Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8823300, 14 pages
https://doi.org/10.1155/2020/8823300

Research Article

An Empirical Study on GAN-Based Traffic Congestion Attack Analysis: A Visualized Method

Yike Li,[1] Yingxiao Xiang,[1] Endong Tong,[1,2] Wenjia Niu,[1,2] Bowei Jia,[1] Long Li,[2] Jiqiang Liu,[1] and Zhen Han[1]

[1]Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, Beijing 100044, China
[2]Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China

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

Received 18 September 2020; Revised 20 November 2020; Accepted 30 November 2020; Published 15 December 2020

Academic Editor: Zhili Zhou

With the development of emerging intelligent traffic signal (I-SIG) system, congestion-involved security issues are drawing attentions of researchers and developers on the vulnerability introduced by connected vehicle technology, which empowers vehicles to communicate with the surrounding environment such as road-side infrastructure and traffic control units. A congestion attack to the controlled optimization of phases algorithm (COP) of I-SIG is recently revealed. Unfortunately, such analysis still lacks a timely visualized prediction on later congestion when launching an initial attack. In this paper, we argue that traffic image feature-based learning has available knowledge to reflect the relation between attack and caused congestion and propose a novel analysis framework based on cycle generative adversarial network (CycleGAN). Based on phase order, we first extract four-direction road images of one intersection and perform phase-based composition for generating new sample image of training. We then design a weighted L1 regularization loss that considers both last-vehicle attack and first-vehicle attack, to improve the training of CycleGAN with two generators and two discriminators. Experiments on simulated traffic flow data from VISSIM platform show the effectiveness of our approach.

1. Introduction

With the development of Internet-of-Things (IoT), transportation system is being transformed by various smart sensing devices and connected vehicle (CV) technology [1, 2]. Based on communication and collaboration among vehicle, road-side unit (RSU) [3], and signal system, such intelligent transportation system shows the desirable efficiency and effectiveness of mobility and safety. A typical case is in September 2016; a Pilot Program [1] of CV-based intelligent transportation system was launched by the USDOT (U.S. Department of Transportation) to firstly deploy and test in three states including California, Florida, and New York.

Unfortunately, an algorithm-level attack on controlled optimization of phases- (COP-) based [4, 5] intelligent signal system (I-SIG) [6] is exposed in 2018, in which through data spoofing of vehicle's GPS location and speed, an attacker can compromise the vehicle-side units of a last vehicle existing with quite low attack cost, then mislead the traffic control decisions at proper timing, causing unexpected heavy traffic congestion. This worst result shows that one single attack vehicle is able to cause total congestion of 14 times higher [7]. This is very surprising, since the I-SIG system uses an optimal signal control algorithm COP that can decrease the congestion degree for an intersection. Thus, it is highly important to analyze the traffic congestion attack caused only by one malicious vehicle instead of lots of vehicles, helping to provide effective defenses before wide deployment to the ground.

Although the previous work [7] reveals the existence of congestion attack on COP and analyzes the reason of COP decisions influence, it still lacks detailed guidance about defense even prediction of such attack. Thus, we aim to study the prediction of I-SIG congestion attack in this work. Compared to traditional congestion prediction, the attack-based congestion prediction is totally different, and it is because any classical traffic flow-related theory such as traffic wave distribution does not well fit. Due to timing spoofing attack to the vulnerability of COP, the congestion occurs unexpectedly which seems impossible in normal signal planning, having a nonlinear traffic delay of 200% in short time. There are several urgent questions: (1) What features can be used to characterize the attack? (2) Is there any correlation between the attack and congestion degree? (3) Is congestion degree able to reflect the details of attack consequence? To the best of our knowledge, no similar work focuses on the above questions. Thus, towards the difficulty of feature representation and extraction for quantifying attack, we aim to realize an approach to congestion prediction of attack, based on unsupervised learning from attack image to congestion image, so as to explore new visualized analyzing method to reveal detailed attack results in each phase of intersection. This is our motivation, aiming to benefit all stakeholders for I-SIG, including experts of transportation and security.

In this work, towards I-SIG congestion attack, we are the first to predict the congestion caused by spoofing attack based on adversarial generative network (GAN) [8], an unsupervised learning of machine learning, through directly utilizing high-level image features of traffic, instead of basic features such as location, speed, and delay of vehicle. We firstly perform a phase-based image processing, through background filtering, splitting, and joining operations to form new image which represents a global image of intersection according to certain phase order. We then take such image pairs of initial attack time and congestion time of 30 minutes later as batch training inputs into CycleGAN [9]. We design a weighted L1 regularization loss to learning and distinguish fine differences between the last-vehicle attack and first-vehicle attack. In addition, we also use the trick of early stop to improve CycleGAN performance.

We implement the I-SIG and experiment through visualized simulation in PTV VISSIM [10]. The experiment shows the effectiveness of our approach compared to the pix2pix [11] framework. In condition of fixing 200-epoch training with 0.0002 learning rate, our CycleGAN-based approach output visualized results with satisfied prediction compared to real values: MAE and RMSE of capacity ratio are 0.0267 and 0.0340, respectively, and MAE and RMSE of congestion degree are 1.1250 and 1.5882, respectively. For common use, we suggest to set 200 epochs and 0.0002 learning rate to train as a baseline reference without more tuning efforts. In dynamic training of different epochs from 200 to 500, we find that 200 epochs can effectively prevent the training's mode collapse, and we have the best results when starting a linear learning rate decay at the 150th epoch: MAE and RMSE of capacity ratio are 0.0114 and 0.0134, respectively, and MAE and RMSE of congestion degree are 0.5333 and 0.6245, respectively.

We summarize our contributions as follows:

(i) We perform the first study to predict the congestion caused by spoofing attack based on adversarial generative network (GAN), through directly utilizing high-level image features of traffic. This is a novel visualized approach towards I-SIG congestion attack to reveal the relation between the attack and congestion of 30 minutes later

(ii) We propose a CycleGAN-based prediction approach, in which we design a weighted L1 regularization loss to learning and distinguish fine differences between the last-vehicle attack and first-vehicle attack. Such approach not only enables a prediction from attack to corresponding consequence but also provides an explanation from congestion to the initial traffic of attack phase

(iii) We evaluate our approach empirically from real COP algorithm through VISSIM and collect 4476 image samples of high quality for experiment, which shows the effectiveness of our approach compared to ground truth. We also find that 200 epochs can effectively prevent the training's mode collapse in our approach and have a satisfied performance as a baseline.

The remainder of this paper is organized as follows. Section 2 introduces the backgrounds. In Section 3, we propose our CycleGAN-based prediction approach. Experiments and detailed analysis are reported in Section 4. Section 5 discusses the related works. Finally, we conclude the paper in Section 6.

2. Backgrounds

2.1. Dataflow of I-SIG. The dataflow of the I-SIG system is revealed in Figure 1. Each on-board unit (OBU) [3] of vehicles sends basic safety messages (BSM) [3] to the RSU for a trajectory collection in real time. Then, such data will be preprocessed to form an arrival table as an input to signal planning which has COP and estimation of location and speed (EVLS) [5] modules. If penetration rate (PR) of OBU for vehicles is less than 95%, the arrival table will be sent to EVLS for update. Otherwise, it will be directly sent to COP for planning. According to the results of COP, a downward signalling command will be transferred to the phase signal controller. After each stage of signal control, the next status of signal will be returned as a feedback for continuous COP planning.

As shown in Figure 1, there are 8 phases in the I-SIG environment, and the EVLS fills the blank monitoring area of the monitoring segment on each phase and inserts the estimated vehicle data between equipped vehicles. The key is to estimate the queued vehicles; it is critical to estimate the queue length based on Wiedemann's car following model. Since it is assumed that a queue always begins at the stop bar, the last vehicle in queue needs to be found to determine the queue length. However, while having an effective support in low penetration rate, such estimation also introduces a new threat of data spoofing attack to COP.

2.2. Threat Model. In I-SIG congestion attack, there is a threat model which characterizes the spoofing attack as input, the congestion as output, and studies corresponding causal relation. Based on the attack goal of creating congestion in the intersection, the data spoofing attack has been experimentally proved feasible on CV-based intelligent transportation system. As shown in Figure 1, dataflow of the I-SIG system involves data from both vehicle-side devices (the OBUs) and infrastructure-side device (RSUs and signal controllers). Ghena et al. [12] have pointed out the weakness

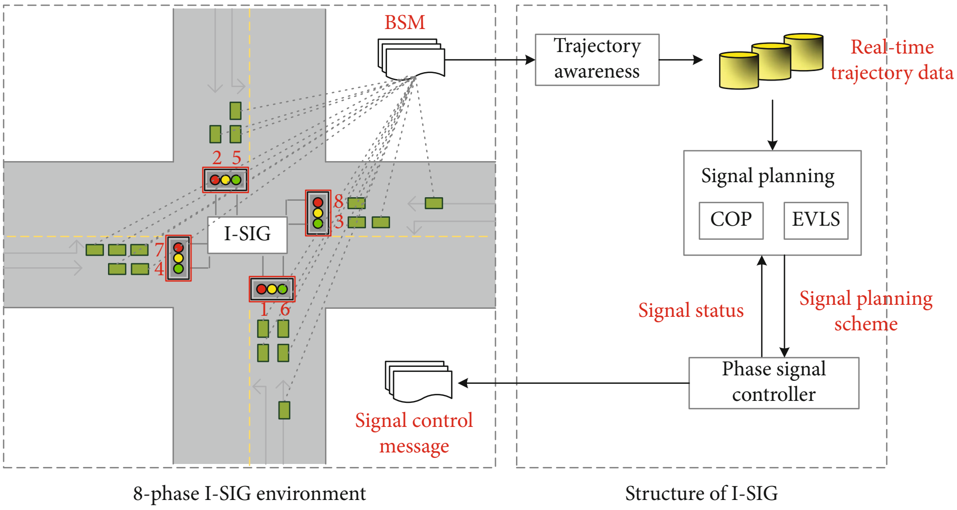


Figure 1: Dataflow of the I-SIG system.

of the infrastructure-side device. In comparison, without considering the weakness of the infrastructure-side device, we aim to realize the attack from vehicle-side devices (the OBUs), in which the attacker sends malicious BSM messages to the OBUs to disrupt signal plan.

More specifically, we focus on single intersection, and the attacker is able to run the ISIG system on a personal computer with a general configuration. Assuming that the attacker has a prior investigation of the system structure and road conditions, after obtaining a set of BSM messages, the attacker can run the I-SIG system to get the prior and subsequent signal planning by COP algorithm. To maximize the realism of the threat model, we mainly explore the effectiveness of attack by a single attack vehicle which is a challenging task as the signal planning of the I-SIG system based on all vehicles in an intersection.

2.3. Congestion Attack on I-SIG. In this paper, two attack strategies of data spoofing have been proposed in I-SIG, one is direct attack on arrival table without considering penetration rate, and the second one is indirect attack on EVLS when penetration rate is less than 95% called the "last-vehicle attack." In the second attack strategy, an attacker adds a spoofing vehicle with speed $v = 0$ at the end of a phase as Figure 2(a) shows. The purpose of this strategy is to extend the queue length estimated by the EVLS algorithm through changing the location and speed values in BSM message. The last-vehicle spoofing can cause the EVLS to have a maximum wrong estimation of queuing length. Such attack further causes an increment of the duration of green light allocated by COP algorithm for the current phase. As a result, it eventually delays the next start time of green light of all the phases and increases the delay for vehicles to pass. As shown in Figure 2(b), the last-vehicle attack causes heavy traffic congestion after just 30 minutes, and the traffic delay has been increased 200%.

Accordingly, we experiment the "first-vehicle attack" as shown in Figure 2(c), in which attacker adds a spoofing vehicle with speed $v > 0$ in front of the original vehicle queue to minimize the queue length. This attack causes the minimum estimating queue length by EVLS and further causes a reduction of the duration of green light allocated by COP algorithm for the current phase and finally increases the delay for vehicles to pass. For all the follow-up phases, it causes early start of the green light. In real simulation on VISSIM, the first-vehicle attack also causes a traffic congestion in the intersection shown in Figure 2(d).

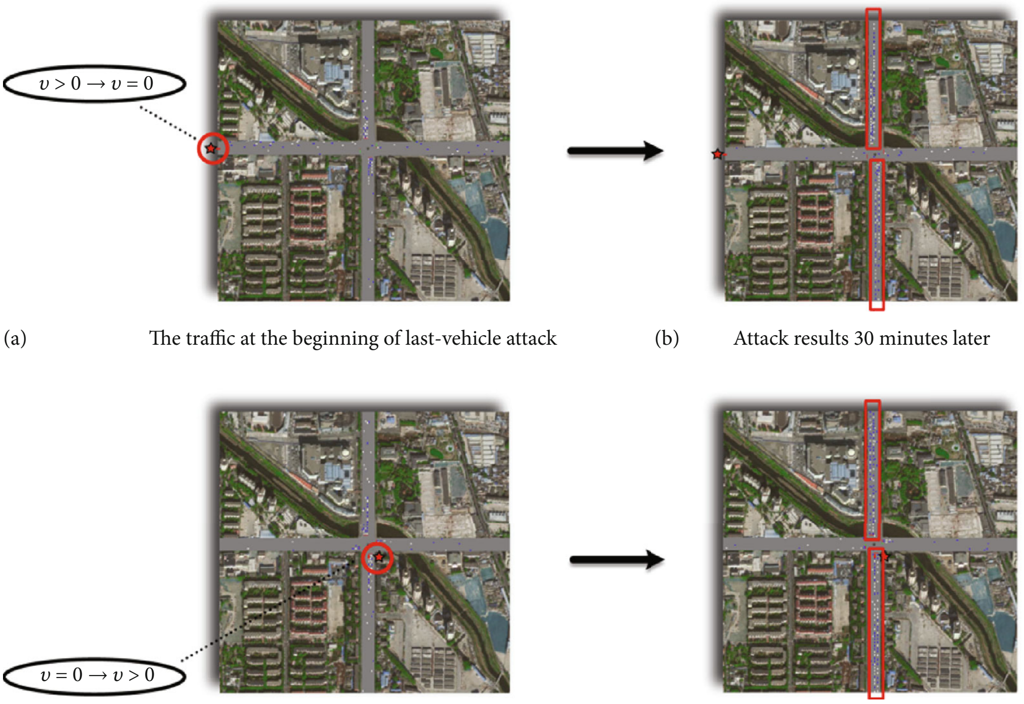
3. CycleGAN Framework Construction

3.1. Sample Image Processing. Figure 3 shows the process to produce samples to form GAN's training dataset, including three main steps: (a) collecting original traffic images from VISSIM; (b) extracting road traffic by background filtering; and (c) forming novel rectangle image of road traffic by split joint. According to the phase order from phase(4,7), phase(8,3), phase(2,5), phase(6,1), joint above four images from top to down to form one sample image.

There are two domains X and Y, $\forall x \in X$ refers to processed original traffic image at the spoofing time, and $\forall y \in Y$ are real congestion traffic images that correspond to domain X 30 minutes later.

3.2. CycleGAN Framework. Figure 4 illustrates the architecture of CycleGAN framework. One training sample is a pair of images x and y to form (x, y), $x \in X$ and $y \in Y$. Here, X and Y denote the source domain and target domain of the framework, x refers to the processed traffic image at the spoofing time, and y is the processed traffic image of congestion 30 minutes later that corresponds to x.

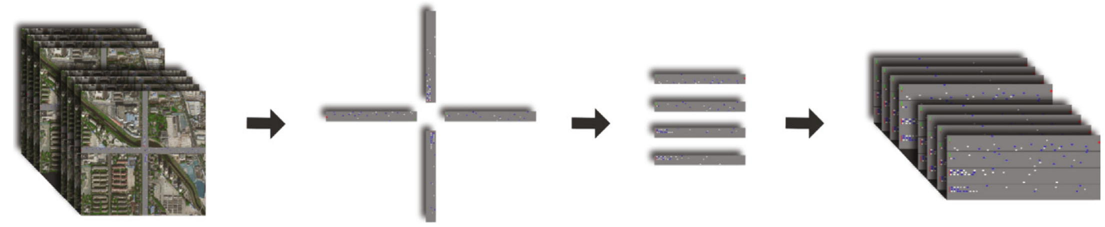
The CycleGAN framework is composed of two generators (G and F) and two discriminators (D_X and D_Y). In the forward direction, the generator G generates fake image $\tilde{Y}$ similar to Y given real image x, i.e., $G : X \longrightarrow Y$. F generates fake image $\tilde{Y}$ similar to X, i.e., $F : Y \longrightarrow X$. The adversarial discriminator D_X aims at distinguishing whether the input image is real and outputs corresponding probability $P(x)$ as a decision.



(a) The traffic at the beginning of last-vehicle attack (b) Attack results 30 minutes later

(c) The traffic at the beginning of first-vehicle attack (d) Attack results 30 minutes later

Figure 2: Illustration of first-vehicle attack and last-vehicle attack.



(a) Collect original traffic images from VISSIM (b) Extract road traffic by background filtering (c) Form novel rectangle image of road traffic by (d) Produce datasets for CycleGAN by split

Figure 3: Process of traffic image preprocessing.

Similarly, D_Y aims at discriminating whether the input image is real and outputs corresponding probability $P(y)$.

The CycleGAN framework has two transform directions to compose a cycle. For $x \in X$, $x \longrightarrow G(x) \longrightarrow F(G(x)) \approx x$ is called forward cycle consistency. Similarly, for $y \in Y$, $y \longrightarrow F(y) \longrightarrow G(F(y)) \approx y$ is called backward cycle consistency. Thus, there are two kinds of losses in the original CycleGAN framework: adversarial loss and cycle-consistency loss.

3.2.1. Adversarial Loss. In the forward direction, based on generator $G : X \longrightarrow Y$ and discriminator D_Y, the adversarial loss can be calculated as follows:

$$\begin{aligned}\mathscr{L}_{\text{GAN}}(G, D_Y, X, Y) = {} & \mathbb{E}_{y\sim p_{\text{data}}(y)}[\log D_Y(y)] \\ & + \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log (1 - D_Y(G(x)))].\end{aligned} \tag{1}$$

$D_Y(y)$ is responsible for determining the probability of y's belonging to real Y, and the generator G is used to generate fake image close to the real one. Thus, we have the objective $\min_{G} \max_{D_Y} \mathscr{L}_{\text{GAN}}(G, D_Y, X, Y)$ to train generator G and discriminator D_Y.

Similarly, for the backward direction, we have loss and corresponding objective function as following:

$$\begin{aligned}\mathscr{L}_{\text{GAN}}(F, D_X, Y, X) = {} & \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D_X(x)] \\ & + \mathbb{E}_{y\sim p_{\text{data}}(y)}[\log (1 - D_X(F(y)))], \\ F^*, {D_X}^* = {} & \min_{F} \max_{D_X} \mathscr{L}_{\text{GAN}}(F, D_X, Y, X),\end{aligned} \tag{2}$$

where $D_X(x)$ is responsible for determining the probability of x's belonging to real X, and the generator F is used to generate fake image close to the real one.

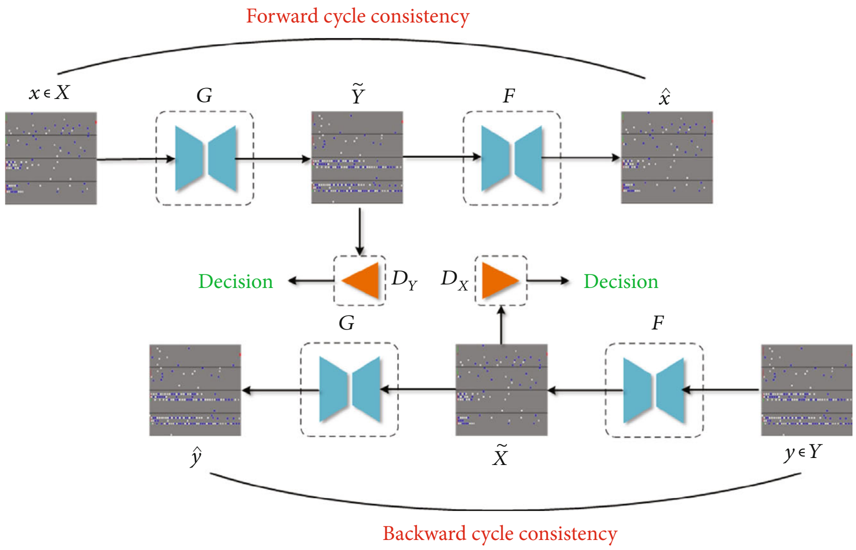


Figure 4: CycleGAN Framework.

The complete adversarial loss function is defined as

$$\mathscr{L}_{\mathrm{GAN}}(G, F, D_X, D_Y) = \mathscr{L}_{\mathrm{GAN}}(G, D_Y, X, Y) + \mathscr{L}_{\mathrm{GAN}}(F, D_X, Y, X). \tag{3}$$

3.2.2. Cycle-Consistency Loss. Cycle-consistency loss is designed to push G and F to be consistent with each other, denoted as $F(G(x)) \approx x$ and $G(F(y)) \approx y$. The cycle-consistency loss can be calculated as follows:

$$\mathscr{L}_{\mathrm{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\mathrm{data}}(x)}\left[\|F(G(x)) - x\|_1\right], \tag{4}$$

in which $\|\bullet\|_1$ is the 1-norm calculation.

3.2.3. Weighted L1 Regularization Loss. For samples of the first-vehicle attack and the last-vehicle attack, we divide processed datasets X, Y into two parts that can be denoted as $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2\}$, $Y = \{\boldsymbol{y}_1, \boldsymbol{y}_2\}$. For $\forall a, b \in \boldsymbol{x}_1$ and $\forall c \in \boldsymbol{x}_2$, the goal of generator G and F is to minimize the difference between a and b as well as to maximize the difference between a and c. The object can be denoted as arg $\min_{F,G} \mathscr{L}_s(F, G) \max_{F,G} \mathscr{L}_d(F, G)$. Weighted L1 regularization loss can be calculated as follows:

$$\begin{aligned}\mathscr{L}_{\mathrm{sep}}(G, F) &= \alpha \mathscr{L}_s(F, G) - \beta \mathscr{L}_d(F, G),\\ \mathscr{L}_s(F, G) &= \mathbb{E}_{a,b \sim p_{\mathrm{data}}(\boldsymbol{x}_1)}\left[\|F(G(a)) - F(G(b))\|_1\right],\\ \mathscr{L}_d(F, G) &= \mathbb{E}_{\substack{a \sim p_{\mathrm{data}}(\boldsymbol{x}_1)\\ c \sim p_{\mathrm{data}}(\boldsymbol{x}_2)}}\left[\|F(G(a)) - F(G(c))\|_1\right],\end{aligned} \tag{5}$$

in which $\mathscr{L}_s(F, G)$ reflects the image difference of same attack type and $\mathscr{L}_d(F, G)$ reflects the image difference of different attack type. α and β are weights.

The whole objective of our CycleGAN framework is defined as follows:

$$\begin{aligned}\mathscr{L}(G, F, D_X, D_Y) = {} & \mathscr{L}_{\mathrm{GAN}}(G, F, D_X, D_Y) + \lambda \mathscr{L}_{\mathrm{cyc}}(G, F)\\ & + \mu \mathscr{L}_{\mathrm{sep}}(G, F),\end{aligned} \tag{6}$$

where λ and μ are parameters, which control the relative importance of different objectives, $\lambda \geq 1$ and $\mu \in (0, 1]$.

The optimal G^*, F^* can be achieved as follows.

$$G^*, F^* = \arg \min_{G,F} \max_{D_x,D_Y} \mathscr{L}(G, F, D_X, D_Y). \tag{7}$$

3.3. Build Generator and Discriminator. Figure 5 illustrates the architecture of structures of generator and discriminator. The two generators G, F share the same structure. Specifically, a generator network contains encoder, transformer, and decoder. The encoder network includes one 7 × 7 Convolution-InstanceNorm-ReLU layer and two 3 × 3 Convolution-InstanceNorm-ReLU layers. Transformer network has 9 residual blocks for 256 × 256 images that contains two 3 × 3 convolutional layers. Decoder network consists of two 3 × 3 fractional-strided-Convolution-InstanceNorm-ReLU layers and one 7 × 7 Convolution-InstanceNorm-ReLU layer.

The two discriminators D_X, D_Y have the same structure. The discriminator networks use the architecture of 70 × 70 PatchGANs [11], and the discriminator architecture includes four 4 × 4 Convolution-InstanceNorm-LeakyReLU, which transforms the input image into a set of feature maps and finally outputs a 1-dimension decision.

3.4. Training Process. There are two training directions in CycleGAN framework; Adam (Adaptive Moment Estimation) [13] is chosen as the optimizer of training. It is an adaptive optimization method that dynamically updates network

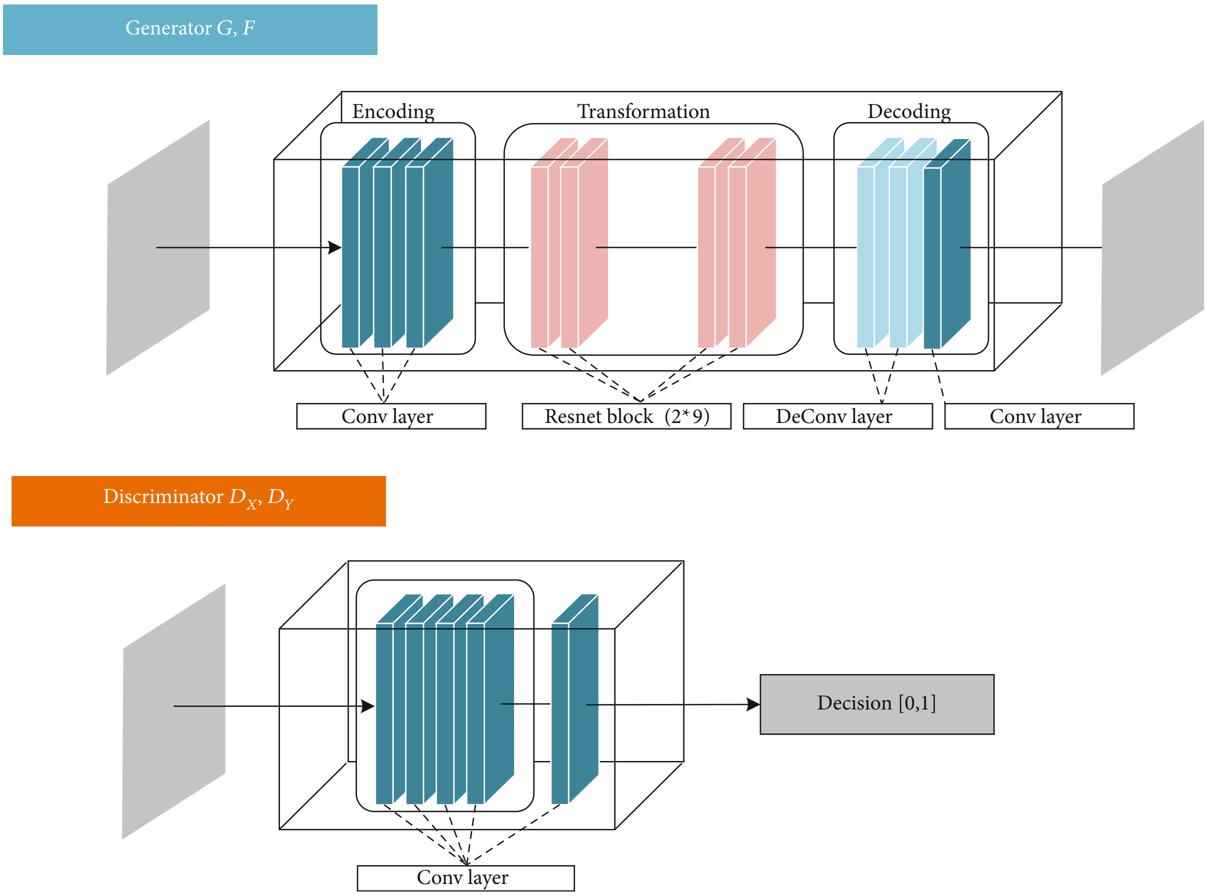


Figure 5: The structures of generator and discriminator.

```
**Input**: original state image set $X_0$, target congestion image set $Y_0$.
**Output**: trained model with optimized parameters.
**Initialization**. initialize network parameters $\theta_g$, $\theta_f$, $\theta_{d_x}$, $\theta_{d_y}$, learning rate $\alpha = 0.002$, $\lambda = 10$, $\mu = 1$, number of training iterations $M$.
while $\theta_g$ and $\theta_f$ has not converged do
  for $t = 1$ to $M$ do
    //Forward cycle
    Generate fake image $G(x)$ and recommend image $F(G(x))$.
    Calculate $G, F$ loss.
    Update the gradient of $G, F$: $\theta_g \longleftarrow \theta_g - \alpha\nabla_{\theta_g}\mathscr{L}(G, F, D_X, D_Y)$
                                   $\theta_f \longleftarrow \theta_f - \alpha\nabla_{\theta_f}\mathscr{L}(G, F, D_X, D_Y)$
    Discriminate fake image $D_Y(G(x))$ and real image $D_Y(x)$.
    Calculate $D_Y$ loss.
    Update the gradient of $D_Y$: $\theta_{d_y} \longleftarrow \theta_{d_y} - \alpha\nabla_{\theta_{d_y}}\mathscr{L}(G, F, D_X, D_Y)$
    //Backward cycle
    Generate fake image $F(y)$ and recommend image $G(F(y))$.
    Calculate $G, F$ loss.
    Update the gradient of $G, F$: $\theta_g \longleftarrow \theta_g - \alpha\nabla_{\theta_g}\mathscr{L}(G, F, D_X, D_Y)$
                                   $\theta_f \longleftarrow \theta_f - \alpha\nabla_{\theta_f}\mathscr{L}(G, F, D_X, D_Y)$
    Discriminate fake image $D_X(F(y))$ and real image $D_X(y)$.
    Calculate $D_X$ loss.
    Update the gradient of $D_X$: $\theta_{d_x} \longleftarrow \theta_{d_x} - \alpha\nabla_{\theta_{d_x}}\mathscr{L}(G, F, D_X, D_Y)$.
  end for
end while
```

Algorithm 1: Iterative training of CycleGAN.

Table 1: Experimental platform and configuration.

Platform	Experimental configuration
VISSIM	Operating system: Windows 10 CPU: AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx 2.10 GHz RAM: 16 G Version: PTV VISSIM 4.30 Interface: VISSIM Component Object Model (COM)
CycleGAN	Operating system: Ubuntu 16.04.6 LTS CPU: Intel(R) Core(TM) i7-9700F CPU @ 3.00 GHz RAM: 32 G GPU: MSI GeForce RTX 2070 VENTUS Graphic memory: 151 MiB Framework: TensorFlow_gpu-1.14.0

weights, which have better convergence performance. The training process is illustrated in Algorithm 1.

4. Experiment

4.1. Setup. We run the I-SIG System and VISSIM simulations to get the original image datasets X_0, Y_0. The platform and experimental environment configuration are shown in Table 1.

4.2. Datasets and Initial Network. Both of the training and test datasets are composed of two parts: the processed traffic image dataset X at the spoofing time and corresponding congestion image dataset Y. Table 2 shows the sample datasets for training and test. The size of all the image is 256 × 256 pixels.

In addition to CycleGAN parameters described in above section, we also set up a comparable GAN model named pix2pix [11], and its parameters are described as Table 3.

5. Evaluation

Actually, our method can be directly compared to NDSS2018's work for the same I-SIG system. In addition, there are also some similar work to discuss. Reporting road traffic congestion can be challenging as there is no standard way of measurement fit for each specific occasion. A series of methods have been proposed to evaluate traffic congestion. Lu and Cao [14], proposed a method based in which level of congestion is considered a continuous variable from free flow to traffic jam, since the source domain and the target domain of our visualized prediction method are both composed of traffic images, and it is hard to extract the high-level image features using traditional text features such as location, speed, and delay of vehicles. Pongpaibool et al. [15] proposed a method based on deep network using image processing technology to deal with the whole image. In comparison, we aim to explore the effectiveness of different attack strategies which need an accurate analyze on each phase instead of the whole region; the former traditional methods are not suitable. Thus, we propose a phase-based evaluation method

Table 2: Datasets of training and testing.

	Name	Sample size
Training dataset	X	Traffic images of initial spoofing First vehicle: 1119 Last vehicle: 1119
	Y	Congestion images 30 minutes later First vehicle: 1119 Last vehicle: 1119
Test dataset	X	Traffic images of initial spoofing First vehicle: 500 Last vehicle: 500
	Y	Congestion images 30 minutes later First vehicle: 500 Last vehicle: 500

Table 3: Experimental parameter settings.

	Parameters	Value
CycleGAN framework	λ	10.0
	μ	1.0
	Initial learning rate	0.0002
	Optimizer	Adam
	Batch size	1
	Dropout rate	No dropout
	Net D	Basic
	Net G	Resnet_9blocks
pix2pix framework	Weight for L_1 loss	100.0
	Initial learning rate	0.0002
	Optimizer	Minibatch SGD, Adam
	Batch size	1
	Dropout rate	0.5
	Net D	Basic
	Net G	U-Net

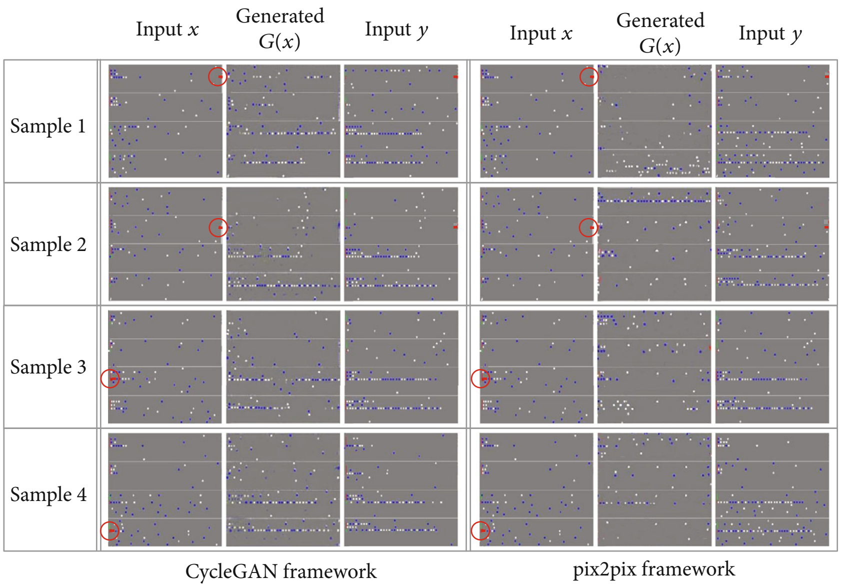


Figure 6: Visualized CycleGAN and pix2pix output compared to the ground truth.

Table 4: MAE_{CR}, RMSE_{CR}, MAE_{ICD}, and RMSE_{ICD} of CycleGAN and pix2pix under different epochs.

Epoch	50		100		200		500		1000	
Framework	CycleGAN	pix2pix	CycleGAN	pix2pix	CycleGAN	pix2pix	CycleGAN	pix2pix	CycleGAN	pix2pix
MAE_{CR}	0.1461	0.1638	0.0317	0.1607	0.0267	0.0929	0.0190	0.1070	0.0137	0.0862
RMSE_{CR}	0.1470	0.1642	0.0342	0.1674	0.0340	0.0985	0.0250	0.1160	0.0177	0.0870
MAE_{ICD}	5.5000	6.5500	2.4167	5.8000	1.1250	2.3250	0.7800	6.7000	0.7000	1.4000
RMSE_{ICD}	5.5675	6.5799	2.6176	6.0027	1.5882	2.7051	0.9623	6.7134	0.9930	1.4509

to quantitatively analyze the congestion results. We first define the evaluation metrics, and we further evaluate them based on the mean absolute error (MAE) and root mean squared error (RMSE), respectively.

5.1. Evaluation Metric

(1) *Vehicle capacity ratio* (CR). $C_k^{\max}$ is the maximum vehicle capacity of each phase, in which k denotes the kth phase, and $C_k^{\max}$ is a constant. For a 300-meter-long road in any phase, the maximum vehicle capacity is 75 assuming that the average vehicle length is 3 meters. $C_{\text{total}}^{\max}$ is used to compute the vehicle capacity of all 8 phases; it can be denoted as $C_{\text{total}}^{\max} = \sum_{k=1}^{8} C_k^{\max}$, and it is also a constant with value 600. For total vehicles of all 8 phases at an intersection, the vehicle capacity ratio can be calculated as follows.

$$\text{CR} = \frac{\sum_{k=1}^{8} N_k}{C_{\text{total}}^{\max}}, \quad (8)$$

in which the N_k is the vehicle number of the kth phase

(2) *Phase congestion degree* (PCD). PCD reflects the ratio of queuing length to normal queuing length. For the kth phase, its PCD_k can be calculated by

$$\text{PCD}_k = \frac{Q_k}{Q_{\text{normal}}}, \quad (9)$$

where the Q_k is the vehicle number of queuing and Q_{normal} is a constant that we set $Q_{\text{normal}} = 10$

(3) *Intersection congestion degree* (ICD). ICD reflects the global congestion degree for an intersection, and it can be calculated by

$$\text{ICD} = \sum_{k=1}^{8} \text{PCD}_k. \quad (10)$$

For N samples testing, we will further evaluate the CR, PCD, and ICD from a statistical view based on the mean absolute error (MAE) and root mean squared error (RMSE),

Table 5: MAE_{CR}, $RMSE_{CR}$, MAE_{ICD}, and $RMSE_{ICD}$ of CycleGAN and pix2pix under different learning rates (LR).

Epoch number (LR = 0.0002) /epoch number (linear decay)	50/50		50/150		100/100		150/50		400/100		250/250		100/400	
Framework	CycleGAN	pix2pix	CycleGAN	pix2pix	CycleGAN	pix2pix	CycleGAN	pix2pix	CycleGAN	pix2pix	CycleGAN	pix2pix	CycleGAN	pix2pix
MAE_{CR}	0.0317	0.1607	0.0114	0.1003	0.0267	0.0929	0.0267	0.1117	0.0113	0.0970	0.0190	0.1070	0.0480	0.1043
$RMSE_{CR}$	0.0342	0.1674	0.0134	0.1017	0.0340	0.0985	0.0297	0.1118	0.0147	0.1037	0.0250	0.1160	0.0497	0.1126
MAE_{ICD}	2.7000	5.8000	0.5333	3.3400	1.1250	2.4750	0.8000	3.7800	0.5600	6.7200	0.8600	6.5200	1.4400	6.5200
$RMSE_{ICD}$	2.7917	6.0027	0.6245	3.4822	1.5882	2.9441	1.0677	3.8499	0.7072	6.7382	1.0536	6.5437	1.5849	6.5403

respectively. We have MAE and RMSE of CR as follows:

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

in which 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 $\mathrm{RMSE}_{\mathrm{ICD}}$.

5.2. Visualization Results. Figure 6 shows congestion traffic images generated by CycleGAN and pix2pix, respectively. The first column is the original image x, the second column is the generated congestion image $G(x)$, and the real congestion image y is given in the third column. The comparison of generated $G(x)$ indicates that our approach has a more satisfied generator of training than pix2pix, having a higher accuracy compared to the ground truth.

5.3. Quantitative Analysis. We quantitatively analyze the performance of CycleGAN and pix2pix. Tables 4–6 show the MAE and RMSE values of CycleGAN and pix2pix under different settings of epoch and learning rate.

As Table 4 shows, when epoch = 1000, CycleGAN and pix2pix both have the best performance of CR and ICD prediction that has very small MAE and RMSE values: for CycleGAN, we have $\mathrm{MAE}_{\mathrm{CR}} = 0.0137$, $\mathrm{RMSE}_{\mathrm{CR}} = 0.0177$, $\mathrm{MAE}_{\mathrm{ICD}} = 0.7000$, and $\mathrm{RMSE}_{\mathrm{CR}} = 0.9930$. Respectively, and for pix2pix, they are 0.0862, 0.0870, 1.4000, and 1.4509, respectively.

Figure 7(a) shows the trend of the MAE and RMSE values of capacity ratio for both CycleGAN and pix2pix, and Figure 7(b) shows the trend of the MAE and RMSE values of intersection congestion degree. We can see that when epoch = 200, both CycleGAN and pix2pix gain a good performance, and performance improvement is not obvious when epoch = 1000. Thus, considering the balance between performance and training cost, we suggest a 200-epoch early stop.

Table 5 shows the performance under different learning rate settings. For example, 400/100 means that in the first 400 epochs, LR is kept with 0.0002, and in the following 100 epochs, we perform a linear decay. We can see that when the learning rate is 50/150, CycleGAN has the best performance of CR and ICD prediction has quiet small MAE and RMSE values: $\mathrm{MAE}_{\mathrm{CR}} = 0.0114$, $\mathrm{RMSE}_{\mathrm{CR}} = 0.0134$, $\mathrm{MAE}_{\mathrm{ICD}} = 0.5333$, and $\mathrm{RMSE}_{\mathrm{CR}} = 0.6245$. While for pix2pix, when the learning rate is 100/100, pix2pix has the best performance: $\mathrm{MAE}_{\mathrm{CR}} = 0.0929$, $\mathrm{RMSE}_{\mathrm{CR}} = 0.0985$, $\mathrm{MAE}_{\mathrm{ICD}} = 2.4750$, and $\mathrm{RMSE}_{\mathrm{CR}} = 2.9441$.The CycleGAN with 50/150 LR is better than the pix2pix 100/100 LR.

Figure 8(a) shows the trend of the MAE and RMSE values of capacity ratio for both CycleGAN and pix2pix, and Figure 8(b) shows the trend of the MAE and RMSE values of intersection congestion degree. Through different compositions within total 100, 200, and 500 epochs, we can see that for CycleGAN, the LR has relative small influence, while for pix2pix, the LR's influence is bigger and the best setting is within 200 epochs.

Table 6: $\mathrm{MAE}_{\mathrm{PCD}}$ and $\mathrm{RMSE}_{\mathrm{PCD}}$ of CycleGAN and pix2pix.

	$\mathrm{MAE}_{\mathrm{PCD}}$		$\mathrm{RMSE}_{\mathrm{PCD}}$	
	CycleGAN	pix2pix	CycleGAN	pix2pix
$k = 1$	0.3833	2.3500	0.4378	2.3611
$k = 2$	0.2500	0.5750	0.2550	0.7697
$k = 3$	0.1500	0.2250	0.1958	0.2398
$k = 4$	0.1833	1.5750	0.2550	2.4418
$k = 5$	0.4167	2.1500	0.4743	2.2383
$k = 6$	0.3167	0.8000	0.3719	0.8307
$k = 7$	0.8667	3.2500	0.8907	3.2550
$k = 8$	0.1833	0.2250	0.1958	0.2693

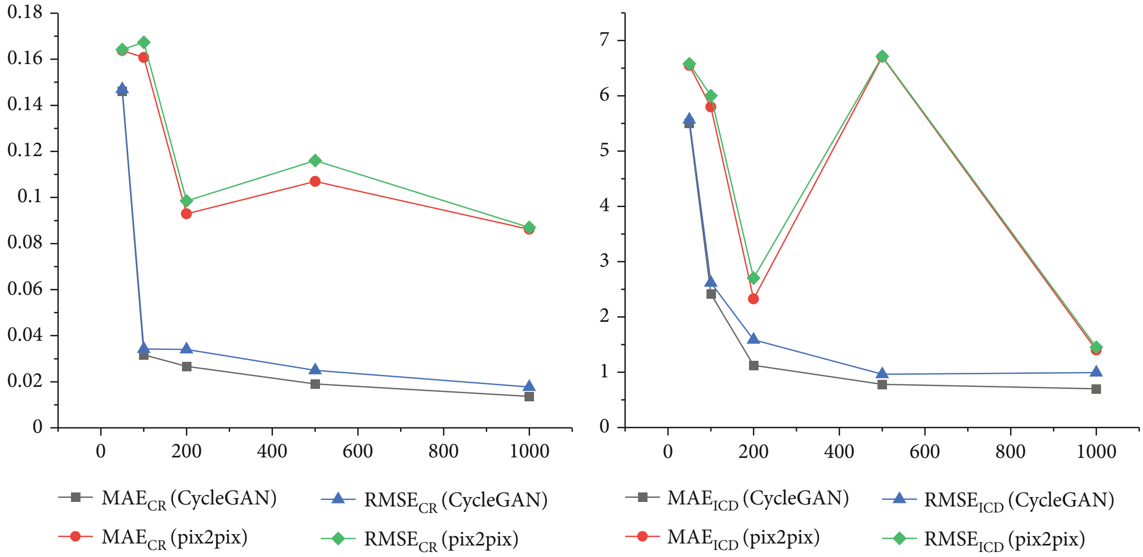
We further reveal the detailed values of each phase for MAE and RMSE of congestion degree in Table 6. We set training epoch as 200, and the LR settings for CycleGAN and pix2pix are 50/150 and 100/100, respectively. We can see that through comparing the values based on 8 phases of CycleGAN and pix2pix; for CycleGAN, the best results occur at $k = 3$, which have the lowest values (0.1500, 0.1958) of MAE and RMSE. Similarly, for pix2pix, the best results are at $k = 3$ with values 0.2250 and 0.2398 of MAE and RMSE.

We also give bar charts for MAE and RMSE of 8-phase congestion degree by Figure 9. In Figure 9(a), the smaller average value of MAE is for CycleGAN with value 0.3438. In Figure 9(b), we have similar results of RMSE; the average value of CycleGAN and pix2pix are 0.3845 and 1.5507, respectively. The MAE and RMSE of CycleGAN are smaller than those of pix2pix; this indicates a better robustness of CycleGAN compared with pix2pix.

6. Defense Discussion

For the relationship between the evaluation metric and the defense of attack, we have the following suggestions.

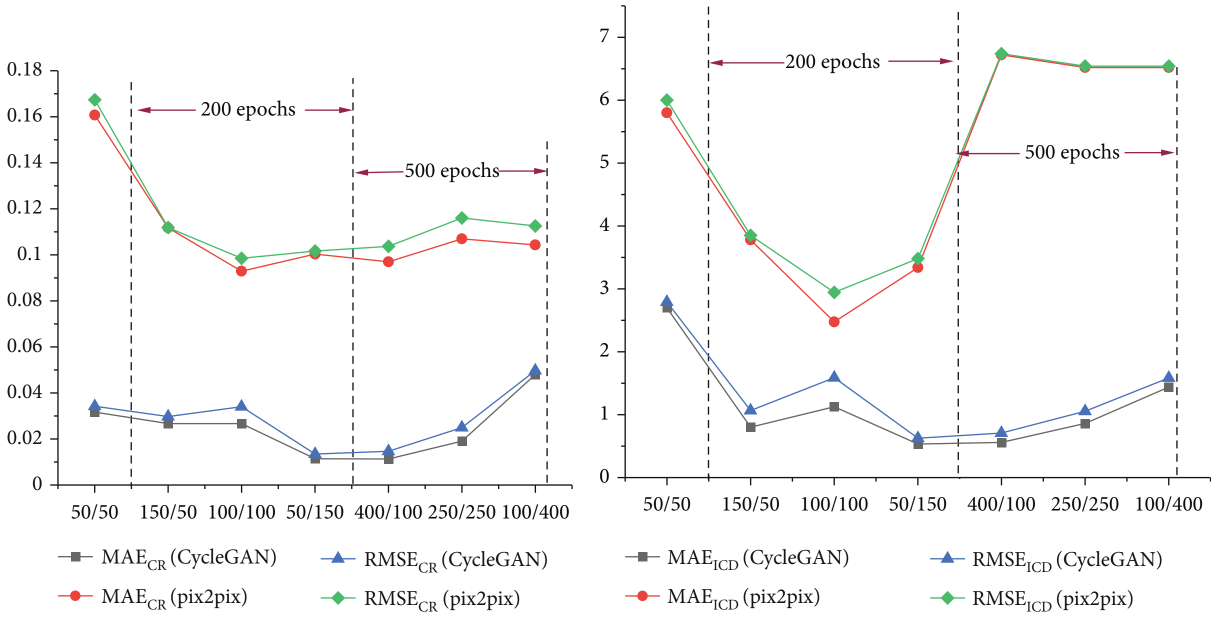
6.1. Attack Strategy Detection. In the signal planning stage of I-SIG system, the COP algorithm generates reasonable green light duration based on the queuing length of each phase estimated by the EVLS algorithm. As shown in our evaluation, the vehicle capacity ratio (CR) reflects the total number of the intersection. In the significance of defense, comparing the estimated queuing length by EVLS with the immediate evaluation metric CR is an efficient way to determine whether the attack vehicle is placed in corresponding phase. For instance, if the phase has long estimate queuing line with low CR index, a last-vehicle attack may occur; on the contrary, if the phase has small estimate queuing line with high CR index, a first-vehicle attack may occur. This can bring feasible defense and improve system robustness.



(a) MAE$_{CR}$ and RMSE$_{CR}$ trends under different epochs

(b) MAE$_{ICD}$ and RMSE$_{ICD}$ trends under different epochs

Figure 7: Training epoch influence on CycleGAN and pix2pix.



(a) MAE$_{CR}$ and RMSE$_{CR}$ of CycleGAN and pix2pix

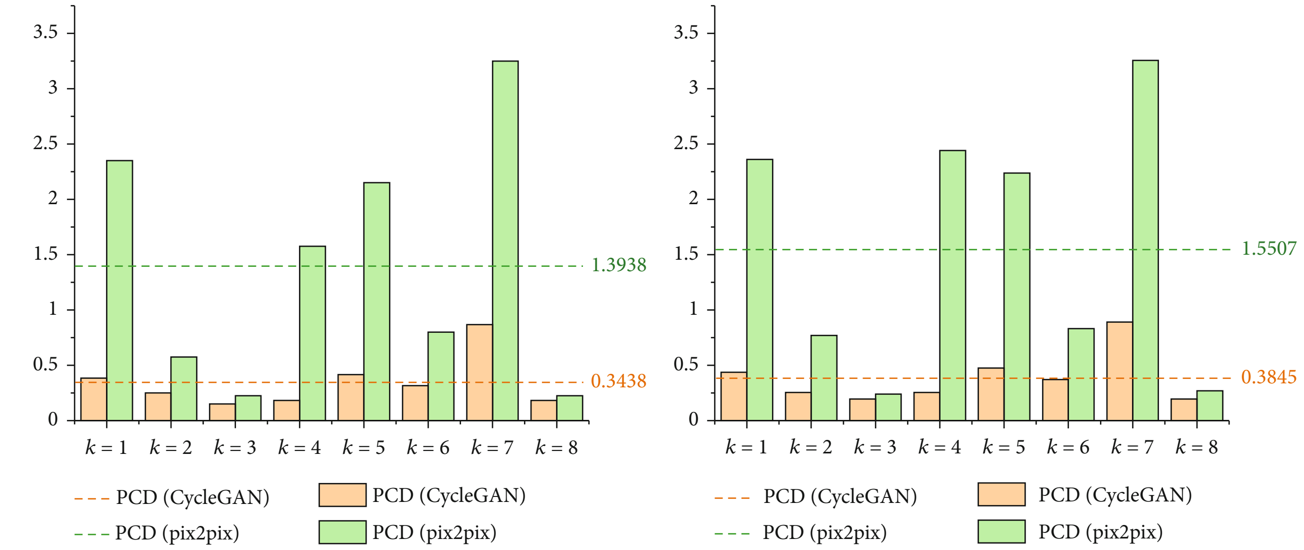
(b) RMSE$_{ICD}$ of CycleGAN and pix2pix

Figure 8: Learning rate influence on CycleGAN and pix2pix.

6.2. Robust Algorithm Design. As the CV-based intelligent transportation system are proved to be vulnerable to data spoofing attack, a notable problem is the lack of data check in the EVLS algorithm. For defense, a validity check procedure should be added to improve the robustness of the algorithm, in which suspicious data will be excluded from the arrival table, e.g., removing the vehicle at the end of the queuing line to defend the detected last-vehicle attack. Considering the long-term application of the CV-based intelligent transportation system, this is a future direction.

7. Related Work

7.1. Spoofing Attack Analysis. I-SIG is exposed to a data spoofing attack causing heavy congestion. Such attack belongs to position faking attack of GPS spoofing, but different with tunnel attack. In tunnel attack, each vehicle of a vehicular ad hoc network (VANET) [16–18] is equipped with a positioning system (receiver), and then the attack can be achieved using a transmitter generating localization signals stronger than those generated by the real satellites [19, 20];



(a) Bar chart of MAE_{PCD}

(b) Bar chart of RMSE_{PCD}

Figure 9: Bar chart of MAE_{PCD} and bar chart of RMSE_{PCD}.

then, 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 that authenticated vehicle only sends wrong position to affect the COP algorithm, which has lower attack cost and easier implementation. In such attack, the spoofing is just a causing factor, while the mechanism of COP algorithm is the key. In comparison, for GPS spoofing attack, our work focuses on the revealing of algorithm-level security analysis caused by spoofing, not the security of GPS spoofing or context-aware sensing [21–24] itself.

The previous work [7] mainly reveals the existence of such congestion attack on COP, analyzes the reason of COP decisions influence called last-vehicle advantage, and also explains how to use the data spoofing to launch an attack. However, it lacks consideration about the potential features and the quantified correlation between the attack and congestion degree. In comparison, we demystify the attack to I-SIG and corresponding congestion from machine learning perspective, through exploring different kinds of features based on unsupervised learning from attack image to congestion image via image search [25, 26], so as to explore new visualized analyzing method to reveal detailed attack results in each phase of intersection. In addition, as the first utilization of image feature in congestion attack, our work can provide a visualization for better understanding.

7.2. Congestion Prediction. Traffic congestion prediction has been studied a lot. Traditional traffic feature-based methods [27–29] are generally used in traffic congestion prediction, in which the traffic scenario is usually illustrated by manually set features such as location, speed, and delay of vehicle. Early researches are focused on single-site prediction based on one-dimensional traffic time series such as the ARIMA model [30] and the nearest neighbour method [31]. Recently, the trend has been shifted to prediction based on spatial temporal correlations between traffic flows [32–34], for instance, the vector ARMA model incorporating both spatial and temporal correlations, and the spatial econometrics models focused on congestion propagation over adjacent links. The core of the existing methods is as follows: They try to predict traffic congestions at one site based on the spatially and temporally correlated information from the sensors distributed on nearby roads, where the number of such sensors contributing to the prediction is referred to as data dimensionality. Recently, a LSTM model-based approach [35] was proposed for region-wide congestion prediction. In comparison, the attack-based congestion prediction is totally different, and it is because any classical traffic flow-related theory of spatial and temporal correlation does not well fit. Thus, this work does not focus on traditional traffic features. Even for image feature, we perform phase-based reprocessing and produce novel image for training; this is a different method for I-SIG congestion prediction towards a COP attack.

8. Conclusion

Towards the spoofing to connected vehicle technology, a congestion attack has been revealed on the COP algorithm of I-SIG. Due to the lack of visualized congestion analysis and attack phase explanation, we focus on the prediction of congestion attack. Compared to traditional congestion prediction, such attack-based congestion prediction is totally different, and it is because any classical traffic flow-related theory of spatial and temporal correlation does not well fit. We perform the first study to predict the congestion caused by spoofing attack based on adversarial generative network, through directly utilizing high-level image features of traffic.

In this paper, we propose a CycleGAN-based prediction approach, in which we design a weighted L1 regularization loss to learning and distinguish fine differences between last-vehicle attack and first-vehicle attack. We evaluate our approach empirically from real COP algorithm through VISSIM, and collect 4476 image samples of high quality for experiment, which shows the effectiveness of our approach compared to ground truth. We also find that 200 epochs can effectively prevent the training's mode collapse in our approach and have a satisfied performance as a baseline.

This work is expected to inspire a series of follow-up studies on security of I-SIG, including but not limited to (1) more machine learning-based approaches and (2) more multimodal feature fusion for visualized congestion analysis caused by spoofing attack.

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 there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (61972025, 61802389, 61672092, U1811264, and 61966009), Fundamental Research Funds for the Central Universities of China (2018JBZ103, No. 2019RC008), Science and Technology on Information Assurance Laboratory (614200103011711), and Guangxi Key Laboratory of Trusted Software (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/cvpilotapps.htm.

[3] "Cohda Wireless OBU and RSU," http://cohdawireless.com/Products/Hardware.aspx.

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

[5] Y. Feng, K. L. Head, S. Khoshmagham, and M. Zamanipour, "A realtime adaptive signal control in a connected vehicle environment," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 460–473, 2015.

[6] "Usdot: multimodal intelligent traffic safety system (mmitss)," https://www.its.dot.gov/researcharchives/dma/bundle/mmitssplan.htm.

[7] 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 2018 Network and Distributed System Security Symposium*, San Diego, CA, USA, February 2018.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, pp. 2672–2680, 2014.

[9] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycleconsistent adversarial networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, p. 22422251, Venice, Italy, October 2017.

[10] "PTV Vissim," http://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim.

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

[12] B. Ghena, W. Beyer, A. Hillaker, J. Pevarnek, and J. A. Halderman, "Green lights forever: analyzing the security of traffic infrastructure," in *Usenix WOOT*, San Diego, CA, USA, August 2014.

[13] D. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, http://arxiv.org/abs/1412.6980.

[14] J. Lu and L. Cao, "Congestion evaluation from traffic flow information based on fuzzy logic," in *Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems*, Shanghai, China, October 2003.

[15] P. Pongpaibool, P. Tangamchit, and K. Noodwong, "Evaluation of road traffic congestion using fuzzy techniques," in *TENCON 2007 - 2007 IEEE Region 10 Conference*, Taipei, Taiwan, October-November 2008.

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

[17] 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.

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

[19] 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.

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

[21] W. Niu, J. Lei, E. Tong et al., "Context-aware service ranking in wireless sensor networks," *Journal of Network & Systems Management*, vol. 22, no. 1, pp. 50–74, 2014.

[22] W. Niu, G. Li, Z. Zhao, H. Tang, and Z. Shi, "Multi-granularity context model for dynamic web service composition," *Journal of Network & Computer Applications*, vol. 34, no. 1, pp. 312–326, 2011.

[23] W. Niu, G. Li, H. Tang, X. Zhou, and Z. Shi, "CARSA: a context-aware reasoning-based service agent model for AI planning of web service composition," *Journal of Network & Computer Applications*, vol. 34, no. 5, pp. 1757–1770, 2011.

[24] E. Tong, W. Niu, G. Li et al., "Bloom filter-based workflow management to enable QoS guarantee in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 39, pp. 38–51, 2014.

[25] Z. Zhou, Q. M. J. Wu, Y. Yang, and X. Sun, "Region-level visual consistency verification for large-scale partial-duplicate image

search," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 16, no. 2, pp. 1–25, 2020.

[26] Z. Zhou, Y. Mu, and Q. M. J. Wu, "Coverless image steganography using partial-duplicate image retrieval," *Soft Computing*, vol. 23, no. 13, pp. 4927–4938, 2019.

[27] P. Mishra, R. Hadfi, and T. Ito, "Adaptive model for traffic congestion prediction," in *Trends in Applied Knowledge-Based Systems and Data Science. IEA/AIE 2016*, vol. 9799 of Lecture Notes in Computer Science, , pp. 782–793, Springer, Cham.

[28] X. Ma, H. Yu, Y. Wang, and Y. Wang, "Large-scale transportation network congestion evolution prediction using deep learning theory," *PLoS One*, vol. 10, no. 3, article e0119044, 2015.

[29] S. Yang, "On feature selection for traffic congestion prediction," *Transportation Research Part C*, vol. 26, pp. 160–169, 2013.

[30] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results," *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[31] B. L. Smith, B. M. Williams, and R. K. Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," *Transportation Research*, vol. 10, no. 4, pp. 303–321, 2002.

[32] W. Min and L. Wynter, "Real-time road traffic prediction with spatio-temporal correlations," *Transportation Research Part C*, vol. 19, no. 4, pp. 606–616, 2011.

[33] L. Romaszko, "IEEE ICDM 2010 contest: traffic prediction – jams," in *2010 IEEE International Conference on Data Mining Workshops*, pp. 1366–1368, Sydney, NSW, Australia, December 2010.

[34] J. He, Q. He, G. Swirszcz et al., "Ensemble-based method for task 2: predicting traffic jam," in *2010 IEEE International Conference on Data Mining Workshops*, pp. 1363–1365, Sydney, NSW, Australia, December 2010.

[35] S. Mohanty, A. Pozdnukhov, and M. Cassidy, "Region-wide congestion prediction and control using deep learning," *Transportation Research Part C: Emerging Technologies*, vol. 116, p. 102624, 2020.