up local credibility. The current ECS then combines the vehicle's local credibility with its global credibility from the previous ECS by Bayesian inference to update final credibility. Then, the updated final credibility will be used to make the final classification.

credibility will be used for final classification. Our contribution in this paper can be summarized as follows:

- We developed a novel Sybil attack detection system implemented on ECSs to provide effective protection for ECSs with graphs modeled by CAVs' connectivity while preserving CAVs' local position information for privacy.
- We designed a credibility accumulation system and introduce a handoff protocol to integrate the CAVs' credibility in the current server with accumulative credibility from previous servers to further improve the detection system's performance.
- We implemented a prototype of the proposed detection system in SUMO [14] and demonstrated its effectiveness in defending against Sybil attacks in experiments.

II. Related Works

A number of methods have been proposed to defend against Sybil attacks in VANET. Among them, learning-based techniques are one of the most commonly implemented approaches in the Sybil detection system. For instance, [15] implements Naive Bayes classifier, Support Vector Machine (SVM), Decision Tree with ten mobility features that are extracted from the location information reported by vehicles. The work in [16] uses edge base stations to collect each vehicle's location and mobility information within a certain time period, and uses the data to define a matrix of movement. The matrix is then processed and trained with extreme learning machine to detect Sybil attacks. The works in [17], [18] propose an SVM based and a k-nearest neighbors based Sybil attack detection methods by utilizing the extracted eigenvalues from the mobility matrix, which summarizes vehicles' location, velocity and acceleration information. However, these mobility-based learning methods require vehicles continuously reporting their position information to RSUs, which is detrimental for maintaining vehicles' privacy. Besides, the methods are vulnerable to the hybrid attack model, e.g., the Sybil vehicles launch the DoS attack or information forgery attack, which is very common when a Sybil attack is initiated. In addition to the learning-based Sybil detection system deployed on RSUs, [19] proposes a novel feature extraction algorithm deployed on the vehicles to get the vehicles' nearby traffic flow, distances between each other, and reported position biases. The features are then trained with an improved growing hierarchical self-organizing map to assist the vehicles with detecting intrusion. Although the proposed method can achieve a high detection rate, it creates lots of computation burdens for vehicles' OBUs and is unable to be deployed on RSUs as each vehicle has prior knowledge of its identity while RSUs don't have such information. There are also non-learning based detection systems. The work in [20] tries to filter out Sybil vehicles by evaluating the similarity of physical parameters such as distances and angles between the vehicle and RSU. But the signal noises in the highly dynamic traffic environment are not considered. [21] uses the neighboring nodes tables exchanged among vehicles to help the legitimate vehicles find Sybil vehicles. The nodes that are simultaneously observed for a period of time will be classified as Sybil vehicles. However, this method is based on the assumption that a vehicle will not keep following another vehicle beyond a specific period. It is hard to establish such an assumption for the one-way traffic scenario such as the highway. In [22], Boolean constraints that are formed by mutually reported information are added to a SAT solver, and the SAT solver then outputs the sets of possible honest vehicles. The set of vehicles with the minimum speed discrepancy between the data reported by them and the data reported by trusted sensors is used to compute the final traffic data. However, the work does not consider that the attacker can acquire nearby traffic information and then let the Sybil vehicles report legitimate vehicles to further confuse the intrusion detection system.

In summary, the existing studies either make strong assumptions on Sybil vehicles' driving patterns and trajectories or do not consider a more comprehensive attack model. In our work, we propose a graph-based Sybil detection system to identify malicious CAVs with CAVs' nearby traffic information and their physical characteristics; therefore, our method relies on the latent pattern unearthed from the topological information instead of the similarity of driving patterns between the attacker and the Sybil vehicle to detect Sybil attacks. Hence, the CAVs' local position information is substituted with topological information to preserve their local privacy. Furthermore, we assume that Sybil vehicles behave the same as legitimate vehicles and report the information similar to the information reported by legitimate vehicles to avoid having a specific similarity in driving patterns or trajectories with the attacker. We also discuss a more comprehensive Sybil attack model, as well as attackers' and Sybil vehicles' possible strategies in falsifying V2X messages.

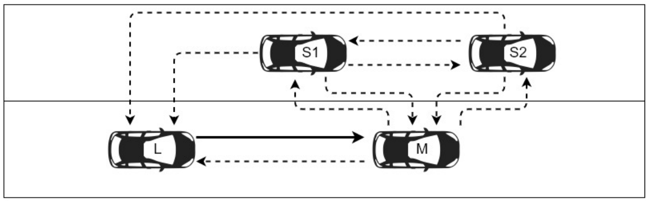

Fig. 2: Attack model considered in our work. The vehicles with different identities will take different reporting strategies to report the existence of the other vehicles in their communication range. The solid line indicates that the reporting information must exist, while the dashed line indicates the existence of the reporting information depends on the vehicle's reporting tactic. The solid arrow points from vehicle L to vehicle M means the legitimate vehicle L must report the existence of the Malicious vehicle M. The dashed line from $S1$ to L means the Sybil vehicle $S1$ can choose to report or ignore the legitimate vehicle L. The dashed line from $S2$ to $S1$ means the Sybil vehicle $S2$ can choose to report or ignore the Sybil vehicle $S1$.

TABLE I: REPORTING TACTICS FOR ATTACKER AND SYBIL VEHICLES

	Report / Not Report
Attacker	• Sybil vehicles launched by it • Legitimate vehicles it can detect
Sybil	• The other Sybil vehicles launched by the same attacker • The attacker launched it • Legitimate vehicles that the attacker can detect

III. Attack Model

In Sybil attacks, a malicious vehicle on the road will report the existence of fabricated virtual vehicles, i.e., Sybil vehicles, to ECSs. Moreover, those Sybil vehicles will compete for edge computing resources by sending requests to ECSs. When the traffic becomes heavy and demand on edge computing becomes more extensive, the attacker and its fabricated Sybil vehicles may be able to overload the limited resources on the ECSs. Consequently, the legitimate vehicles' computing requests to the ECSs could be disrupted, and they may have to slow down for safety concern. The efficiency of the traffic flow could be significantly reduced, and the chance for accidents may increase.

Fig. 2 represents the structure of the graph modeled by the vehicles with different identities. During Sybil attacks, the attacker may attempt different strategies to disguise the Sybil vehicles' identities. In our assumption, the RSUs and ECSs are attack-resilient, so the attackers cannot access the data reported to the infrastructures, but they can falsify the messages' information. Therefore, Sybil vehicles can intentionally choose the vehicles which they want to report to RSUs, but they are incapable of obtaining information beyond what the attacker can access. With all these assumptions, we can summarize attacker's and Sybil vehicles' possible reporting tactics in Table I. Attackers can choose to report or ignore the Sybil vehicles fabricated by it or the legitimate vehicles which it can detect. Meanwhile, the Sybil vehicles can be instructed by the attacker to report or ignore the attacker, the other Sybil vehicles launched by the same attacker, and the legitimate vehicles which the attacker can detect.

It is assumed that legitimate vehicles will report and only report what they observe. We also assume perfect perception and communication for legitimate vehicles. Thus if there is any inconsistency, there should exist attacks. However, note that any inconsistency between reported information cannot be used to identify attackers [22]. For example, a legitimate vehicle L honestly reports an attacker M, but the attacker M chooses to ignore L, and it also instructs Sybil vehicles $S1$ and $S2$ launched by it to ignore L in the messages. L cannot detect $S1$ and $S2$ and therefore cannot mention them in its message. As a result, L may be mistakenly identified as a Sybil vehicle while the real attacker violates the rules with impunity because Sybil vehicles become the majority.

Furthermore, we do not assume that the attacker and the Sybil vehicles will have any distinctive physical characteristics such as speed, acceleration, driving pattern, i.e., they are assumed to perform similarly to the legitimate vehicles on the road.

IV. Methodology

A. Vehicular Ad Hoc Networks Model

1) Privacy-Preserving Vehicle-to-Infrastructure Communication Protocol: In VANETs, we assume that all vehicles use dedicated short range communication (DSRC) but are not limited to DSRC to communicate with everything. Each vehicle periodically communicates with its neighbors via beacon messages, allowing the vehicles to know the information such as velocity and pseudonyms of their surrounding vehicles. Furthermore, in our protocol, for preserving privacy, each vehicle will use the pseudonym assigned by ECS to make both V2V communications and V2I communications; also, vehicles do not need to report their local position information. Besides, we discuss the Sybil detection system's performance with partial-privacy communication protocol (PPCP) and full-privacy communication protocol (FPCP). The difference between PPCP and FPCP is that PPCP will require vehicles to report the distance between themselves and their neighbors in the communication range, whereas FPCP will not have such a requirement. Therefore, at time t, each vehicle i is able to acquire a list $L_t^i = \{\{i, j, distance_{i,j}\}, \{i, k, distance_{i,k}\}, \cdots\}$ in PPCP or $L_t^i = \{\{i, j\}, \{i, k\}, \cdots\}$ in FPCP to indicate its neighbors' pseudonyms, the number of its neighbors N_t^i, and possibly the distances between itself and its neighbors depending on the communication protocol.

Furthermore, based on the Green-shield's model [23], which describes the relationship between speed and density, each vehicle i can also obtain its nearby traffic flow $flow_t^i$ at time t with:

$$\hat{V}_t^i = V_{max}^i - \frac{\rho_t^i}{\rho_{max}} \times V_{max}^i \tag{1}$$

$$Flow_t^i = \hat{V}_t^i \times \rho_t^i \tag{2}$$

where ρ_t^i is the traffic density in i's communication range at time t, $\hat{V}_t^i$ is i's theoretical average speed at density ρ_t^i,

V_{max}^i is i's speed when traffic density is 0, and ρ_{max} is the traffic density when all vehicles' speed become 0 on the road. We assume that the attackers and Sybil vehicles will honestly report $Flow_t^i$ because only ρ_t^i can be falsified in Equation (1), whereas ECSs can quickly identify the falsified ρ_t^i with N_t^i. Besides, each vehicle can request the computation resource allocation from the ECS in the message. We use α_t^i to denote the amount of the resource vehicle i requests or holds from the ECS at time t. And we also use $Velocity_t^i$ to denote vehicle i's velocity at time t. Overall, the format of message M_t^i that each vehicle i is required to periodically send to ECS_a is shown in Equation (3).

$$M_t^i = \{L_t^i, flow_t^i, \alpha_t^i, N_t^i, Velocity_t^i\} \tag{3}$$

So each ECS_a will have a message list $M_t^a = \{M_t^i, M_t^j, \cdots\}$ from all vehicles in its monitored region at time t.

B. Credibility-Enhanced Temporal Graph Convolutional Network based Sybil Detection System

1) Vehicle-to-Vehicle based Graph: For the vehicles run on the road, each vehicle's behavior is more likely to be influenced by a closer vehicle, which means each vehicle's behavior is inversely related to the distance between itself and its neighbors. Thus, if vehicles are using PPCP, ECS_a will use the message list M_t^a to construct the edge-weighted graph $G_t^a = (V_t^a, E_t^a)$ with edge weights $W(e_{i,j}) = \frac{100}{distance_{i,j}}$, $e_{i,j} \in E_t^a$ to describe the spatial and temporal relationship among the vehicles at time t, where V_t^a is the set of vehicles that request resources or hold resources from ECS_a at time t, and E_t^a is the set of detection information reported to ECS_a at time t. If vehicles are using FPCP, G_t^a will be an unweighted graph.

For $vehicle_1 \in V_t^a$ and $vehicle_2 \in V_t^a$, they will form an undirected edge $e_{1,2} \in E_t$ when either $vehicle_1$ or $vehicle_2$ reports the other's existence in L_t^1 or L_t^2. As we assume that attackers and Sybil vehicles can take any strategy to disguise their identities or disturb the $ECSs$' operation. Forming undirected edges by this method can make sure that the legitimate vehicles' information can always be aggregated with malicious vehicles, as well as being aggregated with Sybil vehicles if they choose to report the legitimate vehicles. For example, if an attacker M chooses to ignore the legitimate vehicle L, L's features will still be passed to M while implementing GCN. Then for time t, an adjacency matrix $A_t^a \in \mathbb{R}^{N\times N}$ can be extracted from G_t^a, and an initial node feature matrix $X_t^a \in \mathbb{R}^{N\times D}$ can be extracted from M_t^a.

2) Temporal Graph Convolutional Network: V2V based graph has its distinctions in several aspects: 1) traveling vehicles frequently changing their positions results in the highly inconstant graph; 2) each node's embedding information varies with time, such as speed, acceleration; 3) new nodes keep joining in, while previous nodes keep leaving out. To fully utilize these dynamic features, a TGCN module is introduced in our detection system. Evolving Graph convolution network (EvolveGCN) [24] consists of the graph convolution unit [10] and a gated recurrent unit (GRU). The principle of the model is to use a recurrent architecture for updating the weight matrix with historical information and then use the updated weight matrix for updating the current time step's node feature matrix. Hence, in our application, the GCN is only used to extract the meaningful weight matrix at the specific time step; GCN's function is thereby not limited to the V2V based graph's changes with the time anymore. The graphical changes will be learned via GRU. By applying our V2V based graph to the model, the layer-wise propagation rule can be mathematically written as:

$$W_t^l = GRU(X_t^l, W_{t-1}^l) \tag{4}$$

$$X_t^{l+1} = ReLU(\hat{A}_t \hat{D}_t^{-1} X_t^l W_t^l) \tag{5}$$

where $\hat{A}_t = A_t + I_N$ is the adjacency matrix A_t of graph G_t^a at time t added with identity matrix I_N, $\hat{D}_t$ is the degree matrix, X_t^l and W_t^l are respectively the node embedding matrix and the weight matrix at the l-th layer. In Equation (4), GRU updates the hidden state W_t^l with weight matrix W_{t-1}^l in the last step and X_t^l at the current time step. Next, W_t^l is fed into GCN's propagation rule and output a new node embedding X_t^{l+1} for $(l+1)$-th layer in Equation (5). Finally, X_t^{l+1} can be either further updated by weight matrix W_t^{l+1} to X_t^{l+2} or directly output for classification depending on the setting of GCN's layer number.

Moreover, in Equation (4), the input X_t^l's dimension has to match the hidden state W_{t-1}^l's dimension to implement GRU, so we use the same method described in [24], [25] to choose top-k nodes to achieve that. The expression is given below:

$$scores_t^l = X_t^l p^l / \|p^l\| \tag{6}$$

$$topk_t = (scores_t^l, k) \tag{7}$$

$$\tilde{X}_t^l = ([X_t^l \odot tanh(scores_t^l)](topk_t, :))^T \tag{8}$$

$$Z_t = \sigma(V_Z \tilde{X}_t^l + U_Z W_{t-1}^l) \tag{9}$$

$$R_t = \sigma(V_R \tilde{X}_t^l + U_R W_{t-1}^l) \tag{10}$$

$$\tilde{W}_t^l = tanh(V_H \tilde{X}_t^l + U_H (R_t \odot W_{t-1}^l)) \tag{11}$$

$$W_t^l = (1 - Z_t) \odot W_{t-1}^l + Z_t \odot \tilde{W}_t^l \tag{12}$$

where p^l is a trainable vector at layer l, $\|p^l\|$ is L_2 norm of p^l, $\odot$ is element-wise multiplication, $\sigma(\cdot)$ is sigmoid function, Z_t and R_t are respectively update and reset gates, matrices U, V are trainable parameters which are adjusted during training. In Equation (7), the indices of the k nodes with largest values in $scores_t^l$ are chosen and output as $topk_t$. In Equation (8), the transpose of the matrix consisting of the nodes in X_t^l with corresponding indices in $topk_t$ are subsequently output. As k is equal to the number of columns of W_{t-1}^l, $\tilde{X}_t^l$ thereby has the same dimension as W_{t-1}^l.

3) Credibility Enhanced Detection System: The ECS will classify the vehicles' identities every second with the TGCN model in our design. Accordingly, at time t, ECS_a will use the classification results to evaluate and credit $Prediction_t^{VID_a}$ points to vehicle VID_a based on the messages reported by VID_a in the last 5 seconds. If VID_a is classified as a legitimate vehicle at time t, $Prediction_t^{VID_a}$ will be 1; otherwise, it will be 0. We also denote by $Local_L^{VID_a}$ ECS_a's confidence in VID_a's legitimate identity since VID_a enters ECS_a's monitored region; the closer $Local_L^{VID_a}$ approaches 1, the more system is confident in VID_a's legitimate identity. Besides, both $Prediction_t^{VID_a}$ and $local_L^{VID_a}$ are confidential to VID_a. $Local_L^{VID_a}$ is computed as:

$$Local_L^{VID_a} = \frac{1}{t}(\sum_{n=1}^{t} Prediction_n^{VID_a}) \tag{13}$$

Algorithm 1 Handoff Protocol

```
Result: $Global_L^{VID_{new}}$, $Local_L^{VID_{new}}$, $VID_{new}$
Input: $VID_{old}$, current time: $t$, $C^{VID_{old}}$
if $VID_{old}$ starts receiving the beacon message from $ECS_{new}$ then
    $VID_{old}$ sends leaving message to $ECS_{old}$;
    $ECS_{new}$ assigns $VID_{new}$ to $VID_{old}$;
    if $ECS_{old}$ receives the leaving message then
        $T_l = t$;
        $ECS_{old}$ sends the key to $ECS_{new}$ and $VID_{old}$, $C^{VID_{old}}$ to $ECS_{new}$;
        $Global_L^{VID_{new}} = C^{VID_{old}}$;
        if $Global_L^{VID_{new}} == 0$ then
            $Global_L^{VID_{new}} = 0.1$;
        end
        if $Global_L^{VID_{new}} == 1$ then
            $Global_L^{VID_{new}} = 0.9$;
        end
        while $VID_{old}$ has not been recycled do
            if $VID_{old}$ offloads service on $ECS_{old}$ then
                $VID_{old}$ continues offloading computations on $ECS_{old}$;
            end
            Pre-$Local_L^{VID_{new}} = \sum_{n=T_l}^{t} Prediction_n^{VID_{old}}$;
        end
        $ECS_{old}$ frees the resources occupied by $VID_{old}$;
        $T_f = t$;
        $Local_L^{VID_{new}} = \frac{1}{T_f - T_l}$(Pre-$Local_L^{VID_{new}}$);
    end
end
```

For a large traffic network, multiple ECSs will be distributed to cover all regions comprehensively. Therefore, the offloaded computation should be moved to a new ECS [26] when the vehicle drives away from its current ECS. As shown in Algorithm 1, we define a handoff region communication process[1] to ensure that our detection system seamlessly performs its function while vehicles are traveling from the current ECS to a new ECS and further enable servers' cooperation in detecting Sybil attacks. We assume that the current server ECS_{old} and the next server ECS_{new} have a 1 km mutually covered area named the "handoff" region. All the vehicles driving from the region monitored by ECS_{old} to the region monitored by ECS_{new} are required to submit the request of changing server in this region. Also, handoffs can be categorized into hard handoff, soft handoff, horizontal handoff, vertical handoff [27]. In our design, we assume that all vehicles use soft handoff, which means that the connection between ECS_{old} and the vehicle breaks after ECS_{new} establishes the connection with the vehicle. Therefore, at time T_l, for the vehicle VID_{old} has offloaded service on ECS_{old}, it starts sending the leaving message, which contains the information of ECS_{new}, to ECS_{old} when it builds up the connection with ECS_{new}. ECS_{new} will assign the vehicle a new pseudonym VID_{new}, then the vehicle will use VID_{new} to communicate with ECS_{new}. Furthermore, ECS_{old} starts migrating the offloaded service to ECS_{new}, as well as sending VID_{old}'s final credibility $C^{VID_{old}}$, which will be converted to $Global_L^{VID_{new}}$ by ECS_{new} as VID_{new}'s global credibility, to ECS_{new} at T_l, and a key to both VID_{old} and ECS_{new}. After the migration process is finished, the vehicle will use the key to match the migrated offloaded service and $C^{VID_{old}}$ on ECS_{new}, and this time is denoted by T_f. During $(T_f - T_l)$, ECS_{new} starts computing Pre-$Local_L^{VID_{new}}$, and the vehicle still connects with ECS_{old} to offload the computations. After T_f, ECS_{old} will release the resources occupied by VID_{old} and recycle the pseudonym VID_{old}, then ECS_{new} will use pre-$Local_L^{VID_{new}}$ and VID_{new}'s previous final credibility to compute $C^{VID_{new}}$ for classification. For vehicle VID_{old}, which does not offload service on ECS_{old}, ECS_{old} will directly send the vehicle's final credibility and the key to ECS_{new}. The vehicle can still send the request of offloading computations at this time, but the request will not be reviewed by ECS_{new} until the match process is completed. Overall, this process can ensure any resources allocated by the previous ECS will eventually be released, and all honest vehicles will eventually transfer from one server to another server without service interruption.

To compute the final credibility of a vehicle for classification and further strengthen distributed ECSs' cooperation in filtering out Sybil vehicles, Bayesian Inference [28] is introduced in our design. Bayesian Inference takes prior information to establish trust and has been widely implemented to build a reputation system [29]. In our design, each ECS integrates the vehicle's local credibility with its prior global credibility by Bayesian Inference to get VID_{new}'s $C^{VID_{new}}$, then $C^{VID_{new}}$ is used to make the final classification. $C^{VID_{new}}$ is computed as:

$$C^{VID_{new}} = \frac{Global_L^{VID_{new}} \times Local_L^{VID_{new}}}{\sum_{I=L,S} Global_I^{VID_{new}} \times Local_I^{VID_{new}}} \tag{14}$$

where $Global_L^{VID_{new}}$ is the prior probability equal to $C^{VID_{old}}$,

[1] The techniques to migrate the offloaded work from a server to a new server is beyond the scope of this paper, which focuses on designing handoff communication process to enable distributed ECSs' cooperation.

which can be obtained from the message sent from ECS_{old} to ECS_{new}; $Global_S^{VID_{new}} = 1 - Global_L^{VID_{new}}$ and $Local_S^{VID_{new}} = 1 - Local_L^{VID_{new}}$ are respectively global and local confidence in VID_{new}'s Sybil identity. We also set a threshold $\sigma \in [0,1]$ to offer system the adjustability. The closer σ approaches 1, the more sensitive the system will be. For instance, when σ is set to 0, only the vehicle classified as Sybil vehicle since it enters the current monitored region will finally be classified as the Sybil vehicle. The ECS's classification process discussed above is summarized in Algorithm 2.

Algorithm 2 ECS Detection Process

```
Result: Resource Allocation Approval Result
Input: $VID_{new}$, current time: $t$, $Global_L^{VID_{new}}$,$\sigma$,
while $VID_{new}$ in the region do
    if $VID_{new}$ is classified as Sybil vehicle at time $t$ then
        $Prediction_t^{VID_{new}} = 0$;
    end
    if $VID_{new}$ is classified as Legitimate vehicle at time $t$ then
        $Prediction_t^{VID_{new}} = 1$;
    end
    $Local_L^{VID_{new}} = \frac{1}{t}(\sum_{n=1}^{t} Prediction_n^{VID_{new}})$;
    $C^{VID_{new}} = \frac{Global_L^{VID_{new}} \times Local_L^{VID_{new}}}{\sum_{I=L,S} Global_I^{VID_{new}} \times Local_I^{VID_{new}}}$;
    if $C^{VID_{new}} > \sigma$ then
        $VID_{new}$ is classified as legitimate vehicle;
        Approve the resource allocation request from $VID_{new}$;
    else
        $VID_{new}$ is classified as Sybil vehicle;
        Reject the resource allocation request from $VID_{new}$;
    end
end
```

V. Experimental Results

A. Dataset and Parameters

To validate the effectiveness of the proposed detection system, we use the SUMO simulator to simulate the traffic and V2I, V2V communications in the highway scenario. We assume that each 10 km will be equipped with an ECS, and we focus on analyzing the performance of the proposed Sybil detection system on this server. Simulation parameters are listed in Table II. The vehicles run on a 2-lane highway with a maximum speed of 112 km/h. The total lane length is set to 10 km. Each vehicle can communicate with its surrounding vehicles within 300 meters transmission range [30].

B. Implementation

Within SUMO, we dispatch vehicles requesting different amounts of the resource with an arrival rate of 0.2 vehicle/s. Each attacker runs in the simulation scenario and launches the Sybil attacks by generating five Sybil vehicles once it starts sending messages to the ECS. The Sybil vehicles also start sending the messages based on the traffic information detected by the attacker. Therefore, the attacker's behavior is similar to any other legitimate vehicles except that the attacker will fabricate reporting information. Moreover, to follow the attack model we discussed in III, the amount of requested resource and velocity reported by the Sybil vehicles varied in a reasonable range to avoid the consistent pattern with the attacker. Moreover, two reporting strategies are implemented in the simulation and analyzed separately. The first one is named inconsistent attackers. For this type of attackers, both Sybil vehicles and attackers randomly choose the reporting targets in the communication range; thus, the Sybil cluster patterns in our experiments are thereby comprehensive and unpredictable. The second one is named silent attackers that all Sybil vehicles pretend there are no other vehicles in their communication range.

TABLE II: SIMULATION PARAMETERS

Parameters	Values
Simulation Scenario	Highway
Simulation Time	6000 seconds
Lanes Number	2
Vehicles Maximum Speed	112 km/hr
Vehicle Arrival Rate	0.2 vehicle/s
Communication Range	300 m
Percentage of Attacker	1% 3% 5% 8% 10% 20%

We also implement the simulations in both FPCP and PPCP, and focus on analyzing the percentage of attackers' influence by observing the results of testing group with the same size but with different attacker percentages. Also, we analyze the effectiveness of different σ values in detecting Sybil vehicles. The TGCN module will make the classifications based on last 5 time steps' information. Due to imbalanced datasets, the results are evaluated with precision, recall and F1 scores [31]:

$$Precision = \frac{TP}{TP + FP} \tag{15}$$

$$Recall = \frac{TP}{TP + FN} \tag{16}$$

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall} \tag{17}$$

where TP, FP, FN are respectively true positives, false positives and false negatives.

Fig. 3 shows F1 scores reached by our proposed Sybil detection system on detecting silent attackers. Broadly speaking, our system can accurately find almost all Sybil vehicles which implement the silent strategy with any σ values, and reach maximum performance when sigma is set in a range from 0 to 0.3.

For inconsistent attackers, we compare the F1 scores between the detection systems that implement PPCP either

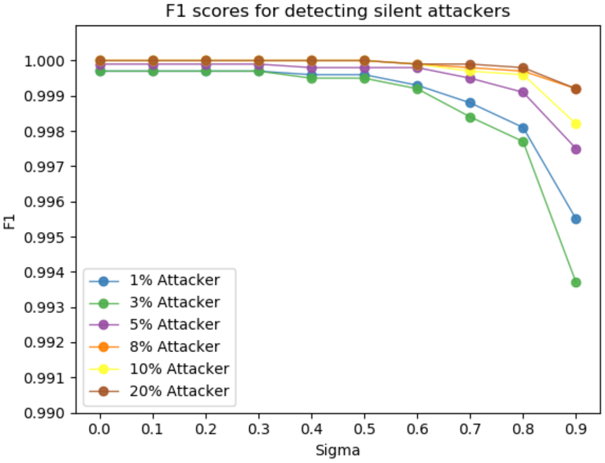

Fig. 3: F1 scores for Sybil detection system on datasets with different percentages' silent attackers.

TABLE III: F1 SCORE COMPARISON BETWEEN CREDIBILITY ENHANCED TGCN AND TGCN

	1%	3%	5%	8%	10%	20%
Credibility Enhanced TGCN	0.937	0.904	0.921	0.924	0.899	0.923
TGCN	0.658	0.717	0.770	0.822	0.814	0.873

with the credibility system or without the credibility system in Table III. The table shows the system's performance with different percentages of attackers. It can be observed that the smaller the percentage of attackers is, the more helpful the credibility system will be.

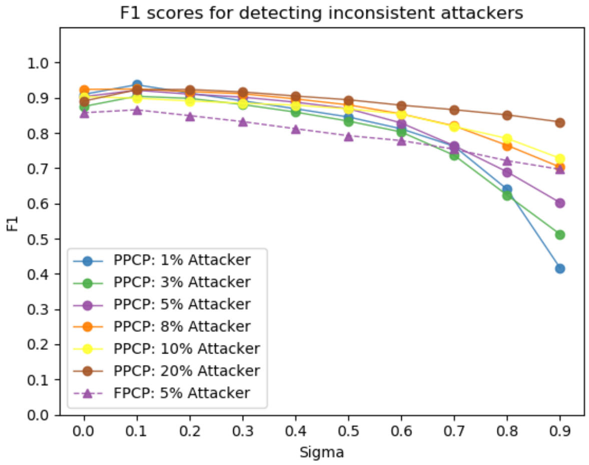

Fig. 4: F1 scores for Sybil detection system that implements PPCP or FPCP with different percentages' inconsistent attackers. For simplicity, we only plot the result for the simulation that implements FPCP with 5% attacker. Generally speaking, the best F1 score for the simulation that implements FPCP is lower than the simulations that implement PPCP.

In addition, as illustrated in Fig. 4, our proposed Sybil detection system can also accurately detect most Sybil vehicles with the inconsistent strategy when σ is set to a relatively small value, for both FPCP and PPCP. For the detection system that implements FPCP, its detection performance is slightly lower than the system that implements PPCP due to the lack of distances as the feature. The detection system that implements PPCP also reaches its best performance when σ is set in a range from 0 to 0.3; after the range, the system's performance decreases with the increase of σ value. As we discussed regarding the usage of σ, the larger σ is, the more sensitive the system will be; therefore, the detection system tolerates fewer uncertainties on the vehicles' legitimate identities with a large σ value.

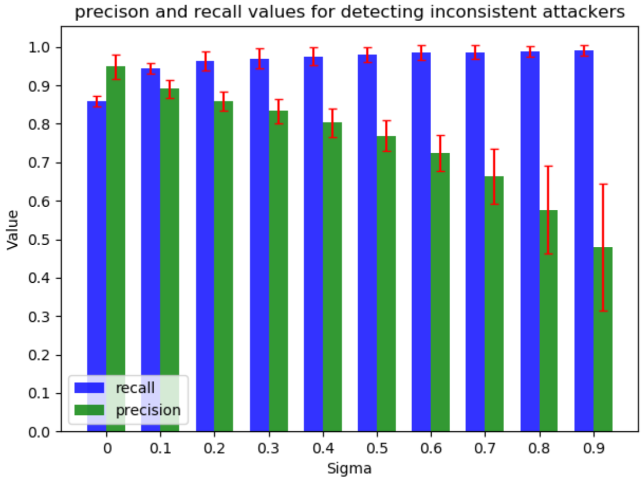

Fig. 5: Recall and precision values for Sybil detection system that implements PPCP with different percentages of inconsistent attackers. The presented recall or precision values are averaged results for the six testing groups with different σ values. The error bar is the standard deviation of the results for the six testing groups.

Besides, as shown in Fig. 5, for the simulations that implements PPCP, both recall and precision reach high values with certain σ values and have the expected changes with the increase of σ value. The precision decreases significantly with the increase of σ value because more and more legitimate vehicles are classified as Sybil vehicles by an increasingly sensitive detection system. It can also be observed that a notable expansion in the precision values' standard deviations. That is because for the testing group with the small percentage of attackers, e.g., 1%, 3%, even if a small proportion of legitimate vehicles is classified as Sybil vehicles, it will result in a disproportionately higher number of false positives than the number of true positives. That also explains the phenomenon that the smaller percentage of the attackers is, the more significantly the f1 score drops when σ is large in Fig. 4. Furthermore, the number of false negatives decreases remarkably, and recall values reach more than 90% when σ is set greater than or equal to 0.1.

VI. Conclusion

In this paper, we present a novel credibility enhanced TGCN based Sybil attack detection system to defend ECSs against Sybil attacks. A TGCN based classifier uses CAVs' reporting information and physical characteristics to make real-time classifications. A local credibility module summaries the local classification results and update the current system's confidence in the CAV's identity. Bayesian inference is then introduced to integrate the accumulative global credibility summarized by the previous ECSs with the current ECS's local credibility to make the final classification. Simulation results show that our proposed Sybil detection system can effectively defend against different quantities of Sybil vehicles with low error rates. As future work, we plan to implement the detection system in an urban traffic network and set up several servers to observe the system's performance, and further improve the design of the system's structure as well as the implementation of different modules.

Acknowledgment

We gratefully acknowledge the support from NSF grants CNS-1839511 and CNS-1834701.

References

[1] S. Raza, S. Wang, M. Ahmed, and M. R. Anwar, "A survey on vehicular edge computing: Architecture, applications, technical issues, and future directions," *Wirel. Commun. Mob. Comput.*, vol. 2019, pp. 3 159 762:1–3 159 762:19, 2019.

[2] S. Liu, L. Liu, J. Tang, B. Yu, Y. Wang, and W. Shi, "Edge computing for autonomous driving: Opportunities and challenges," *Proceedings of the IEEE*, vol. 107, pp. 1697–1716, 2019.

[3] Y. Yao, B. Xiao, G. Wu, X. Liu, Z. Yu, K. Zhang, and X. Zhou, "Voiceprint: A novel sybil attack detection method based on rssi for vanets," *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 591–602, 2017.

[4] S. So, J. Petit, and D. Starobinski, "Physical layer plausibility checks for misbehavior detection in v2x networks," *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, 2019.

[5] J. Yu, "Sybil attack identification for crowdsourced navigation: A self-supervised deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2020.

[6] Y. Hao, J. Tang, and Y. Cheng, "Cooperative sybil attack detection for position based applications in privacy preserved vanets," *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, pp. 1–5, 2011.

[7] T. Zhou, R. Choudhury, P. Ning, and K. Chakrabarty, "P2dap — sybil attacks detection in vehicular ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, pp. 582–594, 2011.

[8] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman, "Sybilguard: defending against sybil attacks via social networks," in *SIGCOMM 2006*, 2006.

[9] A. Mohaisen, N. Hopper, and Y. Kim, "Keep your friends close: Incorporating trust into social network-based sybil defenses," *2011 Proceedings IEEE INFOCOM*, pp. 1943–1951, 2011.

[10] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *ArXiv*, vol. abs/1609.02907, 2017.

[11] Y. Wu, D. Lian, Y. Xu, L. Wu, and E. Chen, "Graph convolutional networks with markov random field reasoning for social spammer detection," in *AAAI*, 2020.

[12] A. Li, Z. Qin, R. Liu, Y. Yang, and D. Li, "Spam review detection with graph convolutional networks," *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019.

[13] K. Cho, B. V. Merrienboer, Çaglar Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *ArXiv*, vol. abs/1406.1078, 2014.

[14] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. WieBner, "Microscopic traffic simulation using sumo," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2575–2582.

[15] Z. Yang, K. Zhang, L. Lei, and K. Zheng, "A novel classifier exploiting mobility behaviors for sybil detection in connected vehicle systems," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2626–2636, 2019.

[16] C. H. O. O. Quevedo, A. M. B. C. Quevedo, G. A. Campos, R. L. Gomes, J. Celestino, and A. Serhrouchni, "An intelligent mechanism for sybil attacks detection in vanets," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[17] P. Gu, R. Khatoun, Y. Begriche, and A. Serhrouchni, "k-nearest neighbours classification based sybil attack detection in vehicular networks," in *2017 Third International Conference on Mobile and Secure Services (MobiSecServ)*, 2017, pp. 1–6.

[18] ——, "Support vector machine (svm) based sybil attack detection in vehicular networks," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, 2017, pp. 1–6.

[19] J. Liang, J. Chen, Y. Zhu, and R. Yu, "A novel intrusion detection system for vehicular ad hoc networks (vanets) based on differences of traffic flow and position," *Applied Soft Computing*, vol. 75, pp. 712 – 727, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494618306835

[20] J. Grover, M. S. Gaur, and V. Laxmi, "A novel defense mechanism against sybil attacks in vanet," in *Proceedings of the 3rd International Conference on Security of Information and Networks*, ser. SIN '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 249–255. [Online]. Available: https://doi.org/10.1145/1854099.1854150

[21] J. Grover, M. S. Gaur, V. Laxmi, and N. K. Prajapati, "A sybil attack detection approach using neighboring vehicles in vanet," in *Proceedings of the 4th International Conference on Security of Information and Networks*, ser. SIN '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 151–158. [Online]. Available: https://doi.org/10.1145/2070425.2070450

[22] Y. Shoukry, S. Mishra, Z. Luo, and S. Diggavi, "Sybil attack resilient traffic networks: A physics-based trust propagation approach," in *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, 2018, pp. 43–54.

[23] K. Zaidi, M. B. Milojevic, V. Rakocevic, A. Nallanathan, and M. Rajarajan, "Host-based intrusion detection for vanets: A statistical approach to rogue node detection," *IEEE Transactions on Vehicular Technology*, vol. 65, pp. 6703–6714, 2016.

[24] A. Pareja, G. Domeniconi, J. J. Chen, T. Ma, T. Suzumura, H. Kanezashi, T. Kaler, and C. E. Leisersen, "Evolvegcn: Evolving graph convolutional networks for dynamic graphs," *ArXiv*, vol. abs/1902.10191, 2020.

[25] H. Gao and S. Ji, "Graph u-nets," in *International Conference on Machine Learning*, 2019, pp. 2083–2092.

[26] L. Ma, S. Yi, and Q. Li, "Efficient service handoff across edge servers via docker container migration," in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, ser. SEC '17. New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: https://doi.org/10.1145/3132211.3134460

[27] K. M. Awan, M. Nadeem, A. S. Sadiq, A. Alghushami, I. Khan, and K. Rabie, "Smart handoff technique for internet of vehicles communication using dynamic edge-backup node," *Electronics*, vol. 9, no. 3, 2020. [Online]. Available: https://www.mdpi.com/2079-9292/9/3/524

[28] J. Pearl, "Chapter 2 - bayesian inference," in *Probabilistic Reasoning in Intelligent Systems*, J. Pearl, Ed. San Francisco (CA): Morgan Kaufmann, 1988, pp. 29 – 75. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780080514895500084

[29] M. Raya, P. Papadimitratos, V. D. Gligor, and J. . Hubaux, "On data-centric trust establishment in ephemeral ad hoc networks," in *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, 2008, pp. 1238–1246.

[30] "How connected vehicles work." [Online]. Available: https://www.transportation.gov/research-and-technology/how-connected-vehicles-work

[31] D. L. Olson and D. Delen, *Advanced Data Mining Techniques*, 1st ed. Springer Publishing Company, Incorporated, 2008.