Predicting Congestion Attack of Variable Spoofing Frequency for Reliable Traffic Signal System

Yingxiao Xiang, Tong Chen, Yike Li, Yunzhe Tian, Wenjia Niu(✉), Endong Tong(✉), Jiqiang Liu, Bowei Jia, Yalun Wu, and Xinyu Huang

Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, Beijing 100044, China
{niuwj,edtong}@bjtu.edu.cn

Abstract. As a key component of next-generation transportation systems, the intelligent traffic signal system is designed to perform dynamic and optimal signal control. The USDOT (U.S. Department of Transportation) has sponsored a kind of such system - I-SIG based on Controlled Optimization of Phases (COP). Unfortunately, it has been revealed that a serious congestion attack can be caused just by one vehicle's data spoofing. However, the existing methods focus on detecting the congestion attack and have a certain disadvantage of delay even facing periodic attacks. Thus, how to timely detect and even predict the congestion attack has become a key issue. Considering that a practical and effective congestion attack is usually continuous and periodic, we propose a novel approach for congestion attack prediction. Firstly, we set up a spoofing attack environment and collect traffic flows of variable spoofing frequencies. Among congestion attack-caused flows, we define and extract 30 important features and implement ensemble learning to build correlations between traffic flow features and abnormal congestion and attack frequency. Through supervised learning of historical data, we can recognize the current attack frequency and further realize the prediction of the subsequent congestion attack. We also report on necessary and experienced tricks for performance improvement. Extensive experiments and analyses have been conducted to demonstrate the prediction capability of our proposed approach.

Keywords: Congestion attack · Prediction · Supervised learning · Security analysis · Traffic signal system

1 Introduction

Connected vehicle (CV) technology [1,4], empowering vehicles to communicate with the surrounding environment, such as Roadside Units (RSU) and traffic signal control infrastructure, is now transforming today's transportation systems. As a critical component, the intelligent traffic signal system is responsible

W. Shi et al. (Eds.): SPNCE 2021, LNICST 423, pp. 219–237, 2022.
https://doi.org/10.1007/978-3-030-96791-8_16

for performing dynamic and optimal signal control based on automatic traffic situation awareness. Since September 2016, the USDOT (U.S. Department of Transportation) has sponsored a kind of such system named I-SIG built on the classic algorithm: Controlled Optimization of Phases (COP) [14,23]. As a CV Pilot Program, I-SIG systems [5] are currently under testing in some USA states, including California, Florida, and New York, before spreading widely.

Table 1. I-SIG congestion increasing ratios under last-vehicle attacks of 5 different spoofing frequencies. A last-vehicle attack refers to the addition of an attack vehicle at the end of a queue.

Attack frequency	Sampling time (s)				
	400	800	1200	1600	1800
f_1 (Every 1 stage)	185.6%	198.4%	148.2%	292%	259.6%
f_2 (Every 2 stages)	195.8%	202.9%	128.2%	231.1%	231.2%
f_3 (Every 3 stages)	87.6%	98.5%	94.1%	188.9%	259.6%
f_4 (Every 4 stages)	46.9%	46.2%	36.4%	103.1%	140.2%
f_5 (Every 5 stages)	0	26.7%	2.1%	9.3%	11.9%

Unfortunately, there is a severe security problem with I-SIG, which is revealed by Qi Alfred Chen [10] in NDSS2018. A congestion attack can be launched via data spoofing of one attack vehicle. Due to the vulnerability of the COP algorithm, attackers can compromise the On-Board Units (OBU) on their vehicles and send malicious messages such as speed and location, which can affect the traffic control decisions at proper timing and cause unexpected heavy traffic congestion. Some data shows that one single attack vehicle can cause more than 11 times the total delay, which significantly hinders the development and future large-scale deployment. After repeated the attack scenario, we find that different spoofing frequencies can cause a similar congestion effect even for the last vehicle. The attack frequency is inversely proportional to the attack time interval. Table 1 shows I-SIG congestion increasing ratios under last-vehicle attacks of 5 different spoofing frequencies. f_1 is the highest frequency adopted in Qi Alfred Chen's work, i.e., attacking at each stage. $f_2 \sim f_5$ are low-grade attack frequencies designed in our work and denote attacking every 2, 3, 4, 5 stages, respectively. We can see that f_2, f_3 and f_4 make heavy congestion 30 min later as well, indicating that the frequency-reduced attacks exist.

Therefore, although previous work [10] reveals the congestion attack via the last vehicle, detecting it only through black-box watching of traffic flow is still an open issue. The questions are what traffic flow pattern under congestion attack is, and given a period of traffic flow, can we timely predict whether or not the system suffers a congestion attack, or even corresponding attack frequency? Ignoring such abilities will harm the security situation awareness of I-SIG. Therefore, it is significant to study the predictability of congestion attack,

the timeliness of prediction, and various composed features' influence, but much remains unexplored.

In this work, we propose a prediction approach of traffic congestion attack, which can recognize the abnormal congestion and attack frequency; and timely predict the forthcoming congestion attack. This work is the first to perform a complete prediction task in supervised machine learning using historical data to demystify the predictability of the I-SIG congestion attack, from manual instance collecting, labeling, feature engineering to mainstream single learner implementation and ensemble learning. In our prediction approach, there are four singer learners, including decision tree (DT) [24], support vector machine (SVM) [8], logic regression (LR) [12], and k-nearest neighbors classifier (KNN) [16]. Compared with SVM, DT, LR, etc., Deep Neural Networks (DNN) have higher computational costs and require large-scale data for training, so they are not considered to be used here. To overcome a single learner's weakness, we employ bagging, boosting, and stacking techniques in ensemble learning, respectively. We also utilize common tricks, including rescaling, oversampling, and hyperparameter optimization for improvement. In our experiment, the dimensionality varies from 20 to 600 based on 540 traffic flow samples collected and crafted. The efficiency of the proposed approach is demonstrated through simulations, compared with a baseline approach [28]. By defining quantified traffic flow features, the influence of different feature compositions for feature selection is deeply analyzed. We eventually discover and report some distinguished patterns of traffic flow with high confidence for both general and each frequency-specified congestion attack through massive analysis efforts.

We summarize our contributions as follows:

- We are the first to study traffic congestion attack under various frequencies in the I-SIG system in order to predict the subsequent congestion attack.
- Based on solid feature engineering and special learning tricks, we propose a unified ensemble learning framework to perform supervised classification tasks.
- We evaluate our approach empirically from an actual COP algorithm through VISSIM [3]. We reveal the promising predictability of traffic congestion attack and the timeliness of congestion attack prediction, analysis the influence of different feature compositions to detection performance, and report on explainable patterns to distinguish congestion attack flows under different frequencies.

The rest of the paper is organized as follows: Sect. 2 discusses the related works. Section 3 introduces the preliminary about our research. Section 4 describes our proposed prediction framework. Section 5 demonstrates and analyses the performance of our approach by simulation, followed by a conclusion of the paper in Sect. 6.

2 Related Work

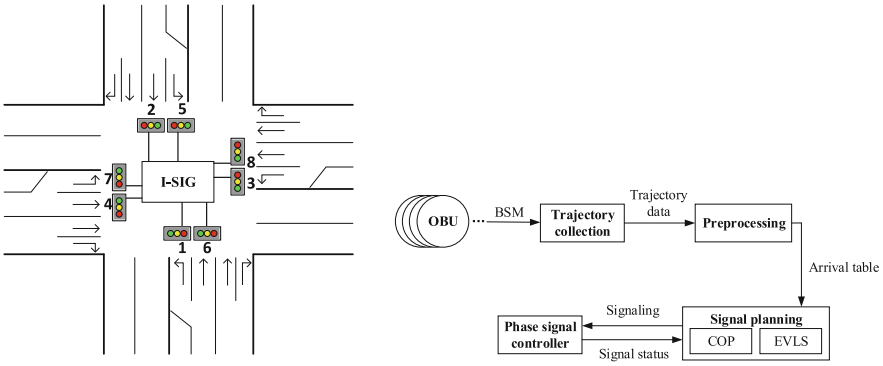
Many studies on traffic congestion problems address the optimization of the traffic signal control system to detect and reduce congestion (see, for example, the studies [6,7,11,13,17,19,26,27]). For instance, the study of [11] alleviated traffic congestion by analyzing and predicting the traffic states. Jaleel et al. [17] proposed a novel Multi-Agent RL approach to control the traffic signals phase and timing to reduce congestion. In [6], the authors proposed a rapid and reliable traffic congestion detection method based on the modeling of video dynamics using deep residual learning and motion trajectories.

The above studies focused mainly on the detection of traffic congestion, rather than the congestion attack problems. From the congestion attack aspect, Jeske [18] demonstrated how hackers can take control of navigation systems in practice, in order to trick navigation services and cause congestion. Chen et al. [10] revealed a threat of data spoofing over the intelligent traffic signal system - I-SIG. The data spoofing can cause traffic severe congestion via one single attack vehicle.

Data spoofing attack in [10] belongs to position faking attack of GPS spoofing, but is different with tunnel attack. In a tunnel attack, where each vehicle of a vehicular ad hoc network (VANET) [30,32] is equipped with a positioning system (receiver), a transmitter that generates stronger localization signals than those generated by the real satellites [9,21] are used, so that 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 that authenticated vehicle only sends the wrong position to affect the COP algorithm, which has a lower attack cost and easier implementation. In such an attack, the spoofing is just a causing factor, while the mechanism of the COP algorithm is the key. For GPS spoofing attack, our work focuses on the revealing of algorithm-level security analysis caused by spoofing, rather than the security of GPS spoofing itself. The work in [10] lacks consideration about the potential features and the quantified correlation between the attack and congestion degree. In comparison, we develop a unified ensemble learning framework to demystify the predictability of I-SIG congestion attack based on solid feature engineering and special learning tricks.

There are only very few studies on the prediction methods of traffic congestion caused by spoofing attacks. Li et al. [20] proposed a CycleGAN-based prediction approach using traffic image feature. Their approach reflects the relation between attack and the caused congestion. They revealed the vulnerability to the spoofing attack in [10] from the perspective of image features. Unlike [20], our approach would extract traffic flow feature and find the traffic flow patterns under different attack frequencies. The authors in [28] proposed an explainable congestion attack prediction approach using a deep learning model - TGRU. They try to explain the relationship between the traffic flow feature and the lanes where the congestion attack vehicle locates. While their model uses the traffic flow of one second as a sample, our approach uses the traffic flow of 20 seconds to represent the traffic feature better.

The main difference between our work and the previous studies is that we propose an approach for detecting the congestion attack to predict the subsequent congestion attack. The main focus of our work is the extraction of traffic flow features, the recognition of the abnormal congestion and the current attack frequency, and the generation of traffic flow patterns to predict the congestion attack.



(a) 8-phase I-SIG signal control (b) Data flow of the I-SIG system

Fig. 1. I-SIG system overview.

3 Preliminaries

3.1 I-SIG Data Flow

In I-SIG, there are 8 phases, as shown in Fig. 1(a), called phases. The data flow of the I-SIG system is revealed in Fig. 1(b). Each OBU of vehicles sends Basic Safety Messages (BSM) to the RSU for a trajectory collection in real time. Such data will then be preprocessed to form an arrival table (see Table 2) as an input to signal planning with COP and EVLS modules. If the penetration rate (PR) of OBU for vehicles is below 95%, the arrival table will be sent to the Estimation of Location and Speed (EVLS) algorithm for an update. The EVLS algorithm will estimate the trajectory data of the unequipped vehicles. Otherwise, it will be sent directly to COP for planning. According to the results of COP, a downward signaling command is transferred to the phase signal controller. After each stage of signal control, the following signal status is returned as feedback for continuous COP planning.

In Table 2, $T_i = i(0 \leq i \leq M)$ denotes the time to arrive at the stop bar from the current location. I-SIG sets $M = 130$ seconds, covering an over two-minute BSM statistic. $N_{ij}(i \in [0, M], j \in [1, 8])$ denotes that in phase j, there will be N_{ij} vehicles are going to reach the stop bar only within T_i seconds. Here, the stop bar is set in front of the traffic light, as what we can see at an actual road intersection.

Table 2. Arrival table.

Arrival time (T_M)	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}

The EVLS is based on Wiedemann's car-following model which fills the blank monitoring area of the monitoring segment and inserts vehicle data between equipped vehicles. The key is to estimate the queued vehicles, and it is critical to estimate the queue length. The queue is assumed to always start from the stop bar, the last vehicle in queue needs to be found to determine the queue length.

First, the history distances to the stop bar and stopping time of the last stopped connected vehicle and the second to the last stopped connected vehicle in the queue are calculated, noted 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 the queue propagation speed v_q is constant, we have $v_q = \frac{L_{q1}-L_{q2}}{T_{q1}-T_{q2}} = \frac{L_{es}-L_{q1}}{T_c-T_{q1}}$. Then, $L_{es} = L_{q1} + v_q(T_c - T_{q1})$. If the average vehicle length is C, the number N_{0i} of vehicles in the queue is then calculated as $N_{0i} = \lceil L_{es}/C \rceil, i \in [1, 8]$.

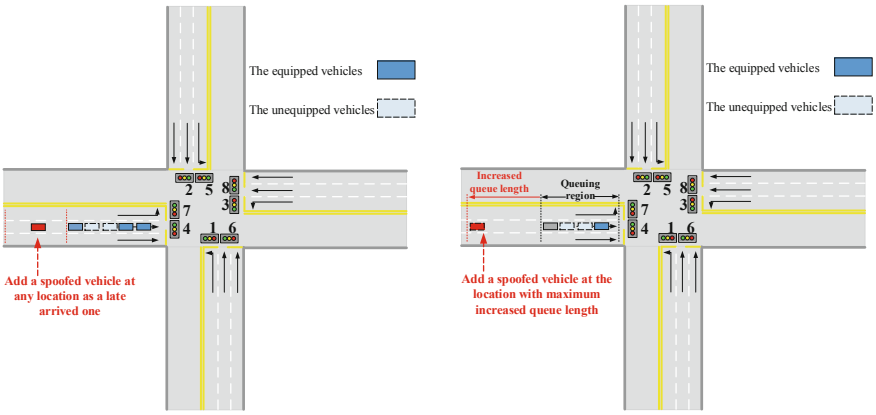
However, while this estimation supports a low penetration rate effectively, it also introduces a new threat of data spoofing attack to COP.

3.2 Congestion Attack

In our case, the main goal of the attack vehicle is to spoof the signal planning algorithm COP by incorrect traffic flow data leading to an ineffective traffic signal schedule, which sets a longer green time for the lanes with a shorter queue length. There are two attack strategies of data spoofing proposed in I-SIG (see Fig. 2). The first one is a direct attack on the arrival table without considering penetration rate; the second one is indirect attack on EVLS when penetration rate is less than 95%.

The first strategy, arrival time and phase spoofing, applies to both full deployment and transition periods. The attacker alters the arrival time of the vehicle and its requested phase by changing the location and speed information in the vehicles' BSM message, thus changing the corresponding arrival table elements in Table 2. This attack strategy can directly attack input data flow regardless of PR. As shown in Fig. 2(a), the attacker adds a spoofed vehicle into the original vehicle queue at any location as a late arrived one. The join of spoofed vehicle makes the queue of vehicles on the lane longer, and there is an increment of the duration of green light allocated by COP algorithm for the current phase, which delays the next start time of green light of all the phases, increasing the delay for vehicles to pass.

The second strategy is queue length spoofing, which only works during the transition period. The purpose of this strategy is to extend the queue length estimated by EVLS algorithm through changing the location and speed values in the BSM message. As shown in Fig. 2(b), the attacker adds a stopped vehicle that has the farthest distance to the stop bar. Since the EVLS algorithm estimates the queue length based on the location of the last stopped connected vehicle, this attack results in an increase in the estimated queue length L_{es}, and the number N_{0i} of vehicles in queue.



(a) Direct attack on arrival table without considering penetration rate

(b) Indirect attack on EVLS when penetration rate is less than 95%

Fig. 2. Two strategies of congest attack.

4 Congestion Attack Prediction

We propose a unified ensemble learning-based framework for binary classification and multi-class classification on I-SIG traffic flow. As Fig. 3 shows, there are three parts: preprocessing, training set, and ensemble learning. In preprocessing, we extract features from the collected traffic flow of the I-SIG system. Next, we use standardization, a kind of rescaling trick, to adjust feature values. Furthermore, through labeling, we separate two dataset D_1 and D_2 for binary classification and multi-class classification respectively. In D_1, for the traffic samples that are not attacked, we give them a label of 0, otherwise 1. While for D_2, we give labels 1, 2, 3, 4 and 5 to those samples with five attack frequencies f_1 to f_5 respectively. To ensure a balanced sample dataset, we also utilize SMOTE oversampling trick. Ensemble learning is the core component of this framework. It involves complete ensemble learning methods, including bagging, boosting and stacking, and grid search trick, an optimization of hyperparameter tuning. We chose four basic learner types: DT, SVM, LR, and KNN. For each class, we train 10 basic learners. Bagging and boosting are used for learners of the same

type, while stacking is used for heterogeneous learners. The above framework is designed for comprehensive supervised learning to discover patterns of traffic flow for both general and each frequency-specified congestion attack.

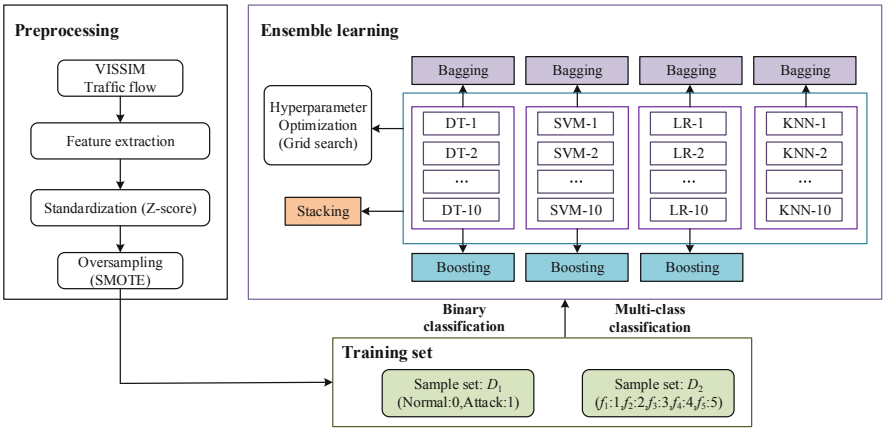


Fig. 3. Ensemble learning-based framework for binary classification and multi-class classification on I-SIG traffic flow.

4.1 Feature Definition

We have feature definition as following:

- *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 $C_{total}^{max} = \sum_{k=1}^{8} C_k^{max}$, then the vehicle capacity ratio can be denoted by $CR = \sum_{k=1}^{8} N_k / C_{total}^{max}$, where N_k is the vehicle number of the kth phase.
- *Congestion degree (CD).* The vehicle number of queuing of the kth phase is denoted as Q_k, and Q_{normal} is the vehicle number of normal queuing, it's a constant, then the congestion degree of the kth phase can be computed by $PCD_k = Q_k / Q_{normal}$, and the global congestion degree for an intersection is $ICD = \sum_{k=1}^{8} PCD_k$.
- *Attack acceleration.* Let t_0 be the start time of data spoofing attack, then the acceleration of CR, PCD_k and ICD at time t are respectively calculated by: $\alpha_{CR}(t) = (CR(t) - CR(t_0))/(t - t_0)$, $\alpha_{PCD}(t,k) = (PCD(t,k) - PCD(t_0,k))/(t - t_0)$ and $\alpha_{ICD}(t) = (ICD(t) - ICD(t_0))/(t - t_0)$.
- *Attack amplification ratio.* Let t_0 be the start time of data spoofing attack, then the amplification ratio of CR, PCD_k and ICD at time t are respectively computed by: $\beta_{CR}(t) = CR(t)/CR(t_0)$, $\beta_{PCD}(t,k) = PCD(t,k)/PCD(t_0,k)$ and $\beta_{ICD}(t) = ICD(t)/ICD(t_0)$.

4.2 Constructing Training Samples

According to whether it is for the whole intersection or a specific phase, features are divided into macro features and micro features, as shown in Table 3. For discussing interpretability, macro features measure the congestion characteristics of the whole intersection, and micro features measure the single signal phase. For a traffic flow of 1800 s, we only sample the first 10 time of flow head and the last 10 time of flow tail. We choose macro or micro or both features for flow head,and then we choose the same ones from flow tail. Therefore, the size of composed features varies from 20 to 600.

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

	Flow head (10 s)	Flow tail (10 s)
Macro features	CR, α_{CR}, β_{CR}, ICD, α_{ICD}, β_{ICD}	CR, α_{CR}, β_{CR}, ICD, α_{ICD}, β_{ICD}
Micro features	PCD_1, PCD_2, $\cdots$, PCD_8, α_{PCD_1}, α_{PCD_2}, $\cdots$, α_{PCD_8}, β_{PCD_1}, β_{PCD_2}, $\cdots$, β_{PCD_8}	PCD_1, PCD_2, $\cdots$, PCD_8, α_{PCD_1}, α_{PCD_2}, $\cdots$, α_{PCD_8}, β_{PCD_1}, β_{PCD_2}, $\cdots$, β_{PCD_8}

We use Z-score as standardization, a kind of rescaling trick, to adjust feature values. For values $(x_1, x_2, ..., x_n)$ of one feature in all samples, the new value is computed by $x' = \frac{x_i - \bar{x}}{s}$, in which s is the standard deviation and $\bar{x}$ is the mean value of $(x_1, x_2, ..., x_n)$.

4.3 Ensemble Learning

Bagging and Boosting. Bagging is one of the simplest but most common techniques for constructing ensembles. Based on bootstrap sampling of dataset, bagging shapes several different training sets, with the purpose of training basic learners on different training sets. The final model will be achieved by aggregating these independent base learners. In this framework, we utilize bagging technique for homogenous base learners. For D_1 of 900 samples and D_2 of 450 samples, our bootstrap sampling selects 70% random sampling and runs 10 times to train 10 base learners for each type of classifiers like DT, SVM, KNN, and LR.

Compared with the parallel construction of base learners in bagging, boosting [22] establishes a set of base classifiers in sequence for boosting the performance of a set of weak classifiers into a robust classifier. The first one learns from the whole data set, while the following learns from training sets based on the performance of the previous one. The misclassified examples are marked, and their weights increased, so they will have a higher probability of appearing in the training set of the next learner. By repeating the process, several weak classifiers will be established to achieve better classification performance. In this

work, we use AdaBoost [15] to perform boosting. Similarly, for each classifier like DT, SVM, and LR, we also set 10 base learners to boost sequentially. Since KNN does not support sample weights, we ignore the boosting of KNN.

Stacking. Stacking is used for heterogeneous learners. We implement the stacking algorithm [29] shown in Algorithm 1.

Algorithm 1. The Stacking algorithm

Require: Data set $D = (x_1,y_1),(x_2,y_2),\cdots,(x_m,y_m)$;
First-level learning algorithms $L_1, \cdots, L_N$;
Second-level learning algorithm L.

for $j = 1, \cdots, N$ **do**
  $h_j = L_j(D)$ %Train a first-level individual learner h_j by applying the first-level
end for %Learning algorithm L_j to the original data set D
$D' = \varnothing$ %Generate a new data set
for $i = 1, \cdots, m$ **do**
  for $j = 1, \cdots, N$ **do**
    $z_{ij} = h_j(\mathbf{x_i})$ %Use h_j to classify the training example $\mathbf{x_i}$
  end for
  $D' = D' \bigcup \{((z_{i1}, z_{i2}, \cdots, z_{iN}), y_i)\}$
end for
$h' = L(D')$ % Train the second-level learner h' by applying the second-level
% Learning algorithm L to the new data set D'

Ensure: $H(\mathbf{x}) = h'(h_1(\mathbf{x}), \cdots, h_N(\mathbf{x}))$

5 Experiment

5.1 Setup

Through running COP and VISSIM for real-time traffic flow signal control, the corresponding traffic simulation is carried out to collect the primary traffic flow data. The traffic flow data are collected without attack and with different attack frequencies ($f_1 \sim f_5$) and converted into the features proposed in Sect. 4. D_1 class labels are 0 (without attack) and 1 (with attack); D_2 is labeled $1 \sim 5$ ($f_1 \sim f_5$). The experimental environment and sample datasets configuration are shown in Table 4.

Table 4. Experimental environment and sample datasets configuration.

Platform	Experimental Environment	Environmental Configuration
COP & 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
Dataset	Sample and feature number	Label
D_1	900, 600	Normal:0, Attack:1
D_2	450, 600	$f_1 : 1$, $f_2 : 2$, $f_3 : 3$, $f_4 : 4$, $f_5 : 5$

5.2 Experiment Process

The prediction process of our approach is described as follows.

1) Collect a period of primary traffic flow data and convert it into the form of features we defined in Sect. 4.
2) The ensemble learning model trained on dataset D_1 can distinguish whether the congestion attack is occurring or not, after inputting the converted traffic data.
3) If the prediction result of step 2) is "Attack", the attack frequency can be output by the ensemble learning model trained on dataset D_2.
4) Compare the appearing frequency f_v of vehicles at the road junction with the attack frequency f_a detected at step 3), the congestion attack vehicles are found if f_v is around to f_a.
5) The discovery of attack vehicles can help the defense of I-SIG against traffic congestion attack.

In the following evaluation, extensive experiments and analyses are conducted, including the comparison with the baseline approach to demonstrate the efficiency of our defined feature and the proposed approach, the influence analyses of various composed features to find the best feature combination for different ensemble learning, the performance comparison of different ensemble learning to obtain the best learning technic, and the patterns analyses of traffic flow to explain the importance of some features.

5.3 Performance Metrics

We adopt two widely-used evaluation protocols, accuracy (ACC) and area under the curve (AUC). For D_1 dataset, we report the ACC, ROC curve [25] with AUC value and for D_2 dataset, we only report the ACC. Also, we present decision trees for both D_1 and D_2 to show the pattern of class. Here we set tree depth = 5.

5.4 Comparison with Baseline

We compare three different approaches which detect congestion based on traffic flow feature, in term of ACC. The first approach, Seasonal Auto-Regressive Integrated Moving Average (SARIMA) method [31], is prevalent in time-series prediction problems. The second approach is based on a deep learning model - TGRU [28]. The third approach is our proposed method based on ensemble learning (EL). In this case, we choose the ensemble learning of boosting with SVM, considering that the full composed feature set has the best ACC of 0.852 in that condition.

We carry out experiments for different approaches under different traffic flow feature sets. For the SARIMA method, we choose the primary traffic flow data as the traffic flow feature FS_1. In [28], the traffic flow feature set FS_2 consists of wait, slow, free, stop bar distance of 8 lanes, 32 features in total. The feature set FS_3 of our approach is shown in Table 3. According to the three approaches, we construct the three feature sets based on the traffic flow data we collect. Here, it is predicted whether or not a congestion attack occurred.

As shown in Table 5, the ACC values of SARIMA, TGRU, and EL on the feature set FS_3 are 0.784, 0.790, and 0.852, respectively. Our approach has the best ACC value, which demonstrates that our approach is superior to others. From the perspective of the feature set, the ACC value of each approach on the feature set FS_3 is higher than on other feature sets, this indicates the effectiveness of our defined traffic flow features.

Table 5. Comparison of different prediction approaches.

Feature set	FS_1(SARIMA)	FS_2(TGRU)	FS_3(EL)
SARIMA	0.744	/	0.784
TGRU	/	0.782	0.790
EL	0.762	0.820	0.852

Table 6. Influence of different feature composition on ACC in Bagging.

Normal/Attack					Five attacks				
	DT	SVM	LR	KNN		DT	SVM	LR	KNN
CR	0.722	0.796	**0.833**	0.765	CR	0.874	**0.904**	0.852	**0.911**
ICD	0.728	0.772	0.815	0.704	ICD	0.763	0.837	0.83	0.748
CR, ICD	0.728	0.784	0.809	0.704	CR, ICD	0.867	0.837	0.822	0.748
$CR, \alpha_{CR}, \beta_{CR}$	0.741	0.759	0.784	0.728	$CR, \alpha_{CR}, \beta_{CR}$	0.881	0.896	0.889	**0.911**
$ICD, \alpha_{ICD}, \beta_{ICD}$	0.747	0.796	0.796	0.716	$ICD, \alpha_{ICD}, \beta_{ICD}$	0.785	0.8	0.837	0.756
$CR, \alpha_{CR}, \beta_{CR}$, $ICD, \alpha_{ICD}, \beta_{ICD}$	0.728	0.735	0.753	0.716	$CR, \alpha_{CR}, \beta_{CR}$, $ICD, \alpha_{ICD}, \beta_{ICD}$	0.889	**0.911**	**0.904**	0.874
Full composed feature set	0.747	0.735	0.728	0.722	Full composed feature set	0.867	**0.926**	**0.926**	**0.919**

5.5 Feature Influence on Accuracy

We evaluate ACC of six feature compositions and even the full composition scheme. As Table 6 shows for bagging, the maximal ACC of D_1 is 0.833 from LR's bagging for single CR feature. For D_2, the ACC of {SVM, KNN}/CR, KNN/{CR, α_{CR}, β_{CR}}, {SVM, LR}/{CR, α_{CR}, β_{CR}, ICD, α_{ICD}, β_{ICD}} and {SVM, LR, KNN}/{full composed feature set} are beyond 0.9.

Table 7. Influence of different feature composition on ACC in boosting

Normal/Attack				Five attacks			
	DT	SVM	LR		DT	SVM	LR
CR	0.747	**0.852**	0.845	CR	0.874	0.889	0.489
ICD	0.722	0.827	0.815	ICD	0.756	0.763	0.533
CR, ICD	0.716	0.827	0.815	CR, ICD	0.896	0.815	0.541
$CR, \alpha_{CR}, \beta_{CR}$	0.741	**0.852**	0.821	$CR, \alpha_{CR}, \beta_{CR}$	**0.904**	0.844	0.585
$ICD, \alpha_{ICD}, \beta_{ICD}$	0.735	**0.852**	0.815	$ICD, \alpha_{ICD}, \beta_{ICD}$	0.793	**0.915**	0.785
$CR, \alpha_{CR}, \beta_{CR}$, $ICD, \alpha_{ICD}, \beta_{ICD}$	0.747	**0.852**	0.809	$CR, \alpha_{CR}, \beta_{CR}$, $ICD, \alpha_{ICD}, \beta_{ICD}$	**0.904**	0.852	0.689
Full composed feature set	0.735	**0.852**	0.815	Full composed feature set	**0.904**	**0.926**	0.763

Table 8. Influence of different feature composition on ACC in stacking

	Normal/Attack	Five attacks
CR	0.781	0.899
ICD	**0.794**	0.777
CR, ICD	0.786	0.879
$CR, \alpha_{CR}, \beta_{CR}$	0.781	**0.921**
$ICD, \alpha_{ICD}, \beta_{ICD}$	0.791	0.756
$CR, \alpha_{CR}, \beta_{CR}, ICD, \alpha_{ICD}, \beta_{ICD}$	0.757	**0.911**
Full composed feature set	0.754	**0.915**

As Table 7 shows for boosting, the maximal ACC of D_1 is 0.852 from SVM's boosting for most feature sets excepting {ICD} and {CR, ICD}. For D_2, the ACC of DT/{CR, α_{CR}, β_{CR}}, SVM/{ICD, α_{ICD}, β_{ICD}}, DT/{CR, α_{CR}, β_{CR}, ICD, α_{ICD}, β_{ICD}} and {DT, SVM}/{full composed feature set} are bigger than 0.9. Table 8 shows that the maximal ACC of D_1 is 0.794 for a single ICD feature. For D_2, the ACC of {CR, α_{CR}, β_{CR}}, {CR, α_{CR}, β_{CR}, ICD, α_{ICD}, β_{ICD}} and {full composed feature set} are beyond 0.9. The ensemble classification has satisfying ACC, and in practice, we recommend CR or {CR, α_{CR}, β_{CR}} for efficiency.

5.6 Comparison of Bagging, Boosting and Stacking

We choose CR as the only feature in this comparison. Table 9 and Table 10 report the ACC comparison without and with grid search, respectively. SVM's boosting has a maximum ACC of 0.852 on D_1, while KNN's bagging has a maximum ACC of 0.911 on D_2. With grid search, the ACC of LR's bagging has the greatest 7.440% improvement.

Table 9. CR-based ACC comparison without grid search.

Accuracy	Bagging				Boosting			Stacking
	DT	SVM	LR	KNN	DT	SVM	LR	
Normal/Attack	0.722	0.79	0.833	0.728	0.716	**0.852**	0.84	0.757
Five attacks	0.874	0.881	0.793	**0.911**	0.867	0.881	0.474	0.898

Table 10. Grid search-based ACC improvement.

Accuracy inc.(%)	Bagging				Boosting			Stacking
	DT	SVM	LR	KNN	DT	SVM	LR	
Normal/Attack	0	+0.759%	0	+5.082%	+4.330%	0	+0.595%	+3.170%
Five attacks	0	+2.611%	**+7.440%**	0	+0.807%	+0.908%	+3.164%	+0.111%

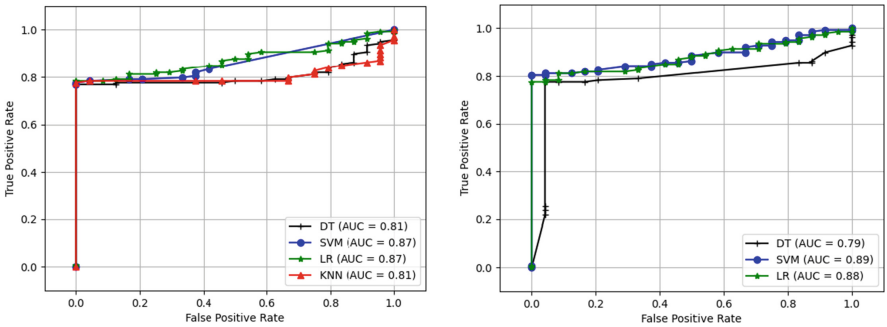
We also give the training time of different ensemble learning in Table 11, and we can see that bagging is more efficient. Besides, AUC is used to compare bagging with boosting. As Fig. 4 shows, SVM and LR have the best AUC values in bagging, while SVM has the best AUC value 0.89 in boosting. There is only a little difference between bagging and boosting.

Table 11. Training time across different ensemble learning.

Training time(s)	Bagging				Boosting			Stacking
	DT	SVM	LR	KNN	DT	SVM	LR	
Normal/Attack	0.819	1.959	1.742	0.698	0.968	15.105	1.276	4.620
Five attacks	1.185	1.860	4.869	0.612	1.120	16.376	1.937	9.910

5.7 Traffic Flow Patterns

For D_1 and D_2, we draw decision trees respectively to characterize explainable patterns. The decision tree generated by Graphviz [2] for D_1 is shown in Fig. 5(a) and for D_2 in Fig. 5(b). $(x_1, x_2, \cdots, x_{20})$ refers to CR features in a traffic flow of 1800 s, through sampling the first 10 time of



(a) ROC curve of D_1 with AUC values in bagging

(b) ROC curve of D_1 with AUC values in boosting

Fig. 4. AUC comparison between bagging and boosting.

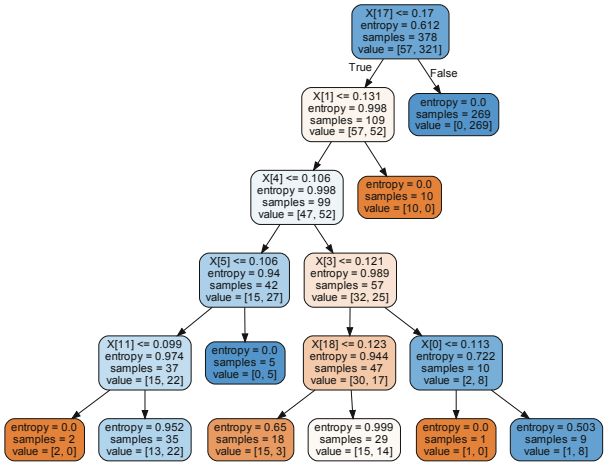
flow head and the last 10 time of flow tail. Therefore, $(x_1, x_2, \cdots, x_{20}) = (CR(t_1), \cdots, CR(t_{10}), CR(t_{1800-9}), \cdots, CR(t_{1800}))$.

Taking Fig. 5(b) for example, the second green rectangle at the bottom level indicates that there are 57 samples as a confidence to support the classification into f_2 class, and the entropy is equal to 0.0. We can found that these patterns are relatively clear and operable. Based on our experiments, if in the first step, we have maximum probability 0.852 to identify the congest attack, and in the second step we have the maximum probability 0.926 to distinguish the five attacks, and we even have about 0.8 probability to find out a specific-frequency congestion attack from a 1800-s flow that is a valuable work as a first step for any randomly-sampled flow analysis.

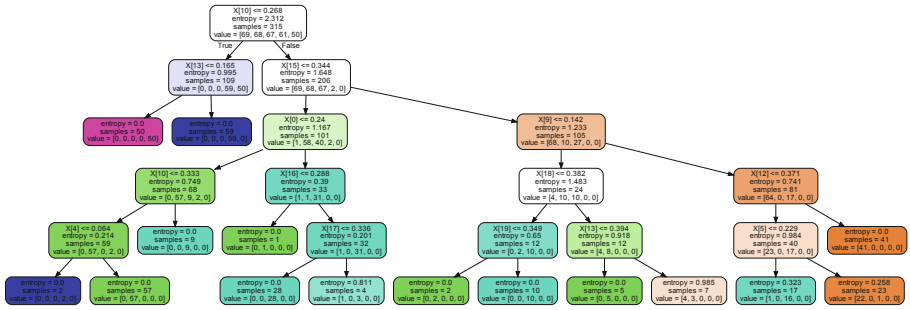
According to the decision trees for D_1 and D_2, Table 12 is drawn to show traffic flow patterns of different classes, $\{Normal, Attack\}$ for D_1 and $\{f_1, f_2, f_3, f_4, f_5\}$ for D_2. The patterns of different classes are counted separately in Table 12, and we can infer the classes from the given sample data. For example, when $X[17] \leq 0.17 \wedge X[1] > 0.131$, we can infer that the sample data belongs to class "Normal", however, $X[17] > 0.17$ is one of the traffic flow patterns of class "Attack". Also, we can see that $X[17]$ is important to distinguish between "Normal" and "Attack", and $X[10]$ is the key feature to distinguish "$f_1 \sim f_5$". In the above analyses, $X[17]$ denotes the attack acceleration of the congestion degree of the 4th phase $\alpha_{PCD}(4)$, $X[1]$ is the global congestion degree for an intersection ICD, an $X[10]$ is the congestion degree of the 5th phase PCD_5. Therefore, the attack acceleration of the congestion degree of the 4th phase and the congestion degree of the 5th phase are the two most essential features for recognizing whether the traffic flow sample is "Normal" or "Attack" and which attack frequency "$f_1 \sim f_5$" the class of the sample is.

Table 12. Class patterns matching based on approximated decision tree for both D_1 and D_2.

Attack classes	Traffic flow pattern
Normal	$X[17] \leq 0.17 \wedge X[1] > 0.131$; $X[17] \leq 0.17 \wedge X[1] \leq 0.131 \wedge X[4] \leq 0.106 \wedge X[5] \leq 0.106 \wedge X[11] \leq 0.099$; $X[17] \leq 0.17 \wedge X[1] \leq 0.131 \wedge X[4] > 0.106 \wedge X[3] \leq 0.121 \wedge X[18] \leq 0.123$; $X[17] \leq 0.17 \wedge X[1] \leq 0.131 \wedge X[4] > 0.106 \wedge X[3] > 0.121 \wedge X[0] \leq 0.113$;
Attack	$X[17] > 0.17$; $X[17] \leq 0.17 \wedge X[1] \leq 0.131 \wedge X[4] \leq 0.106 \wedge X[5] > 0.106$; $X[17] \leq 0.17 \wedge X[1] \leq 0.131 \wedge X[4] \leq 0.106 \wedge X[5] \leq 0.106 \wedge X[11] > 0.099$; $X[17] \leq 0.17 \wedge X[1] \leq 0.131 \wedge X[4] > 0.106 \wedge X[3] > 0.121 \wedge X[0] > 0.113$;
f_1	$X[10] > 0.268 \wedge X[15] > 0.344 \wedge X[9] \leq 0.268 \wedge X[12] \leq 0.371 \wedge X[5] > 0.229$; $X[10] > 0.268 \wedge X[15] > 0.344 \wedge X[9] \leq 0.268 \wedge X[12] > 0.371$;
f_2	$X[10] > 0.268 \wedge X[15] \leq 0.344 \wedge X[0] \leq 0.24 \wedge X[10] \leq 0.333 \wedge X[4] > 0.064$; $X[10] > 0.268 \wedge X[15] \leq 0.344 \wedge X[0] > 0.24 \wedge X[16] \leq 0.288$; $X[10] > 0.268 \wedge X[15] > 0.344 \wedge X[9] \leq 0.268 \wedge X[18] \leq 0.382 \wedge X[19] \leq 0.349$; $X[10] > 0.268 \wedge X[15] > 0.344 \wedge X[9] \leq 0.268 \wedge X[18] > 0.382 \wedge X[13] \leq 0.394$;
f_3	$X[10] > 0.268 \wedge X[15] \leq 0.344 \wedge X[0] \leq 0.24 \wedge X[10] > 0.333$; $X[10] > 0.268 \wedge X[15] \leq 0.344 \wedge X[0] > 0.24 \wedge X[16] > 0.288 \wedge X[17] \leq 0.336$; $X[10] > 0.268 \wedge X[15] > 0.344 \wedge X[9] \leq 0.268 \wedge X[18] \leq 0.382 \wedge X[19] > 0.349$; $X[10] > 0.268 \wedge X[15] > 0.344 \wedge X[9] \leq 0.268 \wedge X[12] \leq 0.371 \wedge X[5] \leq 0.229$;
f_4	$X[10] \leq 0.268 \wedge X[13] > 0.165$; $X[10] > 0.268 \wedge X[15] \leq 0.344 \wedge X[0] \leq 0.24 \wedge X[10] \leq 0.333 \wedge X[4] \leq 0.064$;
f_5	$X[10] \leq 0.268 \wedge X[13] \leq 0.165$;



(a) Decision tree trained based on D_1 for binary classifier.



(b) Decision tree trained based on D_2 for five-class classifier.

Fig. 5. Decision tree trained based on D_1 and D_2, respectively.

6 Conclusion

In this work, we focus on the problem of detection delay and lack of prediction on the I-SIG congestion attack. There are two main challenges, one of which is how to design and extract effective features from traffic flows of variable spoofing frequencies, and the other is how to realize real effective prediction based on ensemble learning, as well as obtain the explainable pattern for detailed analyses. To address the challenges, in this article, we propose a novel approach for congestion attack prediction. We implement ensemble learning to build correlations between abnormal congestion and attack frequency. Based on supervised learning of historical data, we can recognize the current attack frequency and further realize the prediction of the subsequent congestion attack. We also report on necessary and experienced tricks, including rescaling, oversampling, and hyperparameter optimization for performance improvement. Furthermore, we have conducted extensive experiments and analyzed the experimental results in terms

of accuracy and time. The experimental results have shown the superiority of our approach.

To our best knowledge, we are the first to perform and present a prediction approach for periodic I-SIG congestion attack under variable spoofing frequencies. We expect this work guidable for grounding repetition to promote the I-SIG security. This work is also expected to inspire a series of follow-up studies, including but not limited to (1) continuous prediction improvement including both framework and algorithm, (2) more concrete congestion attack prevention by leveraging our insights.

Acknowledgment. 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.

References

1. Connected vehicle applications. https://www.its.dot.gov/pi-lots/cv_pilot_apps.htm
2. Graphvizgraph visualization software. https://graphviz.org/
3. Ptv vissim. http://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim
4. U.s.dot connected vehicle pilot deployment program. https://www.its.dot.gov/pilots/
5. Usdot: Multimodal intelligent traffic safety system (mmitss). https://www.its.dot.gov/research_archives/dma/bu-ndle/mmitss_plan.htm
6. Abdelwahab, M.A., Abdel-Nasser, M., Hori, M.: Reliable and rapid traffic congestion detection approach based on deep residual learning and motion trajectories. IEEE Access **8**, 182180–182192 (2020)
7. Anbaroglu, B., Heydecker, B., Tao, C.: Spatio-temporal clustering for non-recurrent traffic congestion detection on urban road networks. Transp. Res. Part C **48**, 47–65 (2014)
8. Angiulli, F., Astorino, A.: Scaling up support vector machines using nearest neighbor condensation. IEEE Trans. Neural Netw. **21**(2), 351–357 (2010)
9. Boualouache, A., Senouci, S., Moussaoui, S.: A survey on pseudonym changing strategies for vehicular adhoc networks. IEEE Commun. Surv. Tutor. **20**(1), 770–790 (2018)
10. Chen, Q.A., Yin, Y., Feng, Y., Mao, Z.M., Liu, H.X.: Exposing congestion attack on emerging connected vehicle based traffic signal control. In: Network and Distributed System Security Symposium, pp. 39.1–39.15 (2018)
11. Chen, Z., Jiang, Y., Sun, D.: Discrimination and prediction of traffic congestion states of urban road network based on spatio-temporal correlation. IEEE Access **8**, 3330–3342 (2020)
12. Cheng, Q., Varshney, P.K., Arora, M.K.: Logistic regression for feature selection and soft classification of remote sensing data. IEEE Geosci. Remote Sens. Lett. **3**(4), 491–494 (2006)
13. Dimri, A., Singh, H., Aggarwal, N., Raman, B., Ramakrishnan, K.K., Bansal, D.: Barosense: using barometer for road traffic congestion detection and path estimation with crowdsourcing. ACM Trans. Sens. Netw. **16**(1), 4:1–4:24 (2020)

14. Feng, Y., Head, K.L., Khoshmagham, S., Zamanipour, M.: A realtime adaptive signal control in a connected vehicle environment. Transp. Res. Part C. Emerg. Technol. **55**, 460–473 (2015)
15. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. **55**(1), 119–139 (1997)
16. Goin, J.E.: Classification bias of the k-nearest neighbor algorithm. IEEE Trans. Pattern Anal. Mach. Intell. **PAMI-6**(3), 379–381 (1984)
17. Jaleel, A., Hassan, M.A., Mahmood, T., Ghani, M.U., Rehman, A.U.: Reducing congestion in an intelligent traffic system with collaborative and adaptive signaling on the edge. IEEE Access **8**, 205396–205410 (2020)
18. Jeske, T.: Floating car data from smartphones: what google and waze know about you and how hackers can control traffic (2012)
19. Li, Q., Tan, H., Jiang, Z., Wu, Y., Ye, L.: Nonrecurrent traffic congestion detection with a coupled scalable bayesian robust tensor factorization model. Neurocomputing **430**, 138–149 (2021)
20. Li, Y., et al.: An empirical study on gan-based traffic congestion attack analysis: a visualized method. Wirel. Commun. Mob. Comput. **2020**, 8823300:1–8823300:14 (2020)
21. Lu, Z., Qu, G., Liu, Z.: A survey on recent advances in vehicular network security, trust, and privacy. IEEE Trans. Intell. Transp. Syst. **20**(2), 760–776 (2019)
22. Bartlett, P., Freund, Y., Lee, W.S., Schapire, R.E.: Boosting the margin: a new explanation for the effectiveness of voting methods. Ann. Stat. **26**(5), 1651–1686 (1998)
23. Sen, S., Head, K.L.: Controlled optimization of phases at an intersection. Transp. Sci. **31**(1), 5–17 (1997)
24. Suarez, A., Lutsko, J.F.: Globally optimal fuzzy decision trees for classification and regression. IEEE Trans. Pattern Anal. Mach. Intell. **21**(12), 1297–1311 (1999)
25. Sun, X., Xu, W.: Fast implementation of delong's algorithm for comparing the areas under correlated receiver operating characteristic curves. IEEE Signal Process. Lett. **21**(11), 1389–1393 (2014)
26. Ta, V., Dvir, A.: A secure road traffic congestion detection and notification concept based on V2I communications. Veh. Commun. **25**, 100283 (2020)
27. Wang, R., Xu, Z., Zhao, X., Hu, J.: V2v-based method for the detection of road traffic congestion. IET Intell. Transp. Syst. (2019)
28. Wang, X., Xiang, Y., Niu, W., Tong, E., Liu, J.: Explainable congestion attack prediction and software-level reinforcement in intelligent traffic signal system. In: 26th IEEE International Conference on Parallel and Distributed Systems, ICPADS 2020, Hong Kong, 2–4 December 2020, pp. 667–672. IEEE (2020)
29. Wolpert, D.: Stacked generalization. Neural Netw. **5**, 241–259 (1992)
30. Zeadally, S., Hunt, R., Chen, Y., Irwin, A.S.M., Hassan, A.: Vehicular ad hoc networks (vanets): status, results, and challenges. Telecommun. Syst. **50**(4), 217–241 (2012)
31. Zhang, N., Zhang, Y., Lu, H.: Seasonal autoregressive integrated moving average and support vector machine models: prediction of short-term traffic flow on freeways. Transp. Res. Rec. **2215**(1), 85–92 (2011)
32. Zhong, X., Li, L., Zhang, Y., Zhang, B., Zhang, W., Yang, T.: Oodt: obstacle aware opportunistic data transmission for cognitive radio ad hoc networks. IEEE Trans. Commun. **68**(6), 3654–3666 (2020)