Hindawi
Security and Communication Networks
Volume 2021, Article ID 1632825, 17 pages
https://doi.org/10.1155/2021/1632825

Research Article

Congestion Attack Detection in Intelligent Traffic Signal System: Combining Empirical and Analytical Methods

Yingxiao Xiang,[1] Wenjia Niu,[1] Endong Tong,[1] Yike Li,[1] Bowei Jia,[1] Yalun Wu,[1] Jiqiang Liu,[1] Liang Chang,[2] and Gang Li[3]

[1]*Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, Beijing, China*
[2]*Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin, China*
[3]*Australia Centre for Cyber Security Research and Innovation, Deakin University, Geelong, Australia*

Correspondence should be addressed to Wenjia Niu; niuwj@bjtu.edu.cn and Endong Tong; edtong@bjtu.edu.cn

Received 11 June 2021; Revised 31 July 2021; Accepted 11 October 2021; Published 31 October 2021

Academic Editor: Jian Weng

The intelligent traffic signal (I-SIG) system aims to perform automatic and optimal signal control based on traffic situation awareness by leveraging connected vehicle (CV) technology. However, the current signal control algorithm is highly vulnerable to CV data spoofing attacks. These vulnerabilities can be exploited to create congestion in an intersection and even trigger a cascade failure in the traffic network. To avoid this issue, timely and accurate congestion attack detection and identification are essential. This work proposes a congestion attack detection approach by combining empirical prediction and analytical verification. First, we collect a range of traffic images that correspond to specific traffic snapshots which are vulnerable to potential data spoofing attacks. Based on these traffic images, an improved generative adversarial network is trained to predict whether a forthcoming attack will cause congestion with a high probability. Meanwhile, we define a group of traffic flow features. After exploring features and conducting a thorough analysis, a TGRU (tree-regularized gated recurrent unit)-based approach is proposed to verify whether congestion occurs. When we find a possible attack that can cause congestion with high probability and subsequent traffic flows also prove congestion, we can say there is a congestion attack. Thus, we can realize timely and accurate congestion attack detection by integrating empirical prediction and analytical verification. Extensive experiments demonstrate that our approach performs well in congestion attack detection accuracy and timeliness.

1. Introduction

Connected vehicle (CV) technology [1, 2] empowers vehicles to communicate with the surrounding environment (roadside units and traffic signal control infrastructure) and is now transforming today's transportation systems. As one key component, the intelligent traffic signal (I-SIG) system [3] is responsible for performing dynamic and optimal signal control. It is based on automatic traffic situation awareness by leveraging the emerging communication infrastructure of the space-air-ground integrated network (SAGIN) [4, 5] with the advantages of coverage, flexibility, and so on. For instance, since September 2016, a series of I-SIG systems have been deployed in California, Florida, and New York by the U.S. Department of Transportation (USDOT) as a CV Pilot Program [1]. These systems are currently under testing and not yet widespread.

Unfortunately, such dramatically increased connectivity also opens a new door for cyberattacks. Recently, such I-SIG has exposed a vulnerability of the controlled optimization of phases (COP) algorithm [6, 7]. Attackers can compromise the on-board units on their vehicles and send malicious messages (such as those containing speed and location) to influence the traffic control decisions at specific times, thus causing unexpected heavy traffic congestion. Some data show that a single attack vehicle can cause a total delay 11 times greater than the total delay before the attack [8], posing a significant barrier to the development and deployment of I-SIG systems on a wide scale in the future.

Previous research [8] reveals such congestion attacks on the COP algorithm, analyzes how congestion attacks affect the COP algorithm decisions, and explains how to launch an attack using data spoofing in SAGIN. However, developers may still lack a deep understanding of such I-SIG attacks and defenses, raising some pressing concerns: (1) What is the effect of different phases where the attack vehicle is located? The different phases of the attack vehicle can cause different congestion effects. (2) What is the quantified correlation between the attack and congestion degree? The quantified correlation refers to the potential relationship between the attack and congestion degree; once identified, we can infer whether the attack occurred according to the congestion degree. (3) Are there any potential features to be utilized for revealing the above correlation? It is necessary to analyze the congestion attack mechanism firstly to solve these issues. The challenges of solving these issues include how to automatically explore multiple and multidimensional features to quantify the traffic flow characteristics under no attack and congestion attack and analyze the correlation between attack features and attack effects. Thus, demystifying the congestion attack based on the COP mechanism through quantified features and exploring new analysis methods will benefit all stakeholders for I-SIG, including transportation, SAGIN, and security specialists.

We demystify the attack and corresponding congestion from a machine learning perspective by exploring and utilizing quantified features. We deeply analyze data spoofing in SAGIN and the COP algorithm vulnerability under two different attack strategies. To explore the effect of different phases of the attack vehicle, we consider utilizing high-level image features and design a novel analysis model based on the cycle generative adversarial network (CycleGAN) [9] to reflect the relation between the attack and the congestion caused by the attack. Thus, we can predict whether a forthcoming attack will cause congestion and the congestion effect according to the traffic image at a specific moment. To explore the quantified correlation between the attack and congestion degree, we utilize traffic flow features and the TGRU classification model [10] (an explainable gated recurrent unit-based model [11] with tree regularization) to verify whether a congestion attack occurs based on all vehicles' trajectory data in an intersection. Following analysis, we also give some promising suggestions for defending I-SIG systems against a congestion attack.

We implement the I-SIG and experiment through visualized simulation in VISSIM [12]. The experiment shows the effectiveness of our approach. We find that feature-based machine learning can reflect the correlation between the attack and congestion degree well. Through the deep learning-based training, the CycleGAN-based approach output visualized results with satisfied prediction compared with real values: the MAE and RMSE of the congestion degree are near 0.02 and 0.03, respectively, and the MAE and RMSE of the congestion degree are near 0.94 and 1.14, respectively. TGRU has a 0.84 precision and 0.79 recall on predicting the spoofing attack based on 30 features. Generally, for defenses, we suggest improving the estimation of vehicle location and speed (EVLS) [7] algorithm of I-SIG if we would like to keep a limited cost, which requires fewer authentication mechanisms and SAGIN reinforcement efforts.

We summarize our contributions as follows:

(1) We perform the study to demystify the attack to I-SIG and the corresponding congestion from a machine learning perspective by exploring different kinds of features through supervised learning and unsupervised learning.

(2) For predicting the spoofing congestion attack, we automatically explore the image feature to quantify the traffic flow characteristics under no attack and congestion attack. And we propose a CycleGAN-based approach to analyze the potential relationship between the congestion attack and corresponding results two stages later based on the image feature.

(3) For verifying the spoofing congestion attack, we propose a TGRU-based approach to explore the underlying relationship between the congestion attack and traffic flow feature at the current moment based on the traffic flow features, which are firstly defined in this work.

(4) We evaluate our approach empirically from the real COP algorithm through VISSIM. We collect 4476 high-quality image samples and 3600 traffic flow data for the experiment, which enables us to demonstrate the effectiveness of our approach compared with ground truth.

2. Preliminaries

2.1. SAGIN Infrastructure of I-SIG. Figure 1 presents the basic architecture for the space-air-ground integrated network of I-SIG, in which two main segments are included: a space segment and a ground segment. The I-SIG of the CV environment is located in the network-based ground segment. There are three main components within the ground segment: on-board units (OBUs), roadside units (RSUs), and signal planning units. These refer to the devices installed in vehicles, roadside servers, and traffic lights, respectively. Both vehicle-to-vehicle (V2V) [13] communication and vehicle-to-infrastructure (V2I, e.g., roadside servers) [14] communication adopt the dedicated short-range communications (DSRC) [15] transmission protocol as 802.11p-based wireless communication; this provides a channel and enables high-speed direct communication. Every vehicle broadcasts anonymously, and surrounding vehicles receive messages. Messages containing critical information are called basic safety messages (BSMs). These contain core data elements, including vehicle size, position, speed, heading, acceleration, and brake system status. Compared with DSRC, the communication from the RSU to the signal planning unit adopts the US National Transportation Communications for Intelligent Transportation System Protocol (NTCIP) [16]. By providing two-way communication between vehicles and traffic signals, NTCIP is specially designed to achieve interpretability and interchangeability between computers and electronic traffic control equipment from different manufacturers, thus increasing use in smart city initiatives.

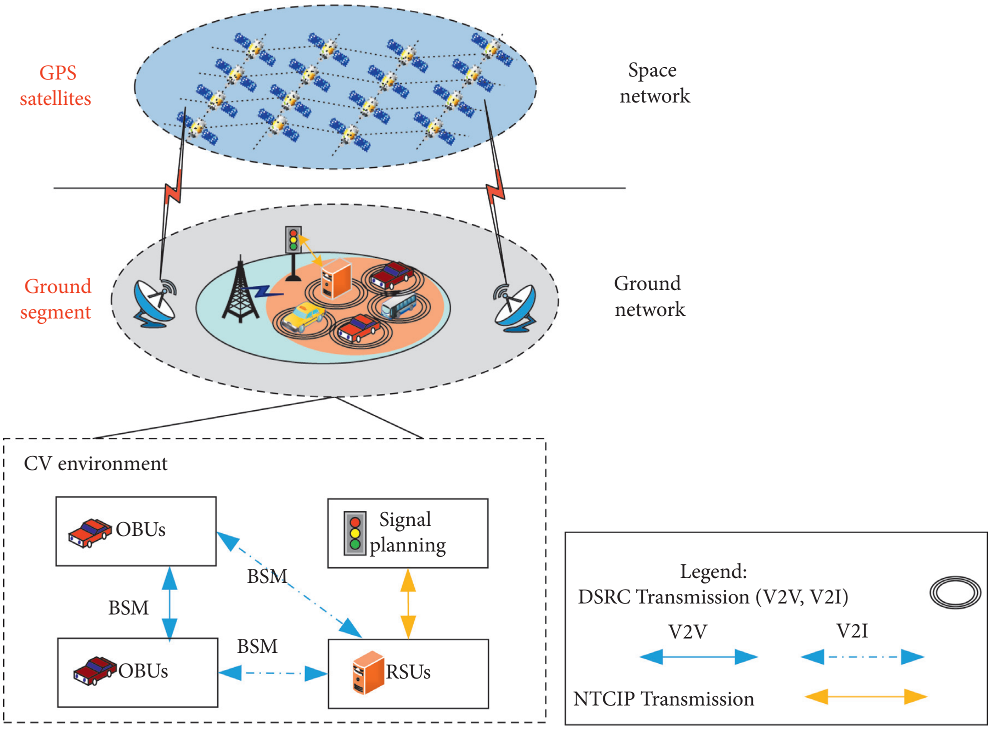

FIGURE 1: The architecture for space-air-ground integrated network of I-SIG.

2.2. I-SIG Data Flow. The data flow of the I-SIG system is revealed in Figure 2. Each OBU of a vehicle sends BSMs to the RSU for real-time trajectory collection. Then, the data are preprocessed to form an arrival table (Table 1) to be used as input for signal planning, which contains COP and EVLS algorithms. If the penetration rate (PR) of OBU for a vehicle is less than 95%, the arrival table will be sent to EVLS for an update. Otherwise, it will be directly sent to the COP algorithm for planning. According to the results of the COP algorithm, a downward signaling command will be transferred to the phase signal controller. After each stage of signal control, the status of the signal will be returned as feedback for continuous COP planning.

There are 8 traffic signals in I-SIG, as shown in Figure 3, called phases; odd numbers are for left-turn lanes; even numbers are for through lanes. Table 1 is the arrival table which is sent to the signal planning model. In Table 1, $T_i = i$ $(0 \le i \le M)$ denotes the time to arrive at the stop bar from the current location. I-SIG sets $M = 130$ seconds, covering a BSM statistic of over two minutes. N_{ij} $(i \in [0, M], j \in [1, 8])$ means that in phase j, there will be N_{ij} vehicles that are going to reach the stop bar within T_i seconds. Here, the stop bar is set in front of the traffic light as it is marked in real road intersections.

The EVLS is based on Wiedemann's car-following model and is used to fill the blank monitoring area of the monitoring segment and insert vehicle data between OBU-equipped vehicles.

The key is to estimate the number of queued vehicles. Because it is assumed that a queue always begins at the stop bar, the last vehicle in the queue needs to be found to determine the queue length.

First, the historical distances to the stop bar and stop time of the last stopped connected vehicle and the second-to-the-last stopped connected vehicle in the queue are calculated; these are denoted as L_{q1}, T_{q1}, L_{q2}, and T_{q2}, respectively. The current time is T_c, and the estimated queue length is L_{es}. Assuming that the queue propagation speed v_q is constant, we have

$$v_q = \frac{L_{q1} - L_{q2}}{T_{q1} - T_{q2}} = \frac{L_{es} - L_{q1}}{T_C - T_{q1}}. \quad (1)$$

Then,

$$L_{es} = L_{q1} + v_q \left(T_C - T_{q1} \right). \quad (2)$$

If the average vehicle length is C, the number N_{0i} of vehicles in queue is then calculated as follows:

$$N_{0i} = \frac{L_{es}}{C}, \qquad i \in [1, 8]. \quad (3)$$

Although such estimation provides effective support for a low PR, it also introduces a new threat of data spoofing attack to the COP algorithm.

3. Demystifying Attack on COP

3.1. Data Spoofing Threat. There are two data spoofing attack strategies proposed in I-SIG (Figure 4). The first one is a direct attack on the arrival table without considering PR; the second one is an indirect attack on EVLS when the PR is less than 95%.

The first strategy is for arrival time and phase spoofing, for both the full deployment period (PR ≥ 95%) and

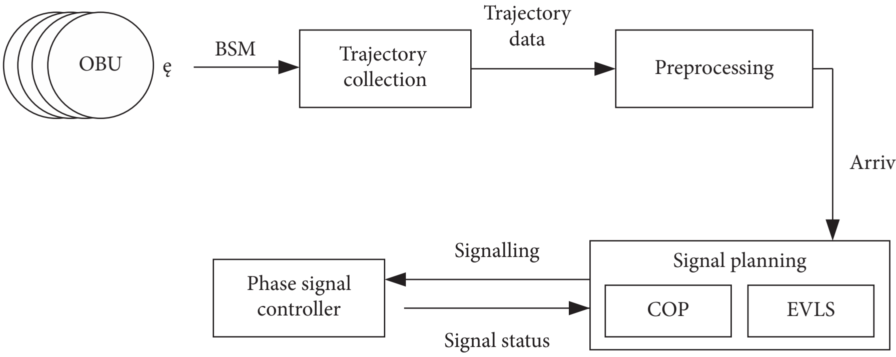

Figure 2: Data flow of the I-SIG system.

Table 1: Arrival table. Numbers 1 to 8 are phases, and T_0 to T_M are the remaining arrival time of vehicles.

Phase	1	2	⋯	8
T_0	N_{01}	N_{02}	⋯	N_{08}
T_1	N_{11}	N_{12}	⋯	N_{18}
T_2	N_{21}	N_{22}	⋯	N_{28}
⋯	⋯	⋯	⋯	⋯
T_M	N_{M1}	N_{M2}	⋯	N_{M8}

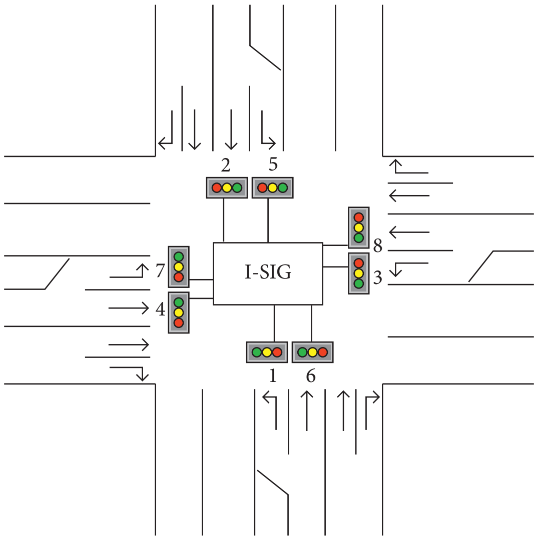

Figure 3: I-SIG signal control scenario, including 8 phases.

transition period (PR < 95%). The attacker changes the location and speed information in vehicle BSMs to alter the vehicle's arrival time and requested phase; thus, the corresponding arrival table elements in Table 1 are changed. This attack strategy can directly attack input data flow no matter what the PR is. As shown in Figure 4(a), the attacker adds a spoofed vehicle into the original vehicle queue at any location. The insertion of a spoofed vehicle makes the queue longer. Moreover, there is an increase in the duration of the green light allocated by the COP algorithm for the current phase, which delays the next start time of the green light of all phases, thus increasing the delay for vehicles to pass through the intersection.

The second strategy is for queue-length spoofing, for the transition period only. This strategy aims to extend the queue length estimated by the EVLS algorithm by changing the location and speed values in BSMs. Figure 4(b) shows that the attacker adds a stopped vehicle with the farthest distance to the stop bar. Owing to the EVLS algorithm estimating the queue length based on the location of the last stopped connected vehicle, this attack causes the estimated queue length L_{es} calculated by equation (2) to increase. Therefore, the number of vehicles in the queue N_{0i} calculated by equation (3) increases as well.

3.2. Planning-Level Congestion Analysis. The COP algorithm is responsible for traffic signal planning; thus, it is essential for planning-level congestion analysis of I-SIG.

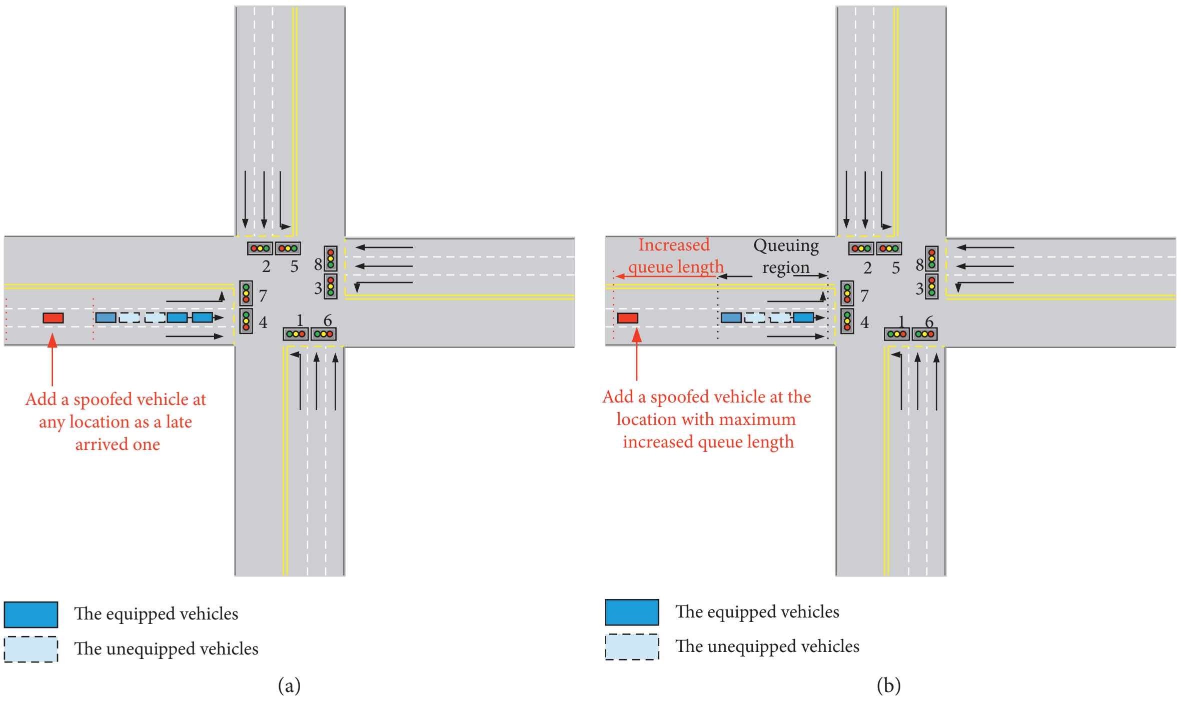

Figure 4: Two strategies of congestion data spoofing attack. PR is short for penetration rate. (a) Direct attack on arrival table without considering PR. (b) Indirect attack on EVLS when PR is less than 95%.

Through reading the published COP-related papers [6, 7] and analyzing the implementation code, we reveal a more complete and detailed COP algorithm for the first time (Algorithms 1 and 2). The authors in [6] first proposed a COP algorithm that allows optimization of various performance indices, including delay, stops, and queue lengths, for the optimal control of a single intersection. However, it did not support flexible or dual ring and phase sequences, and it is difficult to understand for most readers due to the lack of the algorithm flow. Based on the COP algorithm, the authors in [7] presented a real-time adaptive traffic control algorithm by utilizing data from connected vehicles to optimize the phase sequence. However, they did not provide the details of the algorithm. Compared with [6, 7], Algorithm 1 is the first algorithm that provides a complete and detailed flow of signal planning.

In Table 2, we list the meanings of the mathematical symbols that appear in the two algorithms.

The design of the COP algorithm uses the collaboration of two-stage planning and operation. The COP algorithm plans signals for the next-stage based on the vehicle's estimation, and such planned signal duration will be operated at the next-stage signal control time. Thus, this is a continuous alternate process in a fixed phase sequence, which means that the I-SIG system cannot change the order and duration of phases in the current stage since this is set in the previous stage. When bringing foresight of planning, such a design also opens the door to attack signal planning in order to affect next-stage operation continuously.

The spoofing of the arrival table affects the variables $A_{t,k}$ and plan$P_{r,p}$ and the later calculation of Delay$_r$ in line 19 of Algorithms 1. The change in Delay$_r$ causes the variables optV_r in line 21, opt$G_{r,0}$ in line 22, and opt$G_{r,1}$ in line 23 of Algorithms 1 to change as well. Finally, the outputs plan$P_{r,p}$, opt$G_{r,p}$, x_j^*, and v_j are changed.

3.3. *High-Level Image Feature-Based Congestion Attack Prediction.*

In this subsection, we employ an image feature-based CycleGAN to explain the relationship between the phase where the spoofed vehicle is located and the congestion image features two stages later.

As mentioned in the Data Spoofing Threat section, there are two data spoofing attack strategies, but either attack will cause congestion in a period. Different phases of spoofed vehicles lead to different congestion effects. Therefore, the image features of intersection congestion are also different. The CycleGAN model can mine the potential relationship between two different types (X and Y) of images. Through training, CycleGAN can generate the corresponding images Y according to X and generate the related images X according to Y. Therefore, we utilize the CycleGAN model to predict the congestion effects according to the phases of spoofed vehicles, which were considered the attack feature, in order to reveal the relationship between the phase of the spoofed vehicle and the caused congestion image feature.

The CycleGAN architecture is illustrated in Figure 5. One training sample is a pair of image x_i and image y_i to form (x_i, y_i), $\quad x_i \in X$, and $y_i \in Y$. x_i refers to the processed traffic image at the spoofing time, and y_i is the processed traffic congestion image two stages later. The image processing consists of three steps: (1) filter out environment background; (2) extract four images, in which each has 2 phases at one intersection; and (3) join these four images

```
     //The plan of the optimal green duration and phase sequence
     Require: $A_{t,k}$, $G_{\min}^{r,p}$, $G_{\max}^{r,p}$, $R$, T
 (1) Set j = 0, $v_j = 0$
 (2) $X_j^{\min} = \max\{G_{\min}^{0,0} + R + G_{\min}^{0,1} + R,\ G_{\min}^{1,0} + R + G_{\min}^{1,1} + R\}$
 (3) $X_j^{\max} = \min\{G_{\max}^{0,0} + R + G_{\max}^{0,1} + R,\ G_{\max}^{1,0} + R + G_{\max}^{1,1} + R\}$
 (4) $T' = T - s_{j-1} - \cdots - s_1$
 (5) for $r = 0, 1$ do
 (6)     plan $P_{r,0} = 1 + j * 2 + r * 4$
 (7)     plan $P_{r,1} = 2 + j * 2 + r * 4$
 (8) end for
 (9) for $s_j = 1, \ldots, T$ do
(10)     if $s_j \geq X_j^{\min}$ and $s_j \leq \min\{X_j^{\max}, T'\}$ then
(11)         $x_j = s_j$
(12)         effect $G_0 =$ effect $G_1 = s_j - 2R$
(13)         opt$V_0$ = opt$V_1$ = 99999.0
(14)             for $r = 0, 1$ do
(15)                 for $i = G_{\min}^{r,0}, \ldots, G_{\max}^{r,0}$ do
(16)                     $tG_{r,0} = i$
(17)                     $tG_{r,1} = \text{effect}G_r - tG_{r,0}$
(18)                     if $tG_{r,1} \geq G_{\min}^{r,1}$ and $tG_{r,1} \leq G_{\max}^{r,1}$ then
(19)                     $\text{Delay}_r = f(r, \text{plan}P_{r,0}, \text{plan}P_{r,1}, tG_{r,0}, tG_{r,1}, x_j, A_{t,k})$
                             //Calculated by Algorithm 2
(20)                     if $\text{Delay}_r < \text{opt}V_r$ then
(21)                         $\text{opt}V_r = \text{Delay}_r$
(22)                         $\text{opt}G_{r,0} = tG_{r,0}$
(23)                         $\text{opt}G_{r,1} = tG_{r,1}$
(24)                 end if
(25)             end if
(26)         end for
(27) :     end for
(28) else
(29)         $v_j = 99999$
(30)         $x_j = 0$
(31)     end if
(32) end for
(33) $x_j^* = \text{opt}G_{r,0} + R + \text{opt}G_{r,1} + R$
(34) $f_j(x_j^*) = \text{opt}V_0 + \text{opt}V_1$
(35) $v_j = f_j(x_j^*) + v_{j-1}$
     Ensure: $\text{plan}P_{r,p}, \text{opt}G_{r,p}, x_j^*, v_j$
(36) if $j < 2$ then
(37)     $j = j + 1$, go to step 2.
(38) end if
```

Algorithm 1: The COP algorithm

```
    //The delay calculation of ring $r$ at stage j
    // $f(r, \text{plan}P_{r,0}, \text{plan}P_{r,1}, tG_{r,0}, tG_{r,1}, x_j, A_{t,k})$
    Require: $r$, $p1$, $p2$, $g1$, $g2$, $x_j$, $A_{t,k}$
(1) $l_{0,p1} = A_{0,p1}$, $l_{0,p2} = A_{0,p2}$
(2) for $i = 1, \ldots, x_j$ do
(3)     $l_{i,p1} = l_{i-1,p1} - D_{i,p1} + A_{i,p1}$
(4)     $l_{i,p2} = l_{i-1,p2} - D_{i,p2} + A_{i,p2}$
(5)     $d_i = l_{i,p1} + l_{i,p2}$
(6) end for
(7) s.t.
    $D_{i,p1} = \begin{cases} 1, & \text{if } i \leq g1 \text{ and } (i+1)\%2 = 0, \\ 0, & \text{if } g1 < i \leq x_j, \end{cases}$
    $D_{i,p2} = \begin{cases} 1, & \text{if } (g1 + R) < i \leq (g1 + R + g2) \text{ and } (i+1)\%2 = 0, \\ 0, & \text{i } (g1 + R + g2) < i \leq x_j \text{ and } i \leq (g1 + R), \end{cases}$
(8) $\text{Delay}_r = \sum_{i=1}^{x_j} d_i$
    Ensure: $\text{Delay}_r$
```

Algorithm 2: The delay calculation algorithm.

Table 2: Mathematical symbols used in the COP algorithm.

t	Index of arrival time	k	Global phase index
$A_{t,k}$	Element of arrival table denoting the number of vehicle arrivals for phase k at time t	p	Local phase index
r	Ring index in each stage	$G^{r,p}_{\min}$	Minimum green time of phase p in ring r
$G^{r,p}_{\max}$	Maximum green time of phase p in ring r	R	Duration of yellow light and red light
T	Total number of discrete time steps in the planning horizon, in seconds	j	Index of stage
v_j	Value function given state j which represents the accumulated performance measure for the current and all previous stages	$X_j^{\min}$	Minimum possible length of stage j
$X_j^{\max}$	Maximum possible length of stage j	s_j	State variable denoting the total number of time steps allocated to stage j
Plan $P_{r,p}$	Planned phase of phase p in ring r	EffectG_r	Effective total green light time of ring r in stage j
Opt V_r	Optimal delay of ring r in stage j	Opt $G_{r,p}$	Optimal green duration of phase p in ring r
x_j	Length of stage j under the optimal solution	x_j^*	Length of stage j under the optimal solution
$f_j(x_j)$	Performance measure at stage j	$l_{i,k}$	Number of vehicle departing for phase k at time t
$D_{i,k}$	Number of vehicle departing for phase k at time t	Delay$_r$	Delay of ring r at stage j

Table 3: Feature composition schema through selecting equal features from traffic flow head and tail.

	Flow head (10 s)	Flow tail (10 s)
Macrofeatures	CR, α_{CR}, β_{CR}, ICD, α_{ICD}, $\beta_{IC\ D}$	CR, α_{CR}, β_{CR}, ICD, α_{ICD}, β_{ICD}
Microfeatures	PCD$_1$, PCD$_2$, . . ., PCD$_8$, α_{PCD_1}, α_{PCD_2}, . . ., α_{PCD_8}, β_{PCD_1}, β_{PCD_2}, . . ., β_{PCD_8}	PCD$_1$, PCD$_2$, . . ., PCD$_8$, α_{PCD_1}, α_{PCD_2}, . . ., α_{PCD_8}, β_{PCD_1}, β_{PCD_2}, . . ., β_{PCD_8}

from top to bottom to form one sample image according to the phase order of phase (4,7), phase (8,3), phase (2,5), and phase (6,1). Here, the number of phases is consistent with that shown in Figure 3, which is joined by the four parts of an intersection from top to down to form one sample image.

There are four neural networks in the CycleGAN architecture: two generative networks (G and F) and two discriminant networks (D_X and D_Y). The generator G generates a fake image $\widetilde{y}$, which is similar to y given real image x, i.e., G: $X \longrightarrow Y$. Meanwhile, F generates a fake image $\widetilde{x}$, which is similar to x given real image y, i.e., F: $Y \longrightarrow X$. The adversarial discriminator D_X aims to distinguish whether the input image is x and outputs probability $P(x)$. Similarly, D_Y aims to discriminate whether the input image is y and outputs probability $P(y)$.

For $x \in X$, $x \longrightarrow G(x) \longrightarrow F(G(x)) \approx x$ is a cycle, called forward cycle consistency. Similarly, for $y \in Y$, $y \longrightarrow F(y) \longrightarrow G(F(y)) \approx y$ is a cycle called backward cycle consistency. There are two kinds of losses: adversarial loss and cycle consistency loss. Adversarial loss can only guarantee that the samples generated by the generator are distributed with the real samples, but we want the images between the corresponding domains to correspond one by one. That is, X-Y-X can also be migrated back. So, forward cycle consistency and backward cycle consistency are used to make the samples generated by two generators not contradict each other.

Adversarial Loss. This refers to the difference in dataset distribution between generated images and corresponding real images. For discriminators D_{X} and D_{Y}, the closer the output value is to 1, the smaller the loss is.

The losses of G and F can be calculated as L_{G} and L_{F}, respectively, as follows:

$$L_G(G, D_Y, X, Y) = E_{y\sim \text{Pdata}(y)}[logD_Y(y)] + E_{x\sim \text{Pdata}(x)}[\log(1 - D_Y(G(x)))], \tag{4}$$

$$L_F(F, D_X, Y, X) = E_{x\sim \text{Pdata}(x)}[logD_X(x)] + E_{y\sim \text{Pdata}(y)}[\log(1 - D_Y(G(y)))]. \tag{5}$$

Cycle Consistency Loss. This prevents the learned mappings G and F from contradicting each other, making $F(G(x)) \approx x$ and $(F(y)) \approx y$. The loss of $L_{\text{cyc}}(G, F)$ is calculated by the following equation:

$$\begin{aligned} L_{\text{cyc}}(G, F) = E_{x\sim \text{Pdata}(x)}[F(G(x)) - x_1] \\ + E_{y\sim \text{Pdata}(y)}[G(F(y)) - y_1]. \end{aligned} \tag{6}$$

The total loss for CycleGAN is

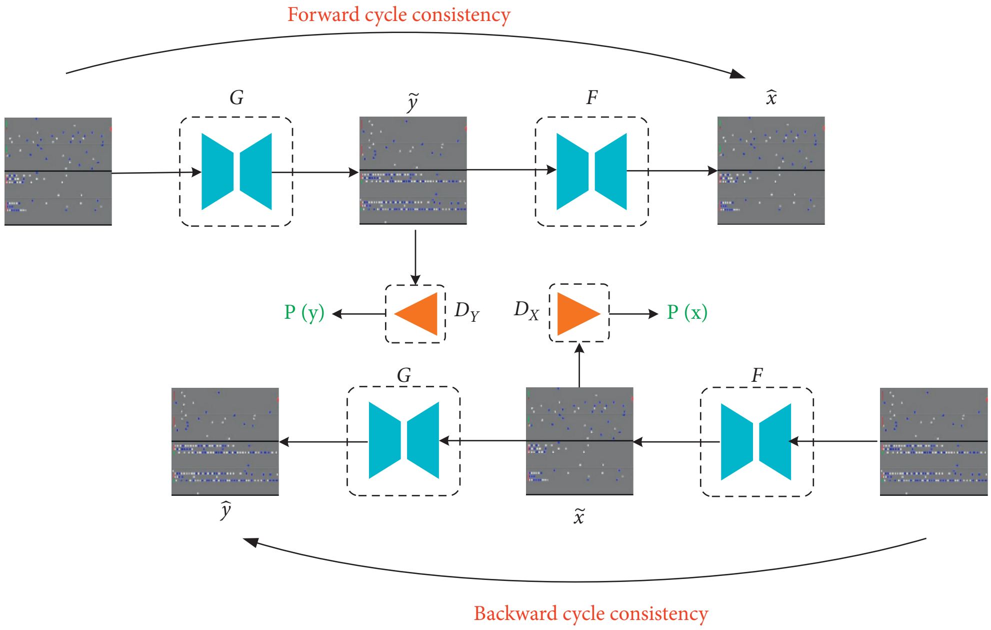

Figure 5: CycleGAN architecture.

$$L(G, F, D_X, D_Y) = L_G(G, D_Y, X, Y) + L_F(F, D_X, Y, X) + \lambda L_{\text{cyc}}(G, F), \tag{7}$$

in which λ is an important parameter. Then, the objective function of the CycleGAN is defined as follows:

$$G^*, F^* = \arg\min_{G,F}\max_{D_X,D_Y} L(G, F, D_X, D_Y). \tag{8}$$

The detailed implementation of neural networks in CycleGAN will be described in the following experiment setup.

3.4. Traffic Flow Feature-Based Congestion Attack Verification. In this subsection, we use a deep learning-based decision tree model, TGRU, to explain the relationship between the traffic flow features and the congestion attack. This relationship can then be used to verify if congestion is occurring. The TGRU model is an interpretable depth time-series model, which is very suitable for intersection traffic flow features with time characteristics. At the same time, interpretability helps to analyze better the relationship between traffic flow features and the congestion attack.

The input of the congestion prediction is the traffic image feature of the intersection, and the prediction model outputs the congestion affects two stages later according to the image feature, which indicates that whether the congestion will occur. However, after the congestion prediction, the verification model is used to verify whether the congestion attack is occurring. The verification input is the defined traffic flow feature that is calculated according to vehicles' information. When we find a possible attack that can cause congestion with high probability and subsequent traffic flows also verify congestion, then we can predict there exists a congestion attack. Thus, we can realize timely and accurate congestion attack detection by integrating empirical prediction and analytical verification.

Feature Definition Based on Traffic Flow. To measure the congestion effects caused by spoofed vehicles, we propose capacity ratio and congestion degree, as well as an attack acceleration and attack amplification ratio based on capacity ratio and congestion degree. We define features as follows:

(1) *Vehicle Capacity Ratio* (CR). $C_k^{\max}$ is the maximum vehicle capacity of each phase, and the vehicle capacity of all 8 phases is computed as $C_{\text{total}}^{\max} = \sum_{k=1}^{8} C_k^{\max}$. Then, the vehicle CR can be denoted by $CR = \sum_{k=1}^{8} N_k / C_{\text{total}}^{\max}$, where N_k is the vehicle number of the kth phase.

(2) *Congestion Degree* (CD). The number of vehicles queuing in the kth phase is denoted as Q_k. Q_{normal} is the number of vehicles during normal queuing and is a constant. Then, the CD of the kth phase can be computed by $PCD_k = Q_k / Q_{\text{normal}}$, and the global CD for an intersection is $ICD = \sum_{k=1}^{8} PCD_k$.

(3) *Attack Acceleration.* Let t_0 be the start time of the data spoofing attack. Then, the accelerations of CR, PCD_k, and ICD at time t are, respectively, calculated by $\alpha_{CR}(t) = (CR(t) - CR(t_0))/(t - t_0)$, $\alpha_{\text{PCD}}(t, k) = (PCD(t, k) - PCD(t_0, k))/(t - t_0)$, and $\alpha_{\text{ICD}}(t) = (ICD(t) - ICD(t_0))/(t - t_0)$.

(4) *Attack Amplification Ratio.* Let t_0 be the start time of the data spoofing attack. Then, the amplification ratio of CR, PCD_k, and ICD at time t is, respectively, calculated by $\beta_{CR}(t) = CR(t)/CR(t_0)$, $\beta_{\text{PCD}}(t, k) = PCD(t, k)/PCD(t_0, k)$, and $\beta_{ICD}(t) = ICD(t)/ICD(t_0)$.

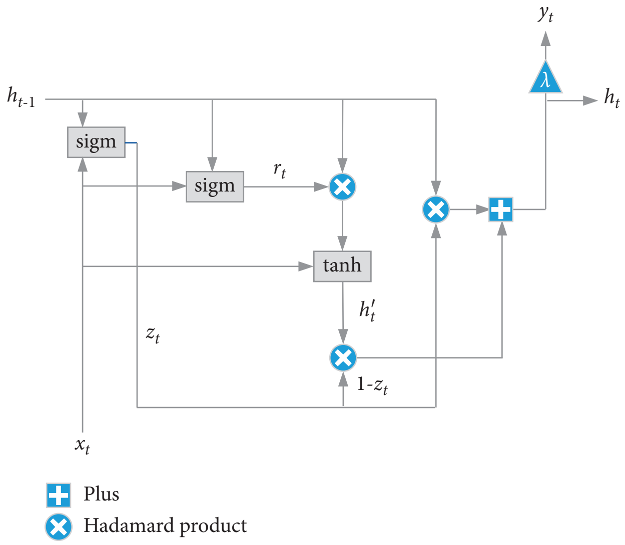

Figure 6: TGRU architecture.

Table 4: Experimental environment configuration.

Platform	Experimental environment	Environmental configuration
COP and VISSIM	Operating system	Windows 10
	CPU	AMD Ryzen5 3550H with Radeon Vega Mobile Gfx 2.10 GHz
	RAM	16G
	Software	PTV VISSIM 4.30, Visual Studio 2019
TGRU and CycleGAN	Operating system	Ubuntu 16.04.6 LTS
	CPU	Intel(R) Core(TM) i7-9700F CPU @ 3.00 GHz
	RAM	32G
	GPU	MSI GeForce RTX 2070 VENTUS
	Graphic memory	151MiB
	Framework	TensorFlow-gpu-1.14.0

Features are divided into macrofeatures and microfeatures for the sake of discussing interpretability, depending on whether they are a feature of the whole intersection or a specific phase (Table 3). Macrofeatures measure the congestion characteristics of the whole intersection, and microfeatures measure the phase of a single signal phase. Unlike the traditional traffic flow characteristics, such as traffic flow, traffic density, and speed, the traffic flow features we defined are related to attacks and are divided into the features for all single signal phases and the features of the whole intersection. For a traffic flow of 1800 seconds, we only sample the first 10 seconds of flow head and the last 10 seconds of flow tail. For flow head, we choose features of macro, micro, or both, and then we choose the same features from the flow tail. Therefore, the number of features is from 20 to 600. We use the Z-score as a standardization to adjust feature values. For values $(x_1, x_2, \ldots, x_n)$ of one feature in all samples, the new value is computed by $x' = x_i - \overline{x}/s$, in which s is the standard deviation and $\overline{x}$ is the mean value of $(x_1, x_2, \ldots, x_n)$.

TGRU Model. We try data spoofing exhaustedly using the last vehicle and collect time-sequence samples. For such data, we use TGRU, a time-series model with decision tree regularization, for interpretability. Figure 6 shows the TGRU architecture for end-to-end calculation.

There are four main calculators: sigm, tanh, plus, and Hadamard product. Sigm refers to the sigmoid function, and tanh refers to the hyperbolic tangent function. The objective function is as follows:

$$\min_{W}\left(\lambda\psi(W)+\sum_{n=1}^{N}\sum_{t=1}^{T}\operatorname{loss}\left(y_{nt},\tilde{y}_{nt}\left(x_n,W\right)\right)\right), \qquad (9)$$

where λ $(\lambda > 0)$ is the regularization strength, W is the whole parameter space, N is the sample number, and T denotes a sampling frequency in one series. The logistic loss function is binary cross entropy.

Next, a single binary decision tree that accurately reproduces the network's thresholded binary predictions $\tilde{y}_n$ given input x_n is found. Then, the complexity of this decision tree as the output of $\Omega(W)$ is measured. The complexity is measured by the average decision path length, i.e., the average number of decision nodes that must be touched to make a prediction for an input example x_n. A regularization function $\widetilde{\Omega}(W)$ is used to map W to an estimate of the

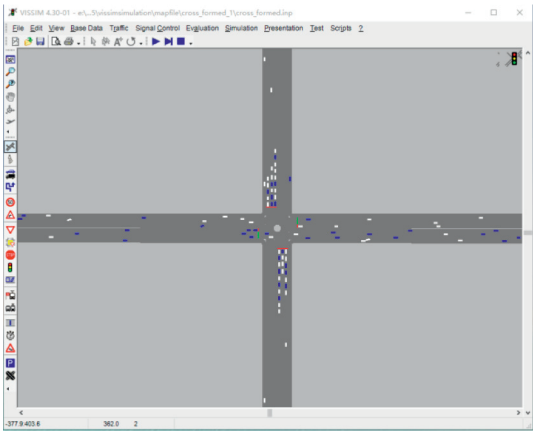

Figure 7: VISSIM simulation environment. The planned results of the COP algorithm and the real-time traffic flow are displayed in VISSIM.

Table 5: Sample datasets for TGRU and CycleGAN.

TGRU	Feature number	32
	Sample number	610
CycleGAN	Image (256×256 pixels) number of X	2238
	Image (256×256 pixels) number of Y	2238

average path length and is implemented by a multilayer perception (MLP) approximator.

Then, tree regularization is conducted, and its objective function is defined as follows:

$$\min_{\xi}\left(\sum_{i=1}^{J}\left(\Omega\left(W_j\right)-\widetilde{\Omega}\left(W_j,\xi\right)\right)^2\right)+\lambda\xi_2^2, \tag{10}$$

where J is the size of the candidate dataset of W and vector ξ denotes the parameters of this chosen MLP approximator.

4. Experiment

4.1. Setup. The platform and experimental environment configuration are shown in Table 4. We use a PC to run the COP algorithm and VISSIM for real-time traffic flow signal control and corresponding traffic simulation. We use another GPU server for both TGRU and CycleGAN training.

VISSIM, the traffic simulation platform, can capture and display the changes of traffic signal and traffic flow planned by the COP algorithm in real-time, as shown in Figure 7. Table 5 shows the sample datasets for TGRU and CycleGAN. In TGRU, we train a 3-layer MLP with 100 first-layer nodes, 100 second-layer nodes, and 10 third-layer nodes. In the CycleGAN, the generator contains encoding, transformation, and decoding. Encoding includes one 7×7 Convolution-InstanceNorm-ReLU layer with stride 1 and two 3×3 Convolution-InstanceNorm-ReLU layers with stride 2. Transformation includes 9 residual blocks for 256×256 images and two 3×3 convolutional layers with the same number of filters on both layers. Finally, decoding includes two 3×3 fractional strided Convolution-InstanceNorm-ReLU layers with stride 2 and one 7×7 Convolution-InstanceNorm-ReLU layer with stride 1. In the discriminator networks, we use 70×70 PatchGANs [17], and the discriminator architecture includes four 4×4 Convolution-InstanceNorm-Leaky-ReLU layers with stride 2. The last layer contains a convolution to produce a 1-dimensional output.

4.2. Congestion Attack Prediction and Visualized Analysis. We evaluate the performance of the CycleGAN model based on image features.

Evaluation Metric. For N samples testing, we further evaluate the CR, PCD, and ICD based on the mean absolute error (MAE) and root mean squared error (RMSE). We have MAE and RMSE of CR expressed as follows:

$$\begin{aligned}\mathrm{MAE}_{CR}&=\frac{1}{N}\sum_{i=1}^{N}\left|CR^i-\widetilde{CR^i}\right|,\\ \mathrm{RMSE}_{CR}&=\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(CR^i-\widetilde{CR^i}\right)^2},\end{aligned} \tag{11}$$

where CR^i is the real value and $\widetilde{CR^i}$ is the estimated value. Similarly, we have $\mathrm{MAE}_{\mathrm{PCD}_k}$, $\mathrm{RMSE}_{\mathrm{PCD}_k}$, $\mathrm{MAE}_{\mathrm{ICD}}$, and $\mathrm{RMSE}_{\mathrm{ICD}}$.

Visualized Results and Quantitative Qnalysis. In Figure 8, the first column is the original image x, the second one is the output image $G(x)$ by CycleGAN, and the third column gives the real image y with congestion. Our approach has a satisfied generator and can predict a future result of congestion attacks to provide a visualization for better human understanding.

Table 6–8 show MAE and RMSE values under different evaluation metrics and test sets. Tables 6 and 7 show the CR

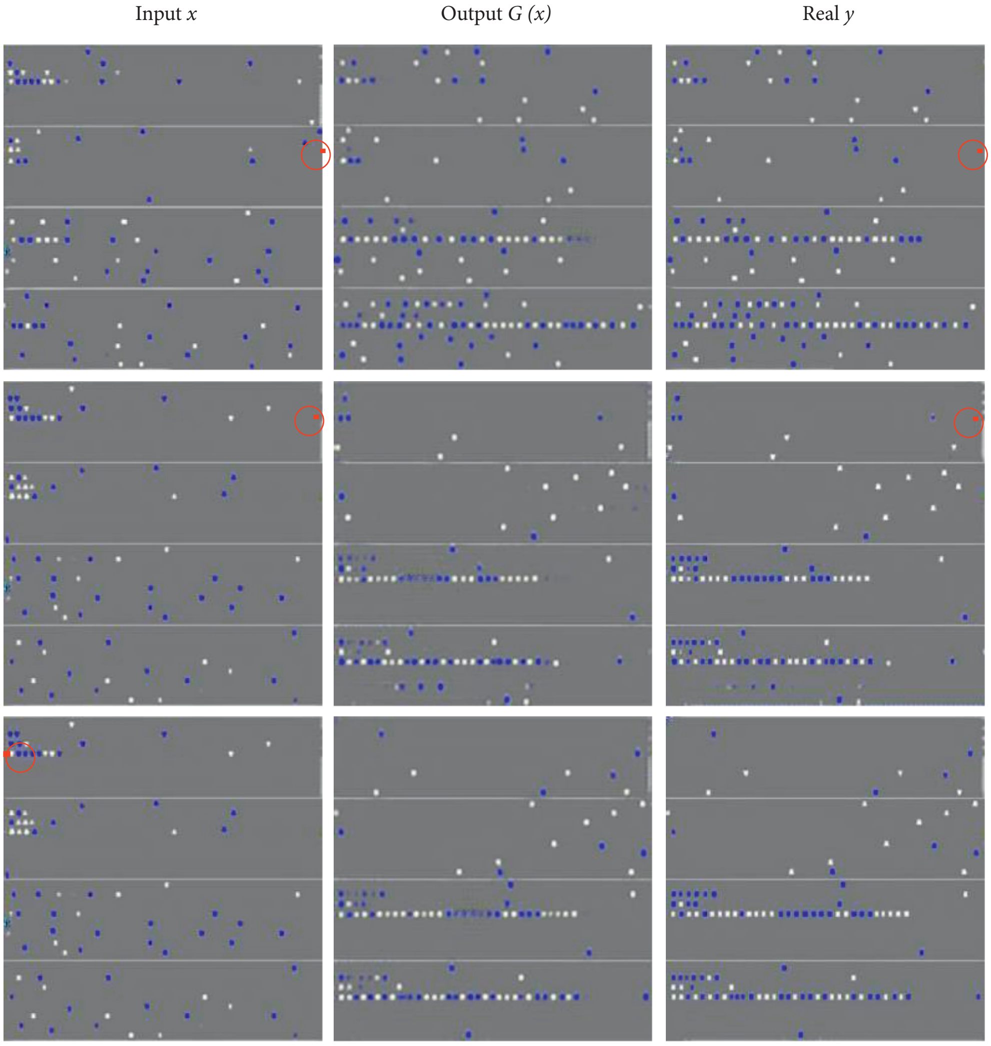

Figure 8: Visualized CycleGAN output compared with real traffic image two stages later based on three different original image inputs at the beginning of spoofing attack. The blue dots represent OBU-equipped vehicles whereas the while dots represent unequipped vehicles.

Table 6: $\text{MAE}_{CR}$and RMSE_{CR} obtained on training set and with 4-fold cross validation or 10-fold cross validation.

	Training set	4-fold cross validation	10-fold cross validation
MAE_{CR}	0.0257	0.0213	**0.0205**
RMSE_{CR}	0.0310	0.0256	**0.0225**

The bold values denote the minimum values of MAE or RMSE on different training sets.

and CD measurements of one intersection and display a satisfying prediction compared with the ground truth. As shown in Table 6, in 10-fold cross validation, our CycleGAN has a pretty good performance in CR prediction. It has very small MAE and RMSE values, 0.0205 and 0.0225, respectively, which is better than that obtained with 4-fold cross validation.

For ICD (Table 7), the 4-fold cross validation results of MAE and RMSE are better than those of 10-fold cross validation, reaching 0.8100 and 0.9987, respectively. We present the detailed values of each phase for MAE and RMSE of congestion degree in Table 8. We can see that through comparing values based on the training set and cross validation, our CycleGAN-based model does not overfit by training. The best results are at $k = 3$, and we have the lowest values of $\text{MAE}_{\text{PCD}_k}$ and $\text{RMSE}_{\text{PCD}_k}$ (0.2250, 0.2050, 0.2050, 0.2617, 0.2519, and 0.2360) compared with the values of other phases. However, the errors at $k = 5$ increase a lot, which is why MAE_{ICD} and RMSE_{ICD} approach 1. This is because the fewer the vehicles, the better the prediction effect of the model. However, the attack vehicle is at phase 3, which has the least number of queues, while the congestion occurs at phase 5, which has the largest number of queues. Therefore, the prediction effect of phase 3 is the best of the 8 phases, so the lowest MAE and RMSE values are $k = 3$, while the prediction errors at phase 5 increased a lot.

We present bar charts for MAE and RMSE of 8-phase congestion degree in Figures 9 and 10, respectively. In

Figure 9, the best average value of MAE is based on 10-fold cross validation (with a value of 0.3844), and the worst average value is based on 4-fold cross validation (with a value of 0.4481). In Figure 10, we have similar results for RMSE; the best and the worst are 0.4471 and 0.5491, respectively. Both average values mean that the CycleGAN is robust and that we have good feature capture in our approach.

In addition, we compare the performance of the CycleGAN model with that of pix2pix [17], another GAN-based model, by quantitatively analyzing experimental results from the whole intersection and the specific phases perspective, respectively. Here, we use the experimental results under 4-fold cross validation. For the measurements of the whole intersection, as shown in Table 9, we have $\mathrm{MAE}_{CR} = 0.0213$, $\mathrm{RMSE}_{CR} = 0.0256$, $\mathrm{MAE}_{\mathrm{ICD}} = 0.8100$, and $\mathrm{RMSE}_{\mathrm{ICD}} = 0.9987$ for CycleGAN and $\mathrm{MAE}_{CR} = 0.1167$, $\mathrm{RMSE}_{CR} = 0.1297$, $\mathrm{MAE}_{\mathrm{ICD}} = 3.7917$, and $\mathrm{RMSE}_{IC\,D} = 3.8500$ for pix2pix. We can see that CycleGAN has lower MAE and RMSE values than pix2pix. Therefore, the CycleGAN model has a better performance than the pix2pix model on the measurements of the whole intersection.

We further compare the model performance for the specific phases. Table 10 shows the detailed MAE and RMSE values of each phase for CycleGAN and pix2pix. There are the lowest MAE and RMSE values at $k = 3$ for both models: 0.2050 and 0.2538 for CycleGAN and 0.2519 and 0.9830 for pix2pix. For all phases, the MAE and RMSE values of CycleGAN are lower than those of pix2pix. Therefore, the CycleGAN model also has a better performance than the pix2pix model on the measurements of all phases.

To sum up, in the CycleGAN-based prediction model, we extract four-direction road images of the intersection and perform phase-based composition for generating a new sample image to quantify the traffic flow characteristics. Based on the image feature, the CycleGAN-based approach analyzes the potential relationship between the congestion attack and the corresponding congestion effect two stages later. Also, the model is used to analyze the congestion effects that different phases of the attack vehicle caused. Meanwhile, we can obtain the visualized results based on the image feature. The experimental results on the CycleGAN-based model and compared experiments with the pix2pix model demonstrated the superiority of the CycleGAN-based model.

4.3. Congestion Attack Verification. Here, we evaluate the performance of the TGRU model based on traffic flow features. We use the confusion matrix, accuracy, AUC value, precision, recall, and F1-score.

The TGRU model is trained to distinguish whether the intersection is under a spoofing attack based on traffic flow features. We collect time-series traffic flow data for 3600 seconds under both the normal state and attack state. We consider 1-second intervals as time steps. Each data vector x_{nt} has 30 features, as defined in Section 3.4. Each outcome y_{nt} is a binary label marking whether the intersection is under a spoofing attack. The sequence length is set to 20 seconds, considering that the maximum green time of each signal is 20 seconds. Hence, 360 samples are obtained in total: 180 samples of which contain 3600 traffic flow data and are used for training and the other 180 samples are used for testing.

We apply the model to the test set and calculate its AUC value, accuracy, precision, recall, and F1-score; these values are reported in Table 11. We see that for different parameter settings, the TGRU model with our defined traffic features can achieve a great prediction quality. The AUC values are all approximately 0.8, and the accuracy values are 0.79, 0.79, and 0.75 when using different parameter settings. Furthermore, the average values of precision, recall, and F1-score are satisfying, almost near 0.8.

Figure 11 shows the three ROC [18] curves of TGRU. Corresponding AUC values are shown as well. We can see that these curves are similar, and their AUC values (0.82, 0.85, and 0.78) are all around 0.8. Moreover, the TGRU model has similar performance with different parameter settings; this indicates that our defined classification features are efficient, and the different parameter settings have little effect on TGRU model's performance.

The decision tree generated by Graphviz [19] is shown in Figure 12. For 3600 traffic flows, this tree has 9 levels. From top to down, according to each feature value, the flow data can be grouped into different classes step by step. For example, when $X\ [13] \leq 0.068$, there are 32 traffic flows of 57 flows correctly predicted as the class of spoofing attack 1; this indicates the importance of the 13th dimension feature, i.e., the congestion degree of the 8th phase PCD_8, in predicting the class of spoofing attack 1.

Also, we compare the TGRU model with a time-series prediction method, seasonal autoregressive integrated moving average (SARIMA). Here, it is detected whether the congestion occurs or not based on traffic flow features. We carry out experiments for different approaches under different traffic flow feature sets. We choose the primary traffic flow data for the first feature set as the traffic flow feature FS_1. The second feature set FS_2 is shown in Table 3. According to the two approaches, we construct two feature sets based on the traffic flow data we collect. As shown in Table 12, the accuracy values of SARIMA and TGRU on the feature set FS_1 are 0.744 and 0.772, respectively, and on the feature set FS_2 are 0.784 and 0.790, respectively, which demonstrates that the TGUR model based on our defined traffic flow features is superior to others.

In conclusion, in the TGRU-based verification model, we propose some timing characteristics, including capacity ratio, congestion degree, attack acceleration, and attack amplification ratio, to measure the congestion effects based on traffic flow. Based on the defined traffic flow features, the TGRU-based model is used to analyze the underlying relationship between the congestion attack and traffic flow features at the current moment. Meanwhile, the decision tree helps better interpret the relationship between traffic flow features and the congestion attack. The experimental results on the TGRU-based model and compared experiments with

Table 7: $\mathrm{MAE}_{\mathrm{ICD}}$ and $\mathrm{RMSE}_{\mathrm{ICD}}$ on training set and with 4-fold cross validation or 10-fold cross validation.

	Training set	4-fold cross validation	10-fold cross validation
$\mathrm{MAE}_{\mathrm{ICD}}$	0.8350	**0.8100**	1.1800
$\mathrm{RMSE}_{\mathrm{ICD}}$	1.0500	**0.9987**	1.3609

The bold values denote the minimum values of MAE or RMSE on different training sets.

Table 8: $\mathrm{MAE}_{\mathrm{PCD}}$ and $\mathrm{RMSE}_{\mathrm{PCD}}$ on training set and with 4-fold cross validation or 10-fold cross validation (k denotes the kth phase).

	$\mathrm{MAE}_{\mathrm{PCD}_k}$			$\mathrm{RMSE}_{\mathrm{PCD}_k}$		
	Training set	4-fold cross validation	10-fold cross validation	Training set	4-fold cross validation	10-fold cross validation
$k = 1$	0.5650	0.4450	0.6575	0.6749	0.5497	0.7884
$k = 2$	0.3850	0.5050	0.3625	0.5074	0.6268	0.4242
$k = 3$	**0.2250**	**0.2050**	**0.2050**	**0.2617**	**0.2519**	**0.2360**
$k = 4$	0.3450	0.3200	0.2250	0.4319	0.3733	0.2639
$k = 5$	**0.8300**	**0.9200**	**0.5925**	**0.9859**	**1.1070**	**0.5720**
$k = 6$	0.3850	0.5350	0.4125	0.5035	0.6535	0.5188
$k = 7$	0.3900	0.3750	0.3600	0.4701	0.4759	0.4602
$k = 8$	0.4300	0.2800	0.2600	0.4990	0.3550	0.3131

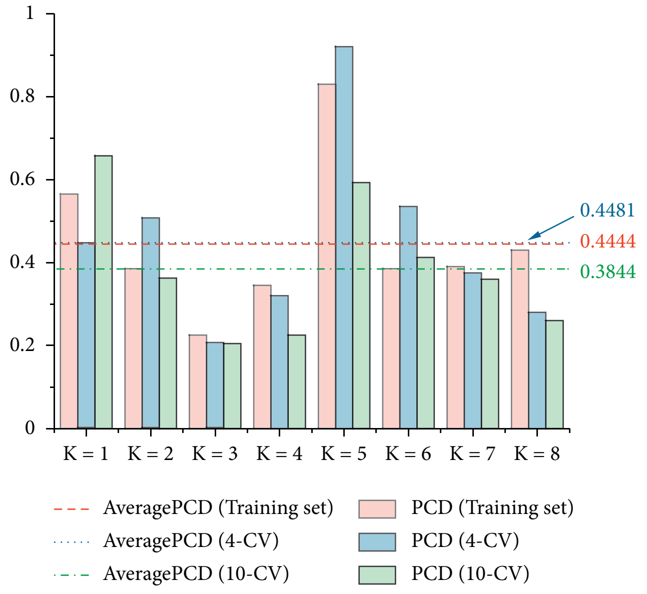

Figure 9: Bar chart of $\mathrm{MAE}_{\mathrm{PCD}_k}$.

the SARIMA model demonstrated the superiority of the TGRU-based approach.

5. Defense Suggestions

To proactively address the congestion attack of the I-SIG system, this section discusses how to defend against the attacks assessed above.

EVLS Improvement for COP Reinforcement. As estimated by the USDOT [20], I-SIG may take 25–30 years to reach a 95% PR for intelligent transportation systems. Thus, for I-SIG under a real low PR, I-SIG needs to adopt an EVLS algorithm to estimate non-OBU-equipped vehicles' location and speed. In the current I-SIG system design, the congestion attack on the COP algorithm utilizes a nonrobust estimation of EVLS. However, it is possible to improve EVLS and thus reinforce the COP algorithm. For single global positioning system (GPS) spoofing, we can introduce more collaboration mechanisms from the transportation field, such as the car-following model. A natural way to accomplish this is to significantly improve queue-length prediction. In the existing EVLS, this could be realized by adding a new software module that interacts with the COP algorithm. Such implementation has a low cost and brings little change to the original COP algorithm.

Another problem is the high impact of PR on security, which we have to change. In the current design, when the PR is smaller, the impact of the attack on the system is more significant because the system cannot accurately obtain the

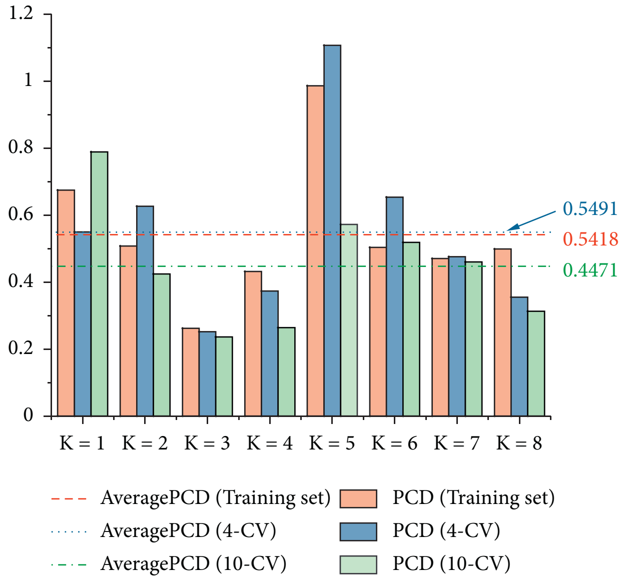

FIGURE 10: Bar chart of $\mathrm{RMSE}_{\mathrm{PCD}_k}$.

TABLE 9: MAE_{CR}, RMSE_{CR}, $\mathrm{MAE}_{\mathrm{ICD}}$, and $\mathrm{RMSE}_{\mathrm{ICD}}$ of CycleGAN and pix2pix with 4-fold cross validation.

	CycleGAN	pix2pix
MAE_{CR}	0.0213	0.1167
RMSE_{CR}	0.0256	0.1297
$\mathrm{MAE}_{\mathrm{ICD}}$	0.8100	3.7917
$\mathrm{RMSE}_{\mathrm{ICD}}$	0.9987	3.8500

TABLE 10: $\mathrm{MAE}_{\mathrm{PCD}}$ and $\mathrm{RMSE}_{\mathrm{PCD}}$ of CycleGAN and pix2pix with 4-fold cross validation (k denotes the kth phase).

	$\mathrm{MAE}_{\mathrm{PCD}_k}$		$\mathrm{RMSE}_{\mathrm{PCD}_k}$	
	CycleGAN	pix2pix	CycleGAN	pix2pix
$k = 1$	0.4450	2.9500	0.5497	3.2383
$k = 2$	0.5050	0.9850	0.6268	1.1697
$k = 3$	**0.2050**	**0.2538**	**0.2519**	**0.9830**
$k = 4$	0.3200	1.7550	0.3733	2.7336
$k = 5$	0.9200	3.3283	1.1070	3.4750
$k = 6$	0.5350	1.1250	0.6535	1.1307
$k = 7$	0.3750	1.9500	0.4759	2.3499
$k = 8$	0.2800	0.3342	0.3550	0.5393

The bold values denote the minimum values of MAE or RMSE when k varies from 1 to 8.

queue length with fewer data. We do not suggest providing two alternative versions of EVLS (i.e., one for high and one for low PR, respectively). Although we analyzed a car-following model in work [21], we believe that a more useful model with a collaboration mechanism should be studied; this will make the estimation of EVLS more accurate as well as COP security more robust.

Authentication and Anomaly Detection. In the current design, authentication is realized through communication between OBU and RSU. However, the attack vehicle might not be a newly joining vehicle or an unauthenticated vehicle; in fact, it can be a normal vehicle with legal authentication. Thus, although authentication reinforcement is not the solution, it can be used to aid in anomaly detection. The idea is that a vehicle cannot appear somewhere suddenly; from the beginning authentication, we should perform analysis on time-series trajectory data to discover any anomaly behavior; this requires a powerful RSU with more computing ability and storing capacity. In addition to an anomaly detection algorithm, implementation needs the support of a collaboration mechanism of multiple I-SIGs; this is a complex global design of intelligent transportation and has not been realized yet. We believe this is critical work that must be accomplished before wide I-SIG deployment.

Prevent Cold-Start Attack. Essentially, the congestion attack is a type of insider attack. Thus, it is challenging to perform anomaly detection for such an attack in a pretty

Table 11: TGRU performance in terms of AUC value, accuracy, precision, recall, and F1-score.

Iteration times	AUC	Accuracy	Classification report			
				Precision	Recall	F1-score
Iters_retrain = 25 Num_iters = 300	0.82	**0.79**	0: normal	0.71	0.98	0.82
			1: attack	0.97	0.59	0.74
			Average	**0.84**	**0.79**	0.78
Iters_retrain = 50 Num_iters = 1000	**0.85**	**0.79**	0: normal	0.72	0.95	0.82
			1: attack	0.93	0.64	0.76
			Average	0.83	**0.79**	**0.79**
Iters_retrain = 100 Num_iters = 3000	0.78	0.75	0: normal	0.69	0.90	0.78
			1: attack	0.86	0.60	0.71
			Average	0.78	0.75	0.75

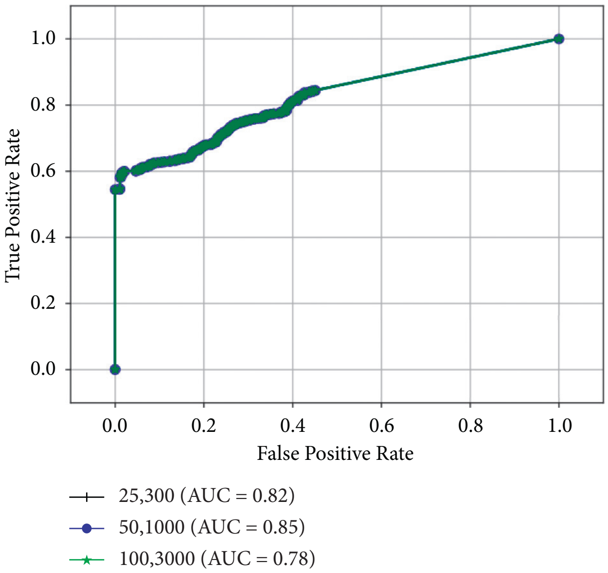

Figure 11: ROC curve of TGRU with AUC values.

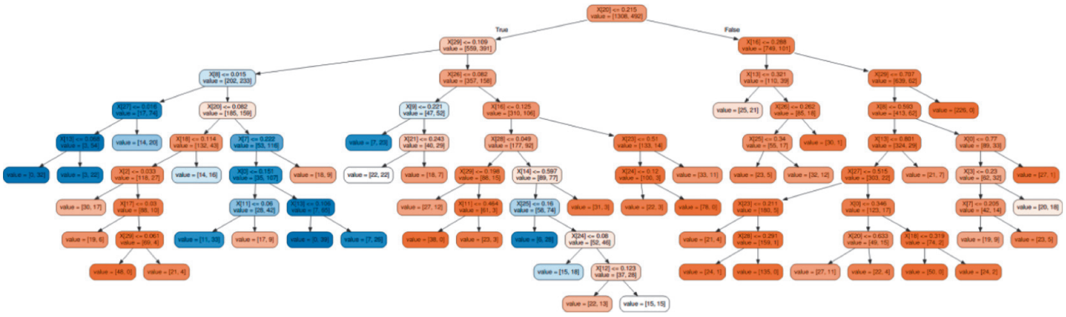

Figure 12: Whole decision tree of 9-level depth.

Table 12: Comparison of different prediction approaches.

Feature set	FS_1	FS_2
SARIMA	0.744	0.784
TGRU	**0.772**	**0.790**

The bold values are the maximum of accuracy when the prediction approach is different.

short time only based on nearby vehicle speed and location information. This means that we cannot avoid the first spoofing data entering the arrival table of the COP algorithm. We suggest that emerging blockchain technology, especially light blockchain, should be considered to rebuild I-SIG or even the whole intelligent transportation system. After that, any data of one node have to be verified by all other nodes. This would result in nearly no chance for spoofed data to be accepted. However, the cost of rebuilding the system is obviously enormous, and more attention should be paid to the light blockchain to test the trade-off between efficiency and security. Regardless, we still believe that this is a promising future for I-SIG security defense.

6. Related Work

Data Spoofing Attacks in SAGIN. The SAGIN has a heterogeneous structure, including vehicle nodes, roadside infrastructure, mobile terminal users, drones, airships, and other stratospheric nodes, as well as high altitude satellite nodes; this brings security challenges [22], such as the various attacks of authenticity, identity, confidentiality, data integrity, and privacy [23]. As a SAGIN-based intelligent transportation system deployed in California, Florida, and New York by the USDOT, I-SIG is exposed to data spoofing attacks [8], which can cause heavy congestion. Such an attack is a position-faking attack of GPS spoofing but is different from a tunnel attack. In a tunnel attack, each vehicle of a Vehicular Ad hoc NETwork (VANET) [24, 25] is equipped with a positioning system (receiver). The attack can be achieved using a transmitter generating localization signals stronger than those generated by the real satellites [26, 27]. The victim could be waiting for a GPS signal after leaving a physical tunnel or a jammed-up area. In comparison, the position spoofing attack to I-SIG refers to an authenticated vehicle only sending the wrong position to affect the COP algorithm, which has lower attack cost and easier implementation. In such an attack, the data spoofing is just one factor, while the mechanism of the COP algorithm is the key factor. Furthermore, for the GPS spoofing attack, our work focuses on algorithm-level security analysis under a spoofing attack.

Congestion Attack Analysis. The previous work [8] reveals the existence of such congestion attacks on the COP algorithm. It analyzes how congestion attacks affect COP decisions and explains how to execute an attack using data spoofing in SAGIN. However, it lacks consideration about the potential features and the quantified correlation between the attack and congestion degree. In comparison, we demystify the attack on I-SIG and corresponding congestion from a machine learning perspective by exploring different kinds of features based on both supervised learning and unsupervised learning. In addition, as the first utilization of both traffic flow features and image features, our work can inspire all stakeholders of I-SIG, including experts of transportation, SAGIN, and security.

7. Conclusions

Toward the spoofing to connected vehicle technology and the SAGIN, a congestion attack has been revealed on the COP algorithm of I-SIG, which performs dynamic and optimal signal control based on automatic traffic situation awareness. Owing to the lack of quantified feature-level analysis, we demystify the attack on I-SIG and the corresponding congestion from both supervised learning and unsupervised learning. We propose a CycleGAN-based approach to analyze the potential relations between the congestion attack and the corresponding results two stages later. We also present a TGRU-based approach to explore the relations between the congestion attack and traffic flow features at a certain moment. In our experiment, we collect high-quality 4476 image samples and 3600 attack-oriented traffic flow data. We then evaluate our approach empirically using the COP algorithm and VISSIM, and our results show the effectiveness of our approach compared with ground truth.

This work is expected to inspire a series of follow-up studies on the security of CV-based I-SIG, but not limited to (1) more machine learning-based approaches, (2) more concrete defense implementation on SAGIN-based I-SIG, and (3) more feature fusion for attack and defense analysis.

Data Availability

All data generated or analyzed during this study are owned by all the authors and will be used to our further research. The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work was supported by the National Natural Science Foundation of China under Grant nos. 61972025, 61802389, 61672092, U1811264, and 61966009, the National Key R&D Program of China under Grant nos. 2020YFB1005604 and 2020YFB2103802, and the Guangxi Key Laboratory of Trusted Software under Grant no. KX201902.

References

[1] U. S. Dot connected vehicle pilot deployment program, https://www.its.dot.gov/pilots/.

[2] Connected Vehicle Applications, https://www.its.dot.gov/pilots/cv_pilot_apps.htm.

[3] Usdot: Multimodal Intelligent Traffic Safety System (Mmitss), https://www.its.dot.gov/research_archives/dma/bundle/mmitss_plan.htm.

[4] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, "Space-air-ground integrated network: a survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2714–2741, 2018.

[5] W. Zhang, L. Li, N. Zhang, T. Han, and S. Wang, "Air-ground integrated mobile edge networks: a survey," *IEEE Access*, vol. 8, pp. 125998–126018, 2020.

[6] S. Sen and K. L. Head, "Controlled optimization of phases at an intersection," *Transportation Science*, vol. 31, no. 1, pp. 5–17, 1997.

[7] Y. Feng, K. L. Z. Head, S. Khoshmagham, and M. Zamanipour, "A real-time adaptive signal control in a

connected vehicle environment," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 460–473, 2015.

[8] Q. A. Chen, Y. Yin, Y. Feng, Z. M. Mao, and H. X. Liu, "Exposing Congestion Attack on Emerging Connected Vehicle Based Traffic Signal Control," in *Proceedings of the Network and Distributed System Security Symposium*, pp. 39.1–39.15, San Diego, CA, USA, February 2018.

[9] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2242–2251, Venice, Italy, October 2017.

[10] M. Wu, M. C. Hughes, S. Parbhoo, M. Zazzi, V. Roth, and F. Doshivelez, "Beyond sparsity: tree regularization of deep models for interpretability," in *Proceedings of the The Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 1670–1678, New Orleans, LI, USA, February 2018.

[11] K. Cho, B. Van Merrienboer, C. Gulcehre et al., "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Empirical Methods in Natural Language Processing*, pp. 1724–1734, Springer, Berlin, Germany, 2014.

[12] Ptv Vissim, http://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim.

[13] A. Krok, "Us department of transportation hopes to mandate v2v communications," 2016, https://www.cnet.com/roadshow/news/us-department-of-transportation-hopes-to-mandate-v2v-communications.

[14] H. Xiong, Z. Tan, R. Zhang, and S. He, "A new dual axle drive optimization control strategy for electric vehicles using vehicle-to-infrastructure communications," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2574–2582, 2020.

[15] J. B. Kenney, "Dedicated short-range communications (dsrc) standards in the United States," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.

[16] R. K. Patel and E. J. Seymour, "The national transportation communication for its protocol (ntcip) for transportation interoperability," in *Proceedings of the Conference on Intelligent Transportation Systems*, pp. 543–548, Boston, MA, USA, November 1997.

[17] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5967–5976, Honolulu, HI, USA, July 2017.

[18] X. Sun and W. Xu, "Fast implementation of DeLong's algorithm for comparing the areas under correlated receiver operating characteristic curves," *IEEE Signal Processing Letters*, vol. 21, no. 11, pp. 1389–1393, 2014.

[19] Graphviz-graph Visualization Software, https://graphviz.org/.

[20] Vehicle-infrastructure integration (vii) initiative: benefit-cost analysis, https://www.pcb.its.dot.gov/connected_vehicle.htm.

[21] X. Gao, J. Liu, Y. Li et al., "Queue length estimation based defence against data poisoning attack for traffic signal control," *IFIP Advances in Information and Communication Technology*, pp. 254–265, 2020.

[22] M. Arshad, Z. Ullah, M. Khalid et al., "Beacon trust management system and fake data detection in vehicular ad-hoc networks," *IET Intelligent Transport Systems*, vol. 13, no. 5, pp. 780–788, 2019.

[23] A. Kapadia, N. Triandopoulos, C. Cornelius, D. Peebles, and D. Kotz, "Anonysense: opportunistic and privacy-preserving context collection," in *Proceedings of the International Conference on Pervasive Computing*, pp. 280–297, New York, NY, USA, May 2018.

[24] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan, "Vehicular ad hoc networks (vanets): status, results, and challenges," *Telecommunication Systems*, vol. 50, no. 4, pp. 217–241, 2012.

[25] X. Zhong, L. Li, Y. Zhang, B. Zhang, W. Zhang, and T. Yang, "Oodt: obstacle aware opportunistic data transmission for cognitive radio ad hoc networks," *IEEE Transactions on Communications*, vol. 68, no. 6, pp. 3654–3666, 2020.

[26] 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*, vol. 20, no. 2, pp. 760–776, 2019.

[27] A. Boualouache, S.-M. Senouci, and S. Moussaoui, "A survey on pseudonym changing strategies for vehicular ad-hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 770–790, 2018.