Transportation Research Part C 149 (2023) 104058

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Online transportation network cyber-attack detection based on stationary sensor data

Ruixiao Sun [a], Qi Luo [b,*], Yuche Chen [a,*]

[a] Department of Civil and Environmental Engineering, University of South Carolina, 300 Main Street, Columbia, 29201, SC, USA
[b] Department of Industrial Engineering, Clemson University, 277B Freeman Hall, Clemson, 29634, SC, USA

ARTICLE INFO

Keywords:
Cyber-attack detection
Transportation network
Sensor location
Machine learning

ABSTRACT

Penetration of connected vehicles and crowdsourced mapping applications give rise to security vulnerabilities in transportation networks. Accurate detection of cyber-attacks on transportation networks is critical to minimize impacts on transportation systems. This task is particularly challenging because the impacts of regional cyber-attacks can be invisible on aggregated traffic data, especially when only sensor data is accessible to transportation agencies. We propose an analytical framework that leverages real-time road link sensory data to conduct online data-driven transportation network anomaly detection using non-parametric long short-term memory (LSTM) and parametric Gaussian process model. The online anomaly detection models can continuously update model coefficients as real-time sensory data arrives. We utilize a city-scale microscopic traffic simulation to validate our cyber-attack detecting framework. The cyber-attack detection model achieves a F_1 score, which is a harmonic mean of the precision and recall of classifiers, between 84% to 96% considering different initial training data sizes. We compare with major offline models to demonstrate the effectiveness and robustness of online models. In addition, we devised a meta-heuristic method to solve the multi-objective sensor location problem to simultaneously enhance anomaly detection efficiency and maximize traffic information gain. This study demonstrates a systematic approach to address the emerging concerns of cyber-security in transportation networks with minimum requirements for infrastructure upgrades. Our results can help transportation security authorities identify potential cyber-attacks and protect transportation infrastructure from malicious cyber-hackers.

1. Introduction

1.1. Background

Modern transportation technologies enable data sharing across infrastructure systems and on-road vehicles, allowing travelers to make informed travel decisions based on dynamic traffic data. These data sources serve as the basis for a multitude of intelligent transportation applications. Dynamic toll pricing and congestion charging zone are powered by algorithms that determine the charges based on the observed level of traffic congestion (Figueiras et al., 2019). Municipalities use these data to predict emergencies, implement adaptive signaling, and send alerts to drivers. However, security threats and cyber-attacks, are becoming a growing concern for the entire transportation system. Transportation is one of the most targeted industries, with each incident costing an

[*] Corresponding authors.
E-mail addresses: qluo2@g.clemson.edu (Q. Luo), chenyuc@cec.edu.cn (Y. Chen).

https://doi.org/10.1016/j.trc.2023.104058
Received 2 March 2022; Received in revised form 13 September 2022; Accepted 11 February 2023
Available online 24 February 2023

estimated $3.58 million in 2020 (Lee et al., 2020). The points of vulnerability include the Controller Area Network (CAN) bus, Electronic Control Units (ECU), V2X communication channels, backend infrastructure, and crowd-sourcing mapping services.

Crowd-sourcing mapping applications, such as Google Maps, can use periodic GPS data to infer driving speed and congestion levels on roadways, and the related real-time projections of traffic conditions have played a significant role in facilitating route selection and navigating travelers. A recent survey showed that 93% of the respondents from the 20 cities with the most cars per capita in the US were dependent on GPS navigation and one in five use it every day (CarPro, 2022). More broadly, idle taxi and ride-hailing drivers use crowd-sourcing data to search for the next passengers. Emerging Autonomous Driving System (ADS) utilizes shared location information as input for assisting self-driving cars.

However, these systems that depend on crowd-sourced data are naturally vulnerable to mischievous or malicious attackers trying to disrupt or game the system (Stefanovitch et al., 2014). The most critical thread is GPS spoofing. For transportation networks, these attacks are possible because the location signal is unencrypted. There are few effective tools to authenticate whether the origin of traffic requests is real mobile devices or software scripts, which have become a credible threat to the public. According to a 2020 report, a Berlin artist used 99 cellphones and a handcart to deceive Google Maps users into believing they were stuck in a traffic jam that did not exist (CNN, 2020). Recently, hackers exploited the Russian ride-hailing app that maliciously paralyzes traffic in Moscow (MSN, 2022). Researchers discovered that an increase in the number of ghost car entry locations on highway segments could cause major traffic congestion (Huang et al., 2021). An experiment to generate "ghost riders" was conducted through the reverse engineering of a crowd-sourcing mapping software's communication protocol with the server (Wang et al., 2018b). It effectively caused the application to report fictitious traffic jams and accidents and automatically reroute user traffic. This study also found that the proposed inferred function can accurately forecast the road's traffic speed for a variety of combinations of fictitious slow-speed vehicles and real fast-speed vehicles.

This research assumes that cyber-attacks aim to serve a small group of travelers, called "hackers" for lack of a better term, and the other road users "normal travelers". In this research, we dismiss hackers' intention of intrusion but consider a general type of malicious cyber-attack that the modification of broadcasting traffic information can benefit hackers (Kerns et al., 2014). Inspired by the ghost rider example, this work uses an attack model in which the hackers can configure a fleet of virtual, low-speed vehicles on chosen road segments, so as to change the road conditions displayed on trip planners (e.g., cloud-sourcing maps). As shown in Fig. 1, these virtual vehicles can indirectly spoof the perceived travel time and simulate network-level traffic congestion. However, the attacker does not have the power to intrude the planner directly, hence it is a weaker adversarial model than other trajectory data and traffic anomaly detection literature (Laxhammar and Falkman, 2013; Ji et al., 2020). The travel time (or cost) on the map can only be manipulated to a certain level with the complex interaction between the created virtual vehicles and normal driving vehicles that formulates the network equilibrium. More specifically, normal travelers who rely on potentially contaminated data to determine travel routes will avoid traversing links on hackers' routes in order to reduce travel disutility. Therefore, hackers can enjoy traveling on routes with significantly reduced congestion.

Detecting the above system-level cyber-attacks is different from detecting cyber-attack at an individual vehicle (Bhuyan et al., 2013; Garcia-Teodoro et al., 2009; Li et al., 2018), connected vehicle systems (Comert et al., 2021), or a smart intersection (Huang et al., 2021; Feng et al., 2022), which has been widely studied in the literature. At the transportation system level, it is challenging to detect cyber-attacks on certain roads or from a small number of hackers, in a real-time manner, based on a large volume of trip data from all vehicles on the whole network.

Most cyber-attack detection methods in transportation systems are based on trajectory-level data collected from probe vehicles (Laxhammar and Falkman, 2013; Oh and Iyengar, 2019; Wang et al., 2016). However, they are unlikely to be widely deployed due to the following reasons. First, probe vehicle data with sufficient granularity is typically generated by third-party probe vendor companies, which are considered confidential for public use and limit the application of cyber-attack detection. Second, the volume of data provided by the probes may not be adequate to represent the entire fleet at a specific road segment, especially during off-peak hours (Mudge et al., 2013). Third, these third-party probe vehicle data are usually provided to transportation agencies in an offline manner, hence they cannot facilitate real-time anomaly detection.

With this consideration, the more pragmatic solution is to utilize the easily accessible and more reliable data collected from stationary sensors to detect possible cyber-attacks. These sensors include intrusive road sensors (e.g., magnetometers, inductive loops), radar sensors, and video cameras that are used for speed and flow rate measurements (Guerrero-Ibáñez et al., 2018). This research's primary goal is to design anomaly detection models to identify the changes in traffic patterns only based on sensory traffic data and recognize the occurrence of system-level cyber-attack-related events with a virtual-vehicle attacker model. The detection model with access to restricted observations from stationary sensors is distinct from the zero-sum-game-based anomaly detection models (Laszka et al., 2019). The stationary sensor such as the inductive loop detector plays dual roles in transportation systems: (1) gathering traffic data (i.e., flow and speed) for planning purposes to support traffic management applications, and (2) continuously monitoring traffic conditions for security requirements. It is intellectually interesting to optimize the utilization of existing traffic sensors in collecting information to simultaneously improve the efficiency of cyber-attack detection and enhance traffic information gain for traffic management.

This study proposes a machine-learning (ML) based framework that leverages real-time road link sensory data to conduct online data-driven transportation network anomaly detection. Additionally, we address the dual role of stationary sensors in monitoring normal traffic conditions and detecting anomalies by developing a meta-heuristic method for near-optimal network sensor locations. The objective is to simultaneously improve anomaly detection effectiveness and maximize traffic information gain. We utilize a city-scale microscopic traffic simulation to validate our cyber-attack detecting framework. The results can help transportation security authorities identify potential cyber-attacks and protect transportation infrastructure from malicious cyber-hackers. The main

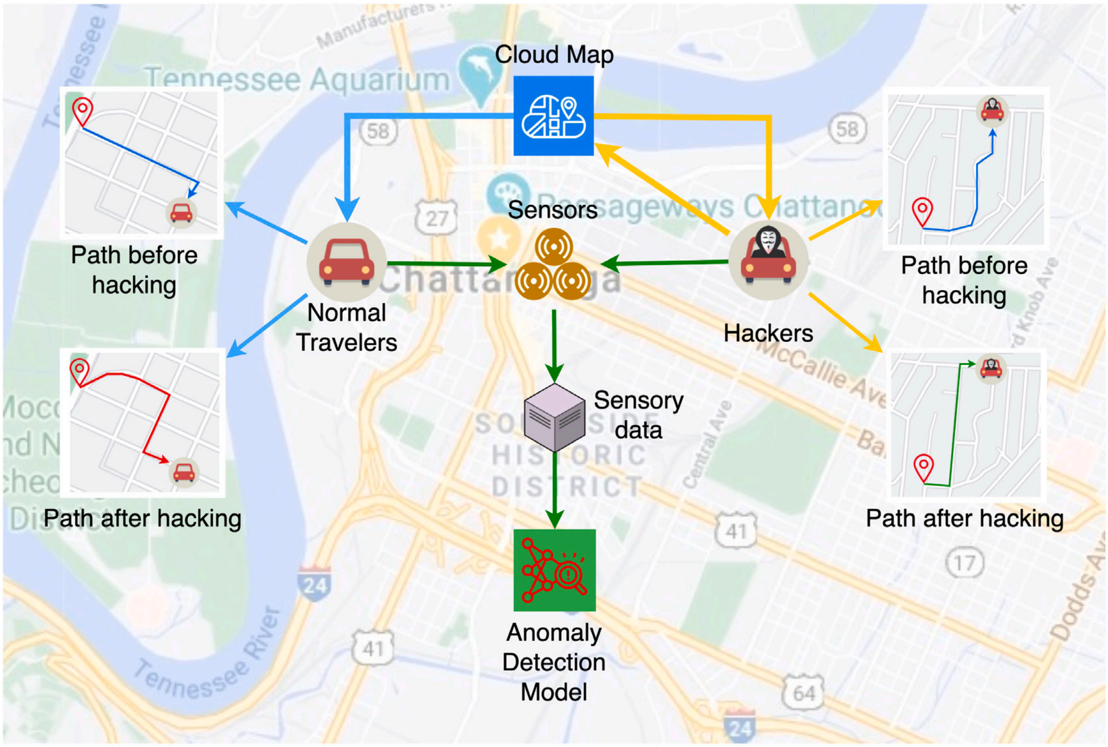


Fig. 1. An example of a cyber-attack on transportation networks; the orange arrows denote the manipulation of hackers, the blue arrows present the behavior of normal travelers, and the green arrows show the process of detectors.

contributions of this study are: (a) only roadway sensory data is utilized for detecting transportation network anomalies; (b) two transportation network anomaly detection models (i.e., parametric-based model and nonparametric-based model) are investigated and compared for online cyber-attack detection; (c) the detection framework is to identify the abnormal traffic patterns and recognize the cyber-attack-related events in network level; (d) a multi-objective optimization along with a meta-heuristic algorithm are developed for sensor network expansion plan to improve detection efficiency and maximize traffic information; (e) a city-scale microscopic traffic simulation based on real O-D matrix data is leveraged to verify the proposed models.

The remainder of this paper is organized as follows. Section 2 reviews the related literature in cyber-attack detection with a focus on transportation applications. Section 3 describes the cyber-attack-related anomaly detection problem and introduces the data. Section 4 discusses methodologies and provides numerical results for real-time cyber-attack-related anomaly detection in traffic networks. Section 5 presents a traffic sensor location optimization method to simultaneously improve the effectiveness of traffic anomaly detection s and enhance traffic information gain. Section 6 concludes the paper and points out future research directions.

2. Related work

We review relevant literature on anomaly detection in transportation fields, which is summarized in Table 1.

2.1. Anomaly detection in transportation networks

In general, anomaly detection applications in transportation can be categorized into specific transportation elements and network levels. Furthermore, the methods used to do anomaly detection can be divided into rule-based (physical-based) or ML-based (data-driven). Element level anomaly detection focuses on utilizing operational data of an element (e.g., vehicle, road segment) to classify if that element is under cyber-attack or not. Network-level anomaly detection investigates whether the traffic flow pattern at the network level is abnormal and is subjected to certain cyber-attack events. Physical-based methods use specific closed-form traffic/vehicle dynamics formulas to evaluate whether vehicle or traffic flow behavior is expected. Data-driven methods leverage ML techniques to detect changes in traffic patterns and evaluate whether cyber-attack-related events cause abnormal patterns.

For element-level anomaly detection studies, Li et al. (2021) developed an information entropy-based anomaly detection method to determine whether the message broadcast in a vehicle's CAN bus is spoofed. The model utilized data obtained from certain in-vehicle sensors to construct an information entropy index and compared it with the message in CAN bus to determine whether a vehicle is compromised. Similar to in-vehicle sensor data, other studies employed data-driven ML models such as the auto-encoder

Table 1
Summary of anomaly detection methods in transportation applications.

Anomaly level	Anomalous event	Data source	Method		Paper
Local anomalies	Vehicle control intrusion	In-vehicle sensors	Offline	Rule-based	Li et al. (2021)
				ML	Bouzeraib et al. (2020), Oucheikh et al. (2020), Berger et al. (2018) Khan et al. (2020)
	Road segment intrusion	Traffic data and GPS	Offline	Rule-based	Thajchayapong and Barria (2010)
				ML	Chawla et al. (2012), Boquet et al. (2020)
	Abnormal driving behavior	Traffic features data	Offline	Rule-based	Bawaneh and Simon (2019), Nguyen et al. (2016)
				ML	Shirazi et al. (2016), Ngan et al. (2015), Dang et al. (2015)
Network anomalies	Spoofing trajectories	Vehicle trajectory data	Offline	Rule-based	Wang et al. (2016), Dias et al. (2020)
				ML	Laxhammar and Falkman (2013), Oh and Iyengar (2019)
		Traffic feature data	Offline	ML	Münz et al. (2007), Davis et al. (2020)
		Smart card	Offline	ML	Wang et al. (2018a)
		Stationary data	Offline	ML	Zhang and Paschalidis (2017)
			Online	ML	This research

convolutional neural network (Oucheikh et al., 2020) and generative adversarial network (Bouzeraib et al., 2020) for anomaly detection. Thajchayapong and Barria (2010) utilized road-link level traffic performance metrics, i.e., average speed, vehicle space, and compared with traffic fundamental identify functions to detect abnormal events on road links. Other researchers applied data-driven methods to detect cyber-attacks on road links, such as clustering-based data mining (Chawla et al., 2012) and variational autoencoder model (Boquet et al., 2020).

Transportation-network-level anomaly detection studies used vehicle trajectory data to reconstruct road network flow patterns and then used traffic fundamental identify functions to determine whether there are vehicle trajectories are spoofed (Wang et al., 2016; Dias et al., 2020). Laxhammar and Falkman (2013) and Oh and Iyengar (2019) also utilized vehicle trajectory data and applied various deep Bayesian learning methods to detect trajectory spoofing attacks. Other studies aimed to detect irregular driving behavior in transportation networks using ML techniques with various types of data sources, such as link-level flow data (Münz et al., 2007; Davis et al., 2020), public transit ridership data (Wang et al., 2018a), and traffic sensory data (Zhang and Paschalidis, 2017).

Although there is a school of literature on cyber-attack-related anomaly detection, most studies focus on anomaly detection using individual trajectory data and other static traffic information. Real-time anomaly detection leveraging sensory monitoring data instead of narrowly accessible trajectory data is necessary for fast response and defense against cyber-attacks considering the dynamics of anomalous events. This research aims to fill the knowledge gap by proposing an online data-driven transportation network anomaly detection framework.

2.2. Sensor location optimization in transportation networks

Since the proposed online cyber-attack detection model utilizes data from stationary traffic sensors, the spatial distribution of sensors is critical in determining the effectiveness of detection models. The sensor location problem is widely studied in the literature. Most of the existing studies used various methods to optimally deploy traffic sensors in order to maximize traffic information. For example, Fei and Mahmassani (2011) proposed the Kalman filtering method to find the sensor locations that maximize information gains of the travel origin and destination demand estimation in transportation networks. Zhou and List (2010) achieved a similar objective in origin–destination estimation with a scenario-based stochastic optimization procedure. Other studies developed optimal sensor location determination algorithms considering uncertainty, such as sensor measurement error or sensor failure (Fei et al., 2013; Xu et al., 2016; Salari et al., 2019). However, to the best of our knowledge, no existing study focuses on network sensor location for balancing the cyber-attack event detection efficiency and regular traffic information gain. While traffic anomaly detection using traffic sensory data has been studied extensively, given the spatial and temporal heterogeneity of traffic conditions on road links, it is challenging to determine the sensor locations that can provide sufficient traffic data to simultaneously enhance the observability of transportation systems and the security against cyber-attacks. To fill this gap, we develop and solve a multi-objective optimization model for traffic sensor expansion plans.

3. Problem description and data preparation

3.1. Network anomaly detection and sensor location problem

This paper designs online data-driven anomaly detection methods for real-time transportation network cyber-attack analysis and develops an optimization model for network sensor location selection. Assuming a traffic network represented by $G = (V, E)$, where V is the set of vertices and E is the set of edges. The O-D matrix λ_{od} provides non-atomic flows of travelers drive from their origins $o \in V$ to destinations $d \in V$ on the shortest routes at equilibrium. We suppose some hackers, which are an unknown subset of travelers denoted as λ_{od}^{H}, intrude into the cloud sourcing map and generate slow speed virtual vehicles to distort traffic conditions. We assume the cloud sourcing map is publicly available and used by normal drivers in the network with O-D matrix λ_{od}^{N}. We assume all drivers use the standard dynamic user equilibrium algorithm to determine their route over a fixed horizon $t \in [0, T]$. Furthermore, we can divide the edges set E into edges E^{H}, including all edges on hackers' routes and edges E^{N} for others. From hackers' points of view, they will concentrate on spoofing the traffic conditions of edges E^{H} on the cloud sourcing map, hoping to gain exclusive right-of-way on those edges. One possible scenario for the cyber-attack intrusion of traffic data and the effect of spoofing through dynamic user equilibrium traffic assignment is illustrated in Appendix A. Transportation authorities typically are unaware of cyber-attacks on cloud-sourcing maps, but they have real-time traffic volume data based on sensors placed on various locations of transportation infrastructure. The problem we are to solve in this study is how to efficiently and promptly identify cloud-sourcing map cyber-attack events based on real-time and historical traffic monitoring data $D_S(t)$. The anomaly detection system only accesses sensory data at fixed locations at each time t, which is termed the transportation network anomaly detection process.

Additionally, once the anomaly detection method is developed, the next task is to solve the sensor location problem to get more traffic information and implement network anomaly detection most efficiently. Assume the network has n existing sensors installed at edges E_S ($E_S \subseteq E$) that are useful for transportation system observation and monitoring. To improve the performance of network anomaly detection algorithm as well as maximize traffic information, the locations of additional sensors are needed to be selected from the remaining edges $E_{\bar{S}}$ ($E_{\bar{S}} \subseteq E$). We formulate the sensor allocation problem as a bi-objective optimization that aims at simultaneously improving the efficiency of cyber-attack detection simultaneously and maximizing traffic information gain for general usage. In addition, we apply a meta-heuristic algorithm to find the optimal allocation of the road sensors.

3.2. Data preparation

We prepare city-scale traffic simulation data to train, develop, and validate the network anomaly detection framework. The traffic simulation uses an open-source microscopic traffic simulator SUMO (Haklay, 2010). The O-D matrix and road network are provided by Chattanooga Metropolitan Planning Organization in Tennessee based on data collection, survey, and traffic analysis in a real-world setting. The SUMO simulation platform and Chattanooga traffic data have been used in previous studies (Sun et al., 2021).

In the city-wide traffic simulation for Chattanooga, Tennessee, this project assumes that cyber-attack events occur randomly and each event occurs in unpredictable hacking patterns in which hackers arbitrarily intrude on different road segments. For each cyber-attack, hackers continuously introduce slow-driving virtual vehicles to alter the traffic conditions displayed on the crowdsourcing map and then generate fictitious congestion on the routes they intend to use. We randomly select a portion of O-D demand as hackers' travel demand and subsequently identify hackers' routes, see red road links in Fig. 2. Hackers will spoof roadway travel time in the cloud-sourcing map to discourage normal drivers from traveling on those routes. Specifically, the spoofing process will artificially and largely increase travel time on links within hackers' routes. Since normal drivers use travel time from the cloud sourcing map to determine routes based on dynamic user equilibrium, they will avoid using links of hackers' routes. Then, hackers will be able to gain exclusive or nearly exclusive right-of-way on their routes. Traffic sensors are placed on selected road segments across the network to obtain average traffic flow and speed in the form of time-series data. With the above setup in traffic simulation, we are able to simulate a whole day of vehicle movements in Chattanooga City. The results of the traffic simulation include time-space trajectories for each vehicle at 1 Hz resolution as well as 5-min or 1-h aggregated traffic volume on each road link during the whole simulation time period. The results enable us to assess the impacts of cyber-attacks on individual vehicles' delays as well as changes in aggregated traffic patterns on road links.

As illustrated above, hackers who are not able to directly intrude the back-end of a cloud sourcing map can still change the displayed link-level traffic conditions and increase the perceived level of traffic congestion. Since the ratio of hacking travel demand and the extent of adversarial events can vary, the simulation uses an equivalent hacking strategy by converting the addition of virtual vehicles on random road links to increase the travel time and congestion level on randomly intruded routes. We test simulations by

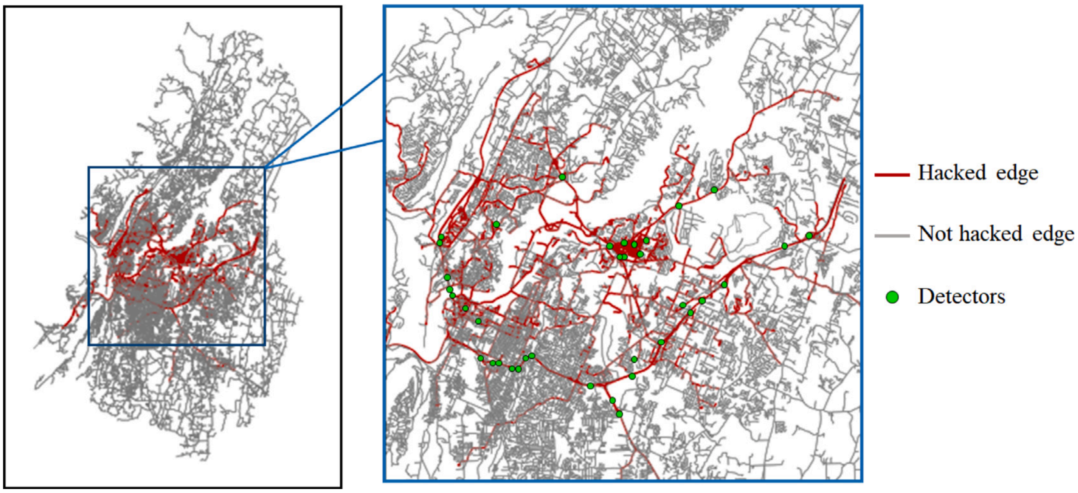


Fig. 2. Road network map in Chattanooga, TN with detectors and hacked edge illustration.

increasing travel time on hackers' links ranging from 30% to 300% and by having a hacking O-D demand ratio ranging from 0.5% to 5%.

Fig. 3(a) reports the average travel time for hackers, non-hackers, and all travelers with a 1 percent hacking O-D demand ratio. We observe that the reduction in the average travel time for hackers increases as the travel time increase ratio on hacked edges increases from 30% to 300%. Hackers can save approximately 3.9 min, 2.8 min, and 0.2 min on average if they manipulate the hacking edge travel time increase ratio by 300%, 100%, and 30%, respectively. As the ratios increase from 30% to 300%, the changes of non-hackers average travel time increase from 0.15 min at 30% to 2.9 min at 300%. Hackers can benefit substantially from spoofing data, which degrades other road users' utility. In addition, spoofing edge travel time on a small proportion of edges can significantly harm the network's overall performance.

Fig. 3(b) depicts the changes in average travel time for hackers, non-hackers, and all travelers as a function of the percentage of hackers in the fleet. The baseline scenario is the unhacked networks. The percentage of hackers ranges from 0.5% to 5% in the simulation, with the fixed hacking edge travel time increase ratio of 100%. For hackers, the average travel time per hacker is reduced by about 2.5 min when 0.5% of the fleet are hackers, and the reduction of the average travel time increases to 2.8 min when the hackers are 1% of the fleet. However, with the hack percentage increasing from 1% to 5%, the hackers' average travel time decreases to 0.8 min at 2.5% hacked and decreased to 0.3 min at 5%. Similarly, the changes in average travel time of non-hackers increase as this percentage increases from 0.5% to 2.5% but goes down at 5%. The overall travel time has a similar trend as non-hackers. The largest changes in the average travel time are achieved when hackers are 2.5% of all vehicles.

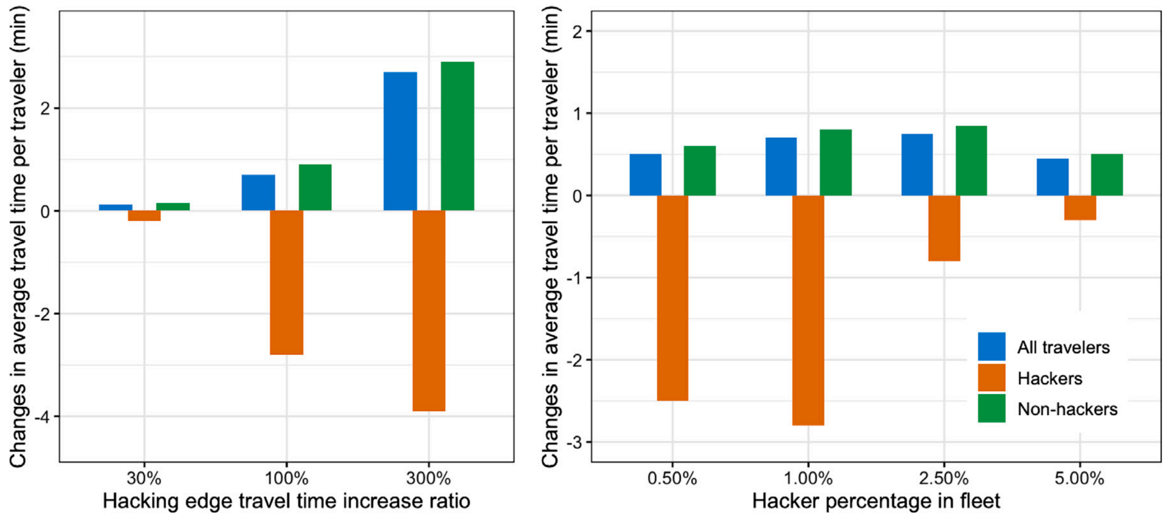
The outputs from the traffic simulations include link-level aggregated traffic volume at a time period between 5-min to 1-h. This information is normally available to transportation authorities through traffic monitor sensors installed on a road network. In the following sections of this paper, we are to use these sensor-generated link-level traffic volume data to construct data-driven methods for traffic network cyber-attack anomaly detection as well as optimal sensor location determination.

4. Online transportation network anomaly detection

This section introduces the transportation network anomaly detection framework and validates it with the simulation for its effectiveness. Specifically, we are to develop a framework for online detecting of transportation network-level cyber-attack anomaly events with dynamically updated traffic information.

Two methods that utilize historical traffic data collected from traffic sensors are capable of dynamically incorporating real-time traffic data into an online decision process. The first method is a nonparametric technique using LSTM-autoencoder that can be trained with extensive one-class data (i.e., normal traffic data only) to achieve high detection performance. The second method is a parametric technique using Gaussian processes to model the dynamic traffic data collected from sensors. Compared to the former method, it can be trained with a small size of historical traffic data, so it is suitable for the early deployment stage, whereas the historical traffic data must contain intrusion records.

We shall begin with defining the notation for the data collection on the fly. Our goal is to detect whether the transportation network is hacked or not based on a batch of traffic data sequentially collected over a horizon $t = 1, \dots, T$. The traffic data are denoted as $D_S^{[T]} := (D_S(1), \dots D_S(T))$, where S means that the traffic data are from on links with deployed sensors. $D_S(t)$ contains link-level flow and average speed information on m pre-selected road links in the transportation network. Thus, each $D_S(t) = (q_1^t, \dots, q_m^t; v_1^t, \dots, v_m^t)$ is a vector with $2m$ entries. Notice that these sensors can locate either on edges covered by hackers' routes, E^H, or other routes, E^N.



(a) **(b)**

Fig. 3. Changes in average travel time for hackers, non-hackers, and all travelers due to spoofing trajectories: (a) impacts of anomaly levels (b) impacts of hacker percentage.

4.1. *Nonparametric LSTM-autoencoder model*

The LSTM-autoencoder (Fig. 4) is a self-training classifier that works as follows. At each time step t, it determines whether the newly collected traffic data $D_S(t)$ shows signs of cyber-attacks on a transportation network. Each $D_S(t)$ will be process through a transformation and convert it to $D'_S(t)$. This is the so-called LSTM encoder–decoder process which is the integration of the Autoencoder model and LSTM model, where the Autoencoder is an artificial neural network that transforms the original one-class traffic data to a vector that best represents its characteristics (Ai et al., 2021; Ashraf et al., 2020), and the LSTM blocks are served as the hidden encoder and decoder layers to handle sequential traffic data. The reconstruction error is represented by the mean absolute error (MAE) $\|D_S(t) - D'_S(t)\|$, which measures the difference between the original and transformed traffic data vectors and is minimized in the training process. Next, the online detection algorithm dynamically searches a threshold value from the MAEs based on the normal traffic data obtained before period t, i.e., $D_S^{[t-1]}$. Since the reconstruction errors determine the distance of the unknown traffic data vector to the normal traffic data vector, the LSTM-autoencoder can determine whether or not the recently collected traffic data $D_S(t)$ is contaminated or attacked based on the updated threshold. The data vectors with reconstruction errors exceeding the threshold value will be labeled as *hacked* events. More specifically, the threshold ϕ for period t are set from the distribution of MAEs of training data from 1 to $t-1$, and the model will classify $y(t) = 1$ for anomaly events if $\|D_S(t) - D'_S(t)\| > \phi$ or $y(t) = 0$ otherwise. For samples labeled as normal events, $D_S(t)$ will be added to the end of the historical data $D_S^{[t-1]}$. This self-training process will continue determining transformation and threshold for anomaly detection in period $t+1$.

4.2. *Parametric Gaussian processes model*

Training an LSTM-autoencoder will require extensive traffic data, which is unavailable under certain circumstances. An alternative method is using Gaussian processes, which is a stochastic process assuming the collection of $2m$-dimensional sensory data follow multivariate normal distributions (Williams, 2006), to model the data generation of $D_S^{[t]}$. Here each traffic data $D_S(t)$ is parametrized by $\mu_t \in \mathbb{R}^{2m}$, the mean vector with $2m$ elements, and ϕ_t, a $2m$-by-$2m$ matrix, as the covariance matrix. The parameters $\theta_t = (\mu_t, \phi_t)$ are continuously calculated based on historical traffic data collected up to period t as $D_S^{[t-1]}$. The log-likelihood is given by:

$$\ell(y_t|D_S^{[t]}) = -\frac{1}{2}(D_S(t) - \mu_t)^T \phi_t (D_S(t) - \mu_t) - \frac{1}{2}\log det(\phi_t) - \frac{t}{2} log(2\pi). \tag{1}$$

Zhu et al. (2021) proposed a min–max formulation for the adversarial anomaly detection problem where the detector's objective is to maximize the above log-likelihood. The main result is that a threshold-based policy is optimal for detecting the anomaly. As mentioned before, the attacker only can indirectly manipulate the traffic conditions displayed on the trip planner by configuring virtual vehicles. Thus, it is a weaker adversarial model than other adversarial anomaly detection models in the literature. Here, the Gaussian process model aims to track the above log-likelihood in a real-time manner and detect the attack with a dynamic threshold. Let η_t denote a threshold value such that the system raises an alarm at time t if

$$\ell(y_t|D_S^{[t]}) > \eta_t, \quad \eta_t \propto \mathbb{E}[y_t|D_H^{[t]}], \tag{2}$$

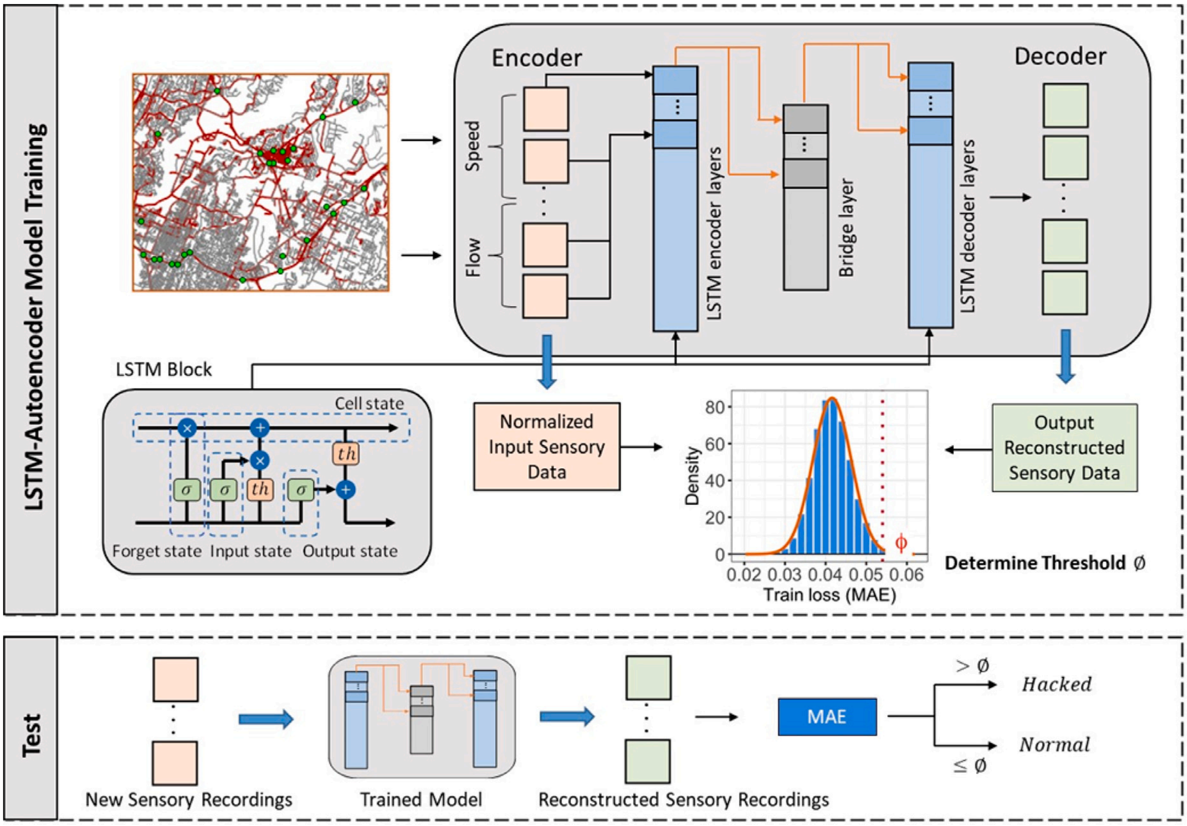


Fig. 4. The proposed LSTM-autoencoder anomaly detection structure.

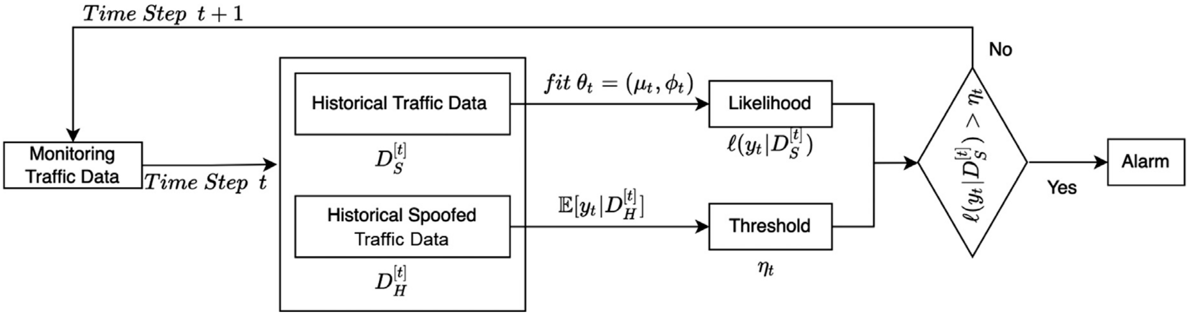


Fig. 5. The procedure of parametric Gaussian process anomaly detection model.

where $D_H^{[t]}$ is the historical traffic data indirectly manipulated by hackers. The procedure of the parametric Gaussian process anomaly detection model is presented in Fig. 5.

The limitation of the Gaussian processes is that the likelihood function is calculated based on historical traffic data that contains both hacking and normal traffic events. Since the model will raise the cyber-attack alarm if the calculated likelihood exceeds the threshold, the fitting of θ_t will terminate thereafter. Otherwise, the model will add $D_S(t)$ to the historical traffic data set in order to estimate mean and variance parameters for the next time step $t+1$. The threshold value η_t decreases as the log-likelihood is negative, and the diminishing rate is of $O(\sqrt{t})$. The only additional parameter to be tuned is the initial threshold value η_0.

4.3. Comparative numerical study of anomaly detection models

As described in Section 3.2 regarding the data preparation, we use the simulated dataset to train, test, and validate the offline anomaly detection models and the proposed online anomaly detection models in a real-time manner.

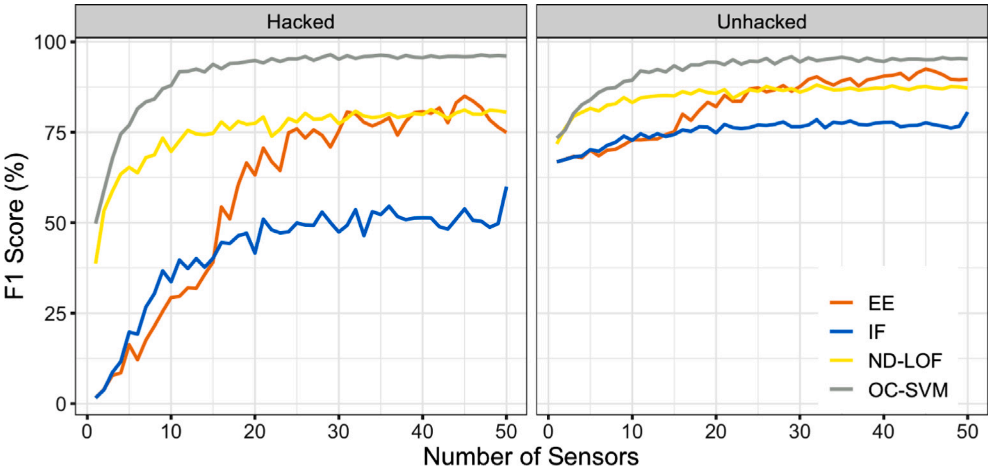


Fig. 6. Performance of offline anomaly detection models based on different number of sensors.

Here, we define the following evaluation metrics in the comparative study of offline and online anomaly detection models. This proposed model is to determine whether the traffic network is under cyber-attack at each timestamp, t. The actual traffic network status at t, which is unobservable for the detection model, is $Y_t = \{0 : normal, 1 : hack\}$. The detection model identifies the traffic network status at t as $y_t = \{0 : normal, 1 : hack\}$. The true positive variable TP_t is 1 if $y_t = 1|Y_t = 1$ (the traffic network is under cyber-attack and our model correctly detects it); The false positive variable FP_t is 1 if $y_t = 1|Y_t = 0$ and false negative variable FN_t is 1 if $y_t = 0|Y_t = 1$. We further define precision and recall as:

$$Precision = \frac{\sum_t TP_t}{\sum_t TP_t + \sum_t FP_t}, \quad Recall = \frac{\sum_t TP_t}{\sum_t TP_t + \sum_t FN_t}, \tag{3}$$

where precision is the accuracy ratio when the model classifies a cyber-attack event, and the recall is the probability that the model classifies the cyber-attack event correctly. Literature shows both $Precision$ and $Recall$ are important metrics in quantifying the accuracy of anomaly detection models (Shirazi et al., 2016; Wang et al., 2016). A model with high precision normally has a low recall and vice versa (Buckland and Gey, 1994). Finally, we use a measurement based on (3), termed as $F1$ score, to comprehensively quantify the accuracy of the proposed models (Li et al., 2018):

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{4}$$

$Precision, Recall$, and the $F1$ score are all between zero and one, and a higher $F1$ score indicates that the performance of the detection model is more powerful. In addition, the false positive rate evaluates the limitation of an anomaly detection model's detection capability (Al Jallad et al., 2020). We define the false positive rate, which is the ratio of normal traffic being falsely classified as cyber-attacks, as:

$$False\ Positive\ Rate = \frac{\sum_t FP_t}{\sum_t FP_t + \sum_t TN_t}. \tag{5}$$

In the remainder of this paper, we use the $F1$ score as the accuracy metric and false positive rate as limitation metric in evaluating the nonparametric and parametric detection models; it is worth mentioning that there are other information-based metrics for online detection algorithms (Oh and Iyengar, 2019), but they are not intuitive in practice.

4.3.1. Sensitivity analysis of sensor number

To have a better knowledge of how the number of sensors influences the performance of anomaly detection models, we conduct a sensitivity analysis on sensor numbers using offline anomaly detection models. These results will guide the sensor location selection problem in Section 5. The four offline classification models are: one-class support vector machine (OC-SVM) (Wang et al., 2004), isolation forest (IF) (Liu et al., 2012), novelty detection with local outlier factor (ND-LOF) (Breunig et al., 2000), and elliptic envelope (EE) (Rousseeuw and Driessen, 1999). These four models are classical offline classification models and are widely used in literature. For completeness, we provide a description of the four models in Appendix B. In this analysis, we vary the number of sensors from 1 to 50, which are randomly selected from a set of candidate locations. To eliminate randomness in the results, we replicate the random selection and simulations 50 times for a fixed number of sensors and conduct the training of models for twenty times. Observing that the classification results are robust to the randomness regarding instances, Fig. 6 reports the average $F1$ scores for both hacked and unhacked events, which measure the effectiveness of detection regarding the number of sensors.

Fig. 6 shows that the $F1$ scores resulting from all models increase with the number of sensors, but the marginal improvement of adding more sensors decreases. On the other hand, the upper-bound of performance improvement is restricted to the functionality

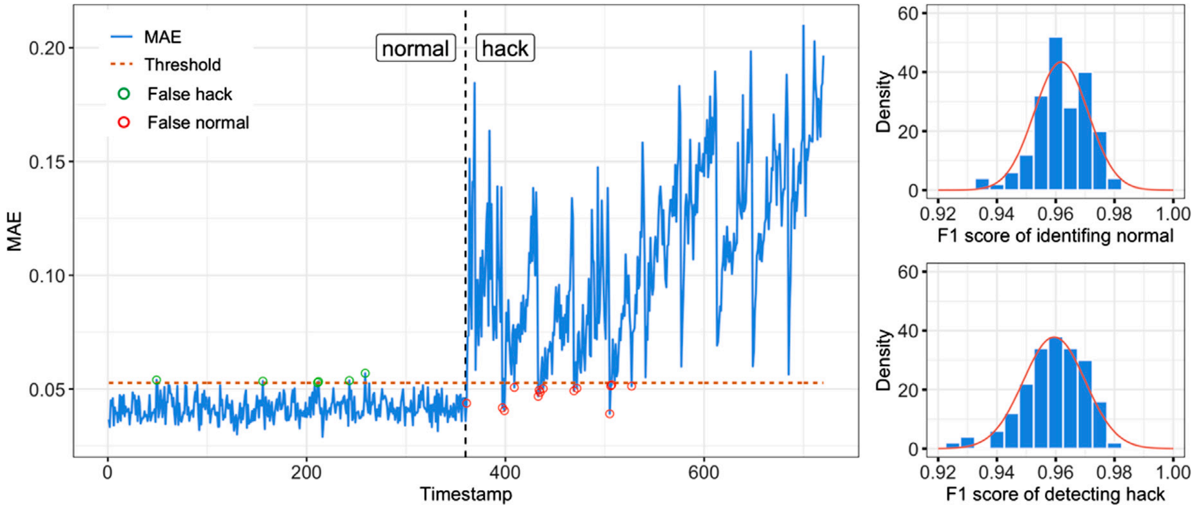


Fig. 7. An example of the detection result and distributions of $F1$ scores for two classes.

of detection models. More specifically, when the number of sensors is 25, the OC-SVM and ND-LOF models achieve the closed best performance. The $F1$ scores of identifying the hacked class for OC-SVM and ND-LOF are 95.1% and 76.7%, respectively; the F1-scores of the unhacked class for the two models are 94.4% and 85.9%, respectively. The performance of the EE model approximates the closed best score when there are 35 sensors, with a score of 60.6% identifying the hacked class and 83.2% identifying the unhacked class successfully. When there are 30 sensors available, the $F1$ scores archive the almost upper limit with 49.6% for identifying the hacked class and 76.8% for identifying the unhacked class. Therefore, To save the marginal budget and energy, a certain number of sensors are sufficiently useful for monitoring hacker intrusion in terms of spoofing the road network.

4.3.2. Evaluation of the LSTM-autoencoder model

The training dataset for the LSTM-autoencoder model consists of 70 days of sensory recording data at a 5-min interval on the pre-selected 50 road links under normal traffic conditions (i.e., without cyber-attack events). Thus, the total timestamps T is 20,040. Given that there are $m = 50$ pre-selected traffic sensors, in each time step, the traffic data is a vector $D_S(t) = (q_1^t, \dots, q_m^t; v_1^t, \dots, v_m^t)$. Labels y_t are provided at each time step and $y_t = 0$ for all t in the training data. The test dataset is traffic data generated with 720 timestamps. The test data contains half normal and half hacked events, and their arrival sequences are randomized. After training the model with the normal traffic data, the LSTM-autoencoder model will continuously classify the traffic status at each timestamp t in the test based on the dynamic threshold. The optimal threshold for the detection was chosen by selecting the threshold that ensured the optimum trade-off between precision and recall, in order to have the best performance in terms of $F1$ score. In this case, the 99% quantiles of MAE for training data from 1 to $t-1$ are set as the threshold ϕ for detection at period t. For the implementation of the LSTM-autoencoder model, we use five hidden layers (including two LSTM encoder layers, two LSTM decoder layers, and one bridge layer between the encoder and decoder modules), a learning rate of 0.0001, the rectified linear activation function. We train the model using the Adam optimizer (Ashraf et al., 2020).

We repeat the experiments 100 times to reduce the variation due to randomness. The detection results are shown in Fig. 7. The left part of Fig. 7 presents an example trajectory in one run. The classification threshold is selected at the 99% percentile of the calculated vector distance of the whole historical training traffic data, which is 0.053. We can observe that there are five normal events out of 360 normal events that are falsely labeled and 12 hacking events out of 360 hacking events that are falsely labeled. The corresponding $F1$ score is calculated as 97.2% and the false positive rate is 1.4%. From the histograms in Fig. 7, the $F1$ scores of identifying normal events of the 100 runs range from 93.5% to 98% with 96.4% on average, and the $F1$ scores of detecting hacking events range from 92% to 98% with 96% on average. The $F1$ scores for both cases perform at a similarly high level, which reveals that the LSTM-autoencoder model can robustly and consistently detect the anomaly for the online sensory data.

We illustrate the performance of the well-trained model in a more realistic setting. The experiment uses the simulated 200 timestamps of new sensory data, which contains normal events at the first 100 timestamps and random hacking and normal events at the second 100 timestamps where the hacking events randomly occur with a probability of 0.1 in each timestamp. We conduct 10 experiments. The average $F1$ score is 97.1% and the average false positive rate is 1.3%. The virtualization of the experimental results till the true hack (alarm) is shown in Fig. 8. It is observed that 9 of the 10 experiments can detect the hacking event the first time it happens, except for the 8th experiment detecting the cyber-attack after missing the first two occurrences. In practice, the cyber-attack detection system will conduct a further investigation at the false alarm but not the false normal. The detector can generally raise a true alarm once the hacking event happens and stops at the true alarm after no more than twice false alarms within 200 timestamps. In conclusion, the well-trained LSTM-based model is applicable to the real-world traffic network with one-class data.

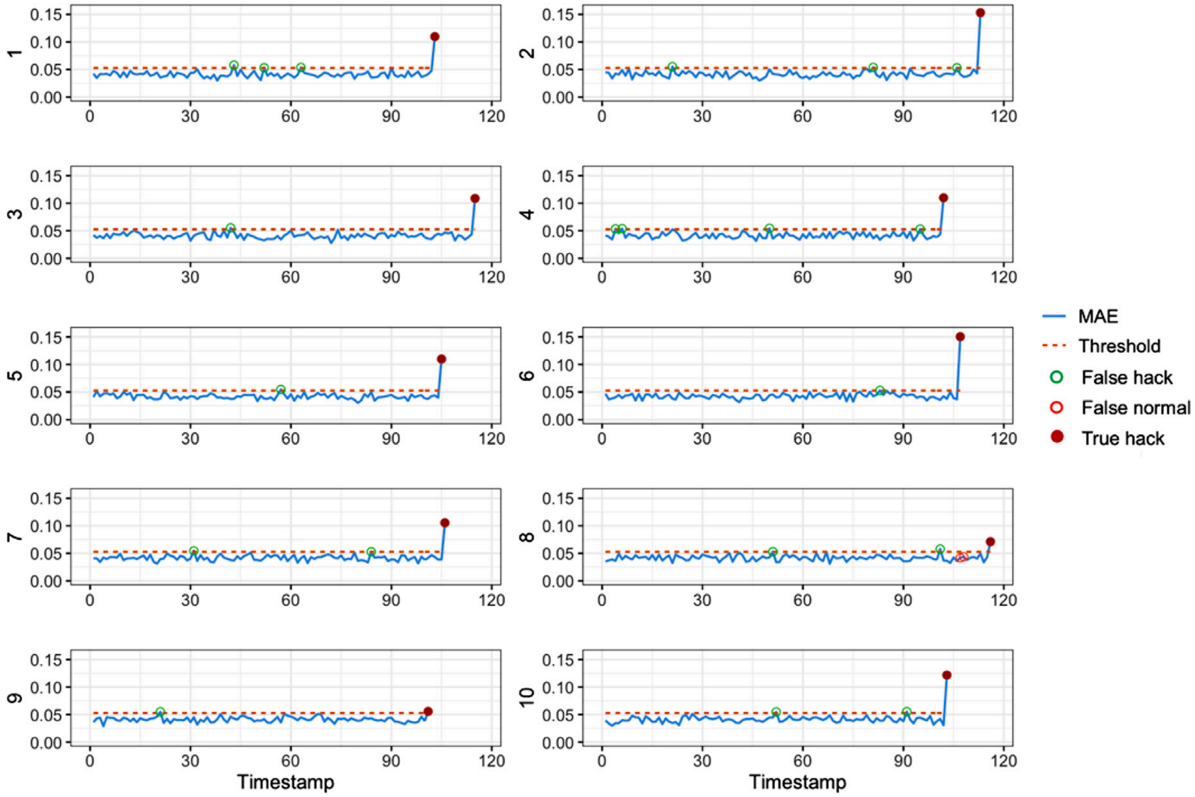


Fig. 8. Experiments of using the proposed LSTM-autoencoder detection model with sampled sequence of sensory data.

4.3.3. Evaluation of the Gaussian process model

To test the proposed Gaussian process model, we need to prepare a smaller training dataset with both hacking and normal events. The initial traffic simulation data contains 200 timestamps, and in each time step, the link-level flow and average speed data $D_S(t) = (q_1^t, \ldots, q_m^t; v_1^t, \ldots, v_m^t)$ are provided. We do not use a large training dataset due to the computational burden of Gaussian processes, particularly with high dimensional data as in this study. The actual status Y_t for time t is also obtained as input to the Gaussian process model. Since the anomalies are rare in practice, there are 20 cyber-attack labels and 180 normal traffic labels in the training data. We continue the experiment with 100 samples as the testing data. The model will continuously check the current status as new data arrives and terminate at time t if it is classified as a cyber-attack event. Otherwise, it will include testing data at time t to the historical data pool to prepare Gaussian process parameters for $t+1$. This means that mislabeled data are likely in the newly added training data. The initial threshold η_0 for this case study is tuned as −50. Fig. 9 presents changes of likelihood based on observed (simulated) traffic data and the time-varying threshold with different initial thresholds −60, −50, and −40 over the horizon. We can see that the detection model can detect the hacking events the second time it appears when the initial threshold is −50. However, the detector will always raise a false alarm if the initial threshold is −60, since there are no hacking events happening before the likelihood exceed the threshold. When the initial threshold is −40, the detection likelihood will never exceed the threshold so that invalid for anomaly detection. This is common for any parametric model as the performance is sensitive to its prior information (e.g., the initial threshold in this case), and the determined initial threshold −50 is appropriate in this network.

To reduce the effect of randomness (i.e., the sequence of events) on the performance evaluation, we repeat the online detection procedure 20 times with the fixed length of the new sequence data (100 timestamps in total). The probability of hacking occurring is 0.1 at each upcoming timestamp. The $F1$ score and false positive rate are recorded for each run, and the average $F1$ score and average false positive rate of the 20 runs is used as the evaluation metric for comparison. The result shows that the average $F1$ score is 89.2% and the average false positive rate is 3.8%. We also find that the detector can raise the alarm at the second occurrence of a cyber-attack on average. We observe that the computational time for training the Gaussian process becomes unmanageable for larger timestamps so a moving-window approach is more pragmatic in a real-world deployment.

4.3.4. Comparative analysis

This subsection compares the performance of the proposed two online transportation network cyber-attack-related anomaly detection with different scales of training data, i.e., the nonparametric LSTM-autoencoder model and the parametric Gaussian processes model. Additionally, four classical offline classification models are employed to serve as comparison benchmarks.

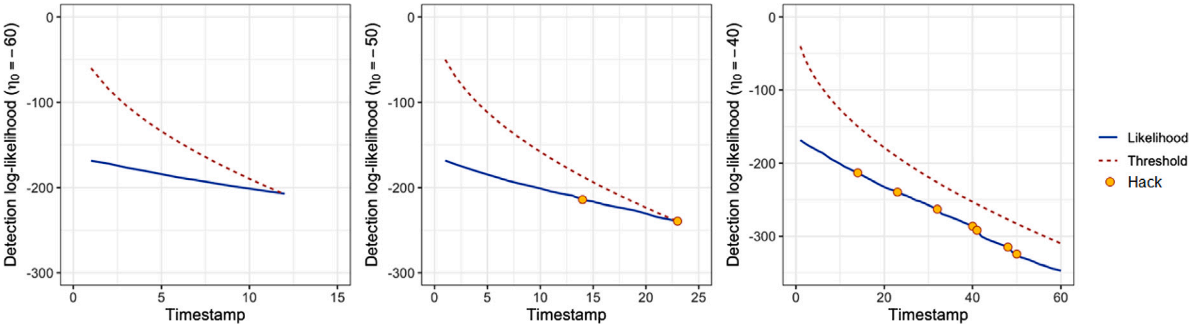


Fig. 9. An example of the Gaussian processes based anomaly detection with different initial thresholds.

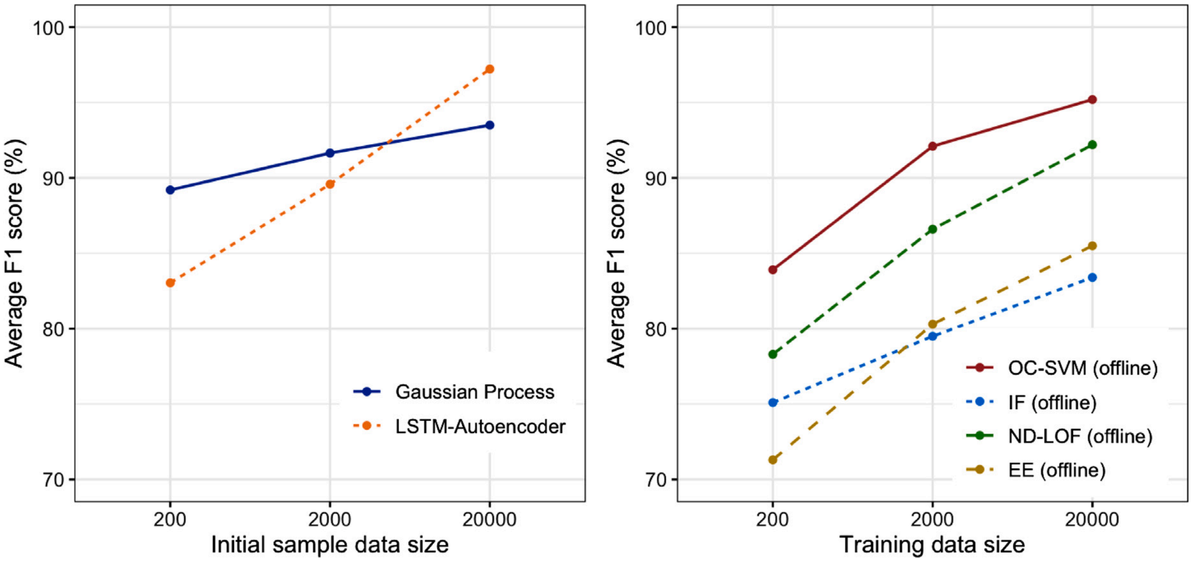


Fig. 10. Comparison of provided models with different data sizes.

Theoretically, the offline model utilizes all traffic data in prediction and should generate higher anomaly classification accuracy. The $F1$ score is used as the evaluation of the accuracy in this comparison.

The right of Fig. 10 presents the $F1$ score as a function of training data size for the four offline classification models. It is expected that a larger training data size leads to higher classification accuracy. Clearly, OC-SVM outperforms other models and has an accuracy rate that is improved from 87.9% to 96.2% when the training data size increases from 200 to 20,000. This is expected because OC-SVM has the advantage of training the classifier and its coefficients using only patterns belonging to the target class distribution, which can improve classification accuracy. IF and EE have comparable accuracy performance and can only achieve as high as 85% when the training size is 20,000.

The left Fig. 10 presents the accuracy index, $F1$ score, for online Gaussian process and LSTM models with an initial sample size ranging from 200 to 20,000. The online models will start with the initial sample size and constantly update training data as new data come in. For both models, a higher sample size results in higher accuracy rates. However, the increase rate of the LSTM model is significantly steeper than that of the Gaussian process model. One possible explanation is that the LSTM model is a type of deep learning model with particular strength in recognizing time-series patterns. Given the time step interval is 5 min in this study, the daily, weekly, and even monthly traffic patterns might not be recognizable in a small initial sample size. But in a larger initial sample size, such as 20,000 or 70 days of data, the LSTM model can capture time-dependent variations and use that to detect the cyber-attack-related anomaly. When the training size is above 2000, both parametric and non-parametric online models can achieve an accuracy rate of above 90%, which shows the robustness of the online cyber-attack event classification models. Our results also show that the classification accuracy rates of the two online models are generally higher than those of offline models. This is expected because online models have the advantage of constantly updating model coefficients as new data arrive, which can help in improving classification accuracy.

4.3.5. Detecting cyber-attack-related traffic anomaly under special events

Many unexpected events that could disrupt normal traffic patterns, such as traffic accidents and extreme weather changes, are not cyber-attacks. The effectiveness of the proposed online anomaly detection model should be validated with the random occurrence of special events. This section investigates the performance of our online anomaly detection model performs in the face of global natural anomalies caused by severe weather and local congestion caused by traffic accidents.

We simulate these two types of special events as follows. To simulate the impact of global natural anomalies due to extreme weather conditions, such as rain and snow, we modify the braking behavior and enlarge the reaction time of drivers in the car-following model during simulations. A longer reaction time means the worse weather conditions. To simulate the traffic congestion caused by special events, such as accidents and crashes, we randomly select road links and generate crashes. Note that the propagation of traffic congestion will use the same traffic flow model, which will differ from the regular traffic patterns.

The training dataset is the sequential attack-free traffic data with changes in weather conditions from normal to extreme conditions and some traffic crashes over 20,040 timestamps. The testing dataset is the sequential unknown traffic data along with changing weather conditions and traffic crashes. It contains 720 timestamps, where half data is attack-free and half contains attacks. To evaluate the impact of special events on the performance of our anomaly detection model, we use the same structure of LSTM-autoencoder as the above model. We conducted 20 runs of the experiments to minimize the randomness during the online detection procedure. The average false positive rate is 1.5% and the average $F1$ score is 94.3%. The false positive rate of detecting attacking anomalies under special events is slightly higher than that of detecting under regular traffic conditions. The $F1$ score for detecting attacking anomalies under special events is smaller than that for detecting anomalies in regular traffic conditions but at a high level. In conclusion, special events do not significantly impact the effectiveness of the robust LSTM-autoencoder model.

5. Sensor location selection optimization

Observing that sensors' locations in the network significantly impact the performance of the developed anomaly detection model, we consider a relevant planning problem in which a transportation agency will expand the sensor coverage to improve the efficiency in detecting cyber-attacks. In addition, transportation agencies are more concerned about the maximization of traffic information gain. To achieve this goal, a multi-objective optimization model is proposed with two objectives: (1) improving the efficiency of cyber-attack detection, and (2) maximizing traffic information gain while considering the budget constraint of transportation agencies.

To be consistent with the notations defined earlier, we define the sensory traffic data at time t as $D_S(t)$, $D_S(t)=(q_1^t,\dots,q_m^t;v_1^t,\dots,v_m^t)$. E_S denotes the set of road links with deployed sensors, and its collected traffic data is defined as $D_{E_S}(t)$. The remainder of links that are not equipped with sensors and are potential candidates for the next phase of sensor network expansion is denoted by $E_{\bar{S}}$ and the traffic data collected from those sensors are defined as $D_{E_{\bar{S}}}(t)$. Assuming the costs of installing sensors on road links are different, and the cost vector is defined as $\boldsymbol{\beta}$. The budget of transportation agencies for deploying new sensors is B. Let $x_e = 1$ denote that the planner decides to install a sensor on each $e \in E_{\bar{S}}$ and $x_e = 0$ denote not installing. The above problem is formulated as

$$\max_{\boldsymbol{x}} Z = (f_1, f_2) \tag{6}$$

$$f_1 = \ell(y_t|\hat{D}^{\boldsymbol{x}}_{E_{\bar{S}}}(t) \cup D_{E_S}(t)) - \ell(y_t|D_{E_S}(t)) \tag{7}$$

$$f_2 = H(\hat{D}^{\boldsymbol{x}}_{E_{\bar{S}}}(t) \cup D_{E_S}(t)) \tag{8}$$

$$s.t.\ \boldsymbol{\beta}^{\intercal}\boldsymbol{x} \le B,$$

$$x_e \in \{0,1\}, \qquad \forall e \in E_{\bar{S}},$$

where $\hat{D}^{\boldsymbol{x}}_{E_{\bar{S}}}(t)$ denotes that the traffic data is collected from links with $x_e = 1$.

The first objective of the sensor location selection problem in Eq. (7) is to maximize the incremental likelihood of detecting anomalies. We can use any online or offline ML-based detection algorithms developed in this work and in Appendix B to estimate this likelihood to measure the efficiency of cyber-attack. In the remainder of this section, we employ the ND-LOF model as it outperforms other algorithms and is sensitive to the sensor number in Fig. 6.

The second objective in Eq. (8) is to maximize the information entropy of traffic flows collected from the selected sensors. Whereas the information theory that quantitatively describes the information distribution is a long-standing field (Shannon, 2001), its definition in the context of traffic flow has only been recently developed in Liu et al. (2022). It proposed a flow-weighted speed-based entropy function that measures the observability of traffic networks. The average speed measured in sensor i is denoted by v_i^t, which is treated as a random variable. The entropy function is given by

$$H(D_S(t)) = -\frac{\sum_{i=1}^{N_s}(v_i^t \cdot q_i^t)\log(v_i^t \cdot q_i^t)}{\sum_{i=1}^{N_s} q_i^t}, \tag{9}$$

where q_i^t is traffic flow reported by traffic sensor i at time t, and N_s is the number of sensors. The flow-weighted speed entropy objective function measures the capability of allocated traffic sensors for capturing accurate information from the traffic system. Note that the speed entropy is further normalized in this study only for the comparison with other objectives, 1 means the highest uncertainty of the traffic information and 0 means no uncertainty.

As mentioned before, solving the problem faces a combinatorial challenge due to: (a) the solution that can simultaneously optimize all the objectives is difficult to be determined, (b) the number of combinations in $E_{\bar{S}}$ satisfying the budget constraint is vast (in the order of the cardinality of the power set $2^{|E_{\bar{S}}|}$), and (c) the ML-based estimation of the likelihood for each combination is time-consuming. Since the traffic information collected for anomaly detection under the potential cyber-attack condition and for monitoring normal traffic are not consistent, previous studies have found that the optimal network locations for two functions differ (Ma and Qian, 2018; Salari et al., 2019). Since the sensor location problem does not have a single solution that optimally meets all objectives simultaneously, the goal is to compute the Pareto front, i.e. a representative set of Pareto optimal solutions. On this Pareto front, none of the objective functions can be improved without degrading the other objective value. Once the Pareto front is found, we can select one solution from it as the best compromise solution for the problem. To find the Pareto front of the above sensor location problem with low computational effort, we propose a meta-heuristic based on the elitist non-dominated sorting genetic algorithm (NSGA-II).

The NSGA-II proposed by Deb et al. (2002) is a popular algorithm to solve multi-objective optimization. The main procedure of NSGA-II is shown in Fig. 11. It uses the fast non-dominated sorting technique and a crowding distance to rank and select non-dominated solutions. First, it randomly generates an initial set of parent solutions P_t of size N, then, an offspring solutions set Q_t is generated after conducting tournament selection, crossover, and mutation on solutions set P_t. Next, a new solutions set R_t of size $2N$ is generated by combining P_t and Q_t. The combined solutions set R_t is classified into different non-dominated front levels L_i with the non-domination sorting technique. Finally, the new solutions set P_{t+1} of size N is found in the order of different front levels. If a front is divided partially like L_3, the algorithm will sort the solutions based on the crowding distance.

The NSGA-II-based multi-objective meta-heuristic algorithm for solving the above sensor location problem is described below:

Algorithm 1: NSGA-II based meta-heuristic algorithm

```
Input: Existing sensors deployed on links $E_S$
Candidate links $E_{\bar{S}}$ to install sensors, $E_S \cup E_{\bar{S}} = E$
Sensor installation budget $B$
Output: Selected links $E_{\hat{S}}$ to install sensors;
Initialization: Parent solutions set $P_t$ of size $N$ that satisfy $\boldsymbol{\beta}^\intercal \boldsymbol{x} \leq B$
$t = 0$, $t_{max}$ = number of iterations
while $t \leq t_{max}$ do
    Binary tournament selection
    Two-point crossover
    Bit flip mutation
    Create offspring solutions set $Q_t$ of size $N$
    Create combined solutions set $R_t = P_t \cup Q_t$
    Use fast non-dominated sorting to create non-dominated fronts $(L)$ of $R_t$, $L = \{L1, L2, \ldots\}$
    $P_{t+1} \leftarrow \infty$ and $i = 0$
    while $|P_{t+1}| + |L_i| \leq N$ do
        $P_{t+1} \leftarrow P_{t+1} \cup L_i$, include $i$th front
        $i = i + 1$
    end
    Calculate crowding distance in $_i$
    Sort $L_i$ in descending order of $\prec_n$
    $P_{t+1} \leftarrow P_{t+1} \cup_i^L [0 : (N - |P_{t+1}|]$
    $t = t + 1$
end
Get the Pareto front $P_{t_{max}}$
Determine the best compromise solution $P^*_{t_{max}}$ from $P_{t_{max}}$
$E_{\hat{S}} \leftarrow$ links $e$ in $P^*_{t_{max}}$ with $x_e = 1$
```

Upon having a Pareto front from the meta-heuristic, we will select one of them as the best compromise solution for sensor allocation using the posterior evaluation of the non-dominated solutions based on a membership function (Zhang et al., 2017). The membership function for each candidate solution i in terms of each objective is represented as:

$$\mu_i = \begin{cases} 1 & \text{if } G_i \leq G_i^{min} \\ \frac{G_i^{max} - G_i}{G_i^{max} - G_i^{min}} & \text{if } G_i^{min} < G_i < G_i^{max} \\ 0 & \text{if } G_i \geq G_i^{max} \end{cases}, \tag{10}$$

where G_i^{min} and G_i^{max} denotes the minimum and maximum values of ith objective function in all non-dominated solution sets. The score for each non-dominated solution k in the multi-objective space is defined as:

$$\mu^k = \frac{\sum_{i=1}^{N_{obj}} u_i^k}{\sum_{k=1}^{M} \sum_{i=1}^{N_{obj}} u_i^k}, \tag{11}$$

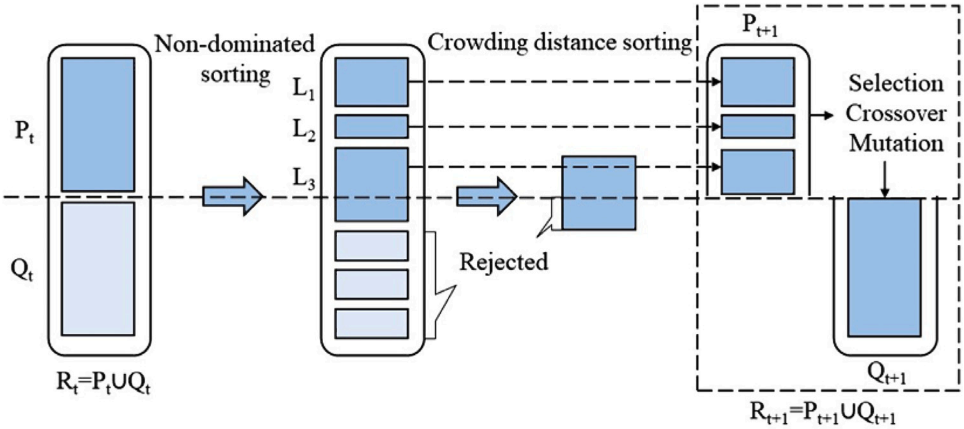


Fig. 11. Procedure of NSGA-II for selecting sensor locations.

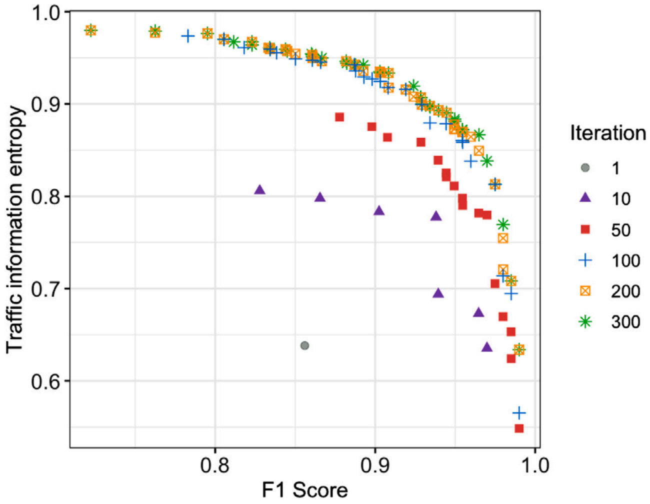


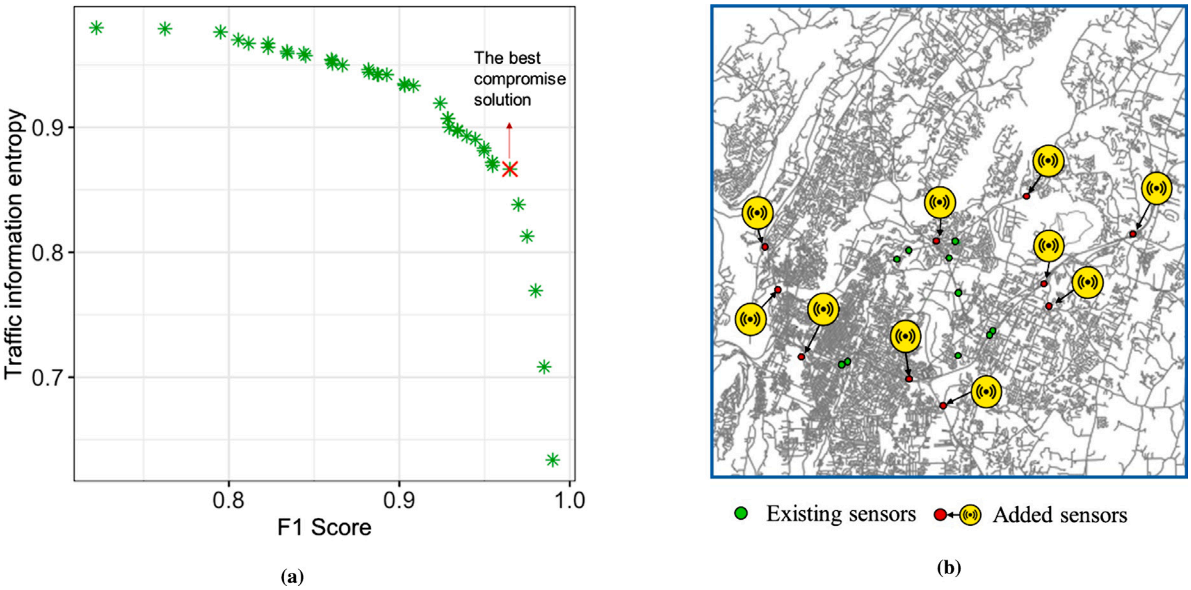
Fig. 12. Pareto fronts evolve along with the generation.

where N_{obj} is the number of objectives and M is the number of non-dominated solutions. The best compromise solution is the one with the maximum value of μ^k.

We implement the above method using traffic simulation data to determine optimal sensor expansion locations. Specifically, we use the simulated three hours morning peak (6–9 am) traffic data and have 10 sensors already existing in the traffic network. Assuming that the cost for installing each sensor is constant, and the budget allows for to installation of at most 10 more sensors into the network to improve cyber-attack event anomaly detection. We implement the multi-objective sensor location optimization based on a set of historical data to do offline anomaly detection. This means we will optimally select 10 additional sensors to improve retrospective detection of cyber-attack events based on the entire historical traffic data. We use four approaches to detect cyber-attack-related anomalies with the data from the initial 10 sensors. These approaches include OC-SVM, ND-LOF, IF, and EE. The detection accuracy rates for the initial 10 sensor cases are 78.1%, 82.9%, 54.8%, and 71.3% for ND-LOF, OC-SVM, IF, and EE models, respectively. By adding more sensors to the traffic network, we expect to see higher accuracy rates.

We apply our NSGA-II-based meta-heuristic algorithm to solve the sensor allocation problem in determining locations to deploy additional sensors on the transportation network. The parent solution size is set as 100 and the maximum number of iterations is 300. The probability of mutation is set as 0.05, and the probability of crossover is set as 0.8. Fig. 12 presents the Pareto fronts that evolve along with the evolution process, different colors represent different iterations from 1 to 300. It is observed that the non-dominated solutions of the 200th iteration are almost overlapping with that of the 300th iteration, indicating that the algorithm is converged before 300 iterations. Thus, the optimal Pareto front is the set of non-dominated solutions from the 300th iteration. Fig. 13(a) shows the optimal Pareto front and the selected best compromise solution by using Eqs. (10)–(11). We can see that the best compromise solution has a $F1$ score of 0.96 and traffic information entropy of 0.87.

Based on the best compromise solution of sensor locations, We find that the detection performances using 20 sensor locations are higher than those based on the data from 10 sensors across all detection algorithms. The traffic information entropy of 0.87 means



(a)

(b)

Fig. 13. The best compromise solution selected from the optimal Pareto front (a), and sensor locations of existing sensors and added sensors of the best solution (b).

a high uncertainty and diversity of the traffic information recorded by these expended sensors. Therefore, our approach consistently improves the cyber-attack detection performance and maximizes the traffic information gain, simultaneously. The detection $F1$ scores for the expended 20 sensors are 96.4%, 96.1%, 73.6%, and 90.1% for ND-LOF, OC-SVM,IF, and EE models, respectively. Fig. 13(b) presents the locations of existing sensors and expanded sensors that are selected by meta-heuristic. The results show the balance between spatial distribution and important traffic features in location selection for new sensors. Specifically, it combines the spatial sparsity of the sensor location to explore more undetected regions as well as the utilization of traffic information collected from the aggregated and probably perturbed traffic flows.

6. Conclusion

Transportation agencies and the industry have rising concerns about security threats, especially cyber-attacks at the transportation network level. This study proposes a data-driven analytical framework leveraging widely accessible stationary sensory data to identify malicious modification or spoofing of roadway traffic speed displayed on the crowd-sourcing map software. We employ a city-scale microscopic traffic simulation to validate our framework based on real-world travel data. The results can help transportation agencies and security authorities identify potential security threats and protect transportation infrastructure from malicious cyber-attacks.

We propose two online cyber-attack-related traffic anomaly detection algorithms for transportation networks, i.e., non-parametric LSTM-autoencoder and parametric-based Gaussian process models. Both online traffic classification models can dynamically update training data as real-time traffic data arrive. We customize the two models to make them suitable to be applied in detecting cyber-attack-related traffic analysis. Results show that both models can achieve an $F1$ score of above 90%, and the performance increases as the training data size increases. Particularly, the LSTM-autoencoder model performs better with a larger training size compared with the Gaussian process model. The results of online models are comparable and even better than major classical offline classification models.

Additionally, we investigate traffic monitoring sensor location selection and expansion optimization problems. Specifically, we develop a multi-objective optimization model and an NSGA-II-based meta-heuristic method to optimally select and expand sensor networks to improve anomaly detection efficiency for potential cyber-attacks and maximize traffic information gain for normal traffic management. We test the performance of the meta-heuristic methods in numerical experiments and find the optimal Pareto front for determining optimal locations to deploy additional sensors in traffic networks. The expanded sensors result in significant improvement in detection accuracy and traffic information gain in all cases.

We acknowledge limitations in the current research, which can motivate future research directions. First, our detection mechanism is effective when hackers' spoofing strategies are to insert virtual vehicles and cause a certain level of speed reduction. When hackers are more strategic in adaptively changing their spoofing targets with traffic conditions, we shall model more complex cyber-attack patterns and test the online detection algorithms in future work. Second, this research focuses on the detection and monitoring of cyber-attacks on crowed sourcing mapping using stationary sensor data. It would be interesting to investigate defense strategies to mitigate such type of cyber-attacks in future studies. Third, this study assumes there are no anomalies caused by sensor malfunction or communication failure. How to handle such communication failure would be another future direction in cyber-attack detection. Fourth, the scenario constructed in our case study is for one mid-size city, Chattanooga. If more data are available, simulations and evaluations in other cities with different regional scales should be implemented to provide more evidence supporting the generalization of this study.

CRediT authorship contribution statement

Ruixiao Sun: Data curation, Formal analysis, Methodology, Validation, Visualization, Writing – original draft. **Qi Luo:** Conceptualization, Formal analysis, Methodology, Supervision, Writing – review & editing. **Yuche Chen:** Conceptualization, Formal analysis, Methodology, Supervision, Writing – review & editing.

Acknowledgments

This material is partially supported by the National Science Foundation Award Number 2213731. The views and opinions of the authors expressed herein do not necessarily reflect those of the U.S. government or any agency thereof. Specifically, neither the U.S. government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. government or any agency thereof.

Appendix A. Supplementary material for Section 3

This section describes one possible scenario for the cyber-attack intrusion of the transportation network, which is then used to generate the input data in our numerical experiments. However, the anomaly detection algorithms developed in this research can be applied to more general settings.

Hackers and normal vehicles make their route decisions under the following information-sharing framework:

1. Contaminated public traffic data $D_N(t)$: At time t, the traveling cost on each edge perceived by normal vehicles is obtained from a centralized mapping source. As the real-time traffic condition is estimated with the centralized planner, this source is at risk of being intruded.
2. Hackers' observation $D_H(t)$: At time t, hackers have access to the actual traffic data in the network.
3. Sensory data $D_S(t)$: At time t, the anomaly detection system only has access to the uncontaminated sensory data at fixed locations.

We illustrate the effect of spoofing trajectory through the standard dynamic user-optimal (DUO) traffic assignment problem (Ran et al., 1993). Let $x^{e,H}(t), x^{e,G}(t), x^{e,N}(t)$ denote the number of hackers, ghost vehicles, and normal vehicles traveling on edge e at time t; $x^{e,H}_{(o,d)}(t), x^{e,N}_{(o,d)}(t)$ denote the number of hackers/normal vehicles originating from o to d through edge e at time t. $u^e(t)$ denotes the inflow rate on edge e at time t and $v^e(t)$ denote the outflow rate on edge e at time t. Both are further split into hackers and normal vehicle flows. The time-dependent edge travel time function is $C_e(x^e(t), u^e(t), v^e(t))$, where $x^e(t)$ is the perceived vehicles on edge e at time t in the planner. We omit the arguments in $C_e(\cdot)$ wherever there is no confusion. For a given path from origin to destination p_{od}, the instantaneous travel time is $\sum_{e \in p_{od}} C_e$. Hackers may distort the perceived travel cost $\hat{C}_e(\cdot)$ by generating ghost vehicles $x^{e,G}(t)$ with slow speed on the edge, so as to reflect on $D_N(t)$. From the normal vehicle's perspective, the number of hackers on each $e \in p_{od}$ is $\hat{x}^e(t) > x^e(t)$. Since $\partial C_e / \partial x^e \geq 0$, the perceived travel time for normal vehicles increases.

We can now formulate the DUO problem with perceived travel time, noting that only normal vehicles are deceived, as follows:

$$\min_{x^H, D_N} \mathbb{E}\left[\sum_{t=0}^{T}\left(z(x^H, D_N) + \sum_{e \in E} \int_0^{u^e(t)} \int_0^{v^e(t)} C_e \, du \cdot dv | D_H(t)\right)\right]$$

$$\begin{aligned}
\text{Subject to: } & \sum_{o \in V, d \in V} x^{e,H}_{(o,d)}(t) = x^{e,H}(t), \quad \forall t \in [T] \\
& \frac{dx^e(t)}{dt} = u^e(t) - v^e(t), \quad \forall e \in E, \forall t \in [T] \\
& AX^H = b, \\
& x^H, \hat{u}, \hat{v} \geq 0
\end{aligned}$$

$$\text{(Sub)} \quad z(x^H, x^G, D_N, u, v) = \min_{x^N} \sum_{t'=0}^{t} \sum_{e \in E} \int_0^{u^e(t')} \int_0^{v^e(t')} \hat{C}_e \, du \cdot dv$$

$$\begin{aligned}
\text{Subject to: } & \sum_{o \in V, d \in V} x^{e,N}_{(o,d)}(t) = x^{e,N}(t) + x^{e,G}(t), \quad \forall t \in [T] \\
& \frac{dx^e(t)}{dt} = u^e(t) - v^e(t), \quad \forall e \in E, \forall t \in [T] \\
& AX^N = b, \quad \forall t \in [T] \\
& x^N \geq 0
\end{aligned}$$

where we use $AX = b$ to represent the flow conservation constraints, flow propagation constraints, and boundary conditions (Ran et al., 1993). Note that the hackers are able to intervene in the normal vehicles' routing behavior by changing D_N in the upper-level optimization, whereas the goal is to minimize their own costs, unlike in other studies where the hackers are modeled as the adversaries and solve min–max problems (Davis et al., 2020; Huang et al., 2021).

We can divide the edges into hacked edges E^H and unhacked edges E^N such that $E^H \cup E^N = E$. Hacked edges are edges covered by the spoofing trajectory. Note that normal vehicles optimizing their routes using $\hat{C}_e$ for $e \in E^H$ will deviate from the optimal routes under C_e, hence the traffic on unhacked edges is also affected.

The problem above is the classic DUO problem by replacing $\hat{C}_e$ with C_e for all edges. Let X^*_{hacked} be the optimal solution for solving the above DUO problem under $\hat{C}_e$ and $X^*_{unhacked}$ be the solution using the true trajectory. The analysis of the DUO problem shows that:

$$X^{e,*,N}_{hacked} \leq X^{e,*,N}_{unhacked}, \forall e \in E^H, \qquad X^{e,*,N}_{hacked} \geq X^{e,*,N}_{unhacked}, \forall e \in E^N.$$

This means that the other normal road users will detour from the original paths with fake low-traffic-speed if they use hacked edges. Thus, the hacker will enjoy traveling on routes with significantly reduced flow.

Appendix B. Supplementary material for Section 4

We summarize four offline anomaly detection methods that are commonly used to train a classifier with one-class traffic data (i.e., with normal traffic data only).

- **OC-SVM**: The main idea of OC-SVM is to find an optimal hyperplane in the sensory data configurations (feature space) by constructing a measurement function $f(x) = sign(w \cdot \phi(x)) - \rho$, named signed distance. w is the normal vector of $f(x)$, ρ refers to the offset term, $\phi(x)$ is the mapping of the sample in the high dimensional space. The data points are separated by the signed distance from the origin at the maximum distance. The data on one side of the origin to the hyperplane are classified as the hacked class, and those on the outer side of the hyperplane are classified as the unhacked class. The classifier aims to identify samples as the hacked class (labeled −1) and unhacked class (labeled 1) with high accuracy. In order to separate the sample points from the origin as much as possible, the optimization of one-class SVM is considered as a quadratic programming problem as shown below:

$$min_{w,\xi,\rho} \frac{\|w\|^2}{2} - \rho + \frac{1}{vn} \sum_{i=1}^{n} \xi_i \tag{B.1}$$

$$s.t.\ w^T \phi(x_i) \geqq \rho - \xi_i, \xi_i \geqq 0, \tag{B.2}$$

  where ξ_i refers to the penalty term added to the objective function to avoid over-fitting, n is the size of the normal training data set, v is referring to the regularization parameter.
- **IF**: The basic model of an isolation forest is the isolation tree. The isolation tree partitions the traffic data samples by randomly selecting data attributes (i.e., the flow and speed of vehicles driving through the roadway sensors) and divided sample data. The cyber-attack detection task herein can be solved by ranking the abnormality of the hacking sensor sample points. The isolation tree can compute the abnormal score of the sample through:

$$S(x,n) = 2^{-\frac{E(h(x))}{c(n)}}, \tag{B.3}$$

  where $h(x)$ refers to the path length of novel point x in the novel sample set (unknown abnormal and normal samples). $E(h(x))$ refers to the average of $h(x)$ from all the isolation trees. $c(n)$ refers to the path length of unsuccessful searches isolation tree. The results of cyber-attack detection are acquired by evaluating the value of $S(x,n)$. When $S(x,n)$ is close to 0, the sample is judged as the unhacked class. When $S(x,n)$ is close to 1, the sample is judged as the hacked class.
- **ND-LOF**: The outlier detection with the local outlier factor is a density-based local outlier detection algorithm. The main idea is that, for a novel traffic sample, it can be considered as a normal traffic condition if all the other samples in its local neighborhood are dense. In contrast, an abnormal sample point that can be identified as the hacked case is far from the nearest neighbor's normal sample points. The LOF algorithm can quantify the prementioned criterion by calculating the local outlier factor:

$$LOF_k(x) = \frac{\sum_{o \in N_k(x)} \frac{Lrd_k(o)}{Lrd_k(x)}}{|N_k(x)|}, \tag{B.4}$$

  where $N_k(x)$ is the k-distance neighborhood and $Lrd_k(x)$ is the local reachability density. If $LOF_k(x)$ is greater than 1, it means that the density of novel sample x is smaller than the density of its surrounding sample points, and sample point x is more likely to be the hacked class. If $LOF_k(x)$ is less than 1, indicating the density of sample x is greater than the density of its surrounding points, and sample x is considered as the unhacked class.

- **EE**: Elliptic envelope, also known as the minimum covariance determinant (MCD), is a robust location and distribution estimation algorithm for detecting cyber-attacks. The primary idea is to utilize a tolerance ellipse to cover the normal sample data. A tolerance ellipse can be obtained by using the Mahalanobis distance:

$$MD(x) = \sqrt{(x - \hat{\mu}_{MCD})^t \sum_{MCD}^{-1} (x - \hat{\mu}_{MCD})}, \tag{B.5}$$

  where x is the sample point, $\hat{\mu}_{MCD}$ is the location estimated by the MCD algorithm, and $\sum_{MCD}^{-1}(\cdot)$ refers to the covariance estimated by the MCD algorithm. The tolerance ellipse prefers to cover the normal sample points rather than the abnormal ones. The data points that are not covered in the tolerance ellipse are identified as the unhacked class.

References

Ai, L., Soltangharaei, V., Bayat, M., Greer, B., Ziehl, P., 2021. Source localization on large-scale canisters for used nuclear fuel storage using optimal number of acoustic emission sensors. Nucl. Eng. Des. 375, 111097.

Al Jallad, K., Aljnidi, M., Desouki, M.S., 2020. Anomaly detection optimization using big data and deep learning to reduce false-positive. J. Big Data 7 (1), 1–12.

Ashraf, J., Bakhshi, A.D., Moustafa, N., Khurshid, H., Javed, A., Beheshti, A., 2020. Novel deep learning-enabled LSTM autoencoder architecture for discovering anomalous events from intelligent transportation systems. IEEE Trans. Intell. Transp. Syst. 22 (7), 4507–4518.

Bawaneh, M., Simon, V., 2019. Anomaly detection in smart city traffic based on time series analysis. In: 2019 International Conference on Software, Telecommunications and Computer Networks. SoftCOM, IEEE, pp. 1–6.

Berger, I., Rieke, R., Kolomeets, M., Chechulin, A., Kotenko, I., 2018. Comparative study of machine learning methods for in-vehicle intrusion detection. In: Computer Security. Springer, pp. 85–101.

Bhuyan, M.H., Bhattacharyya, D.K., Kalita, J.K., 2013. Network anomaly detection: Methods, systems and tools. IEEE Commun. Surv. Tutor. 16 (1), 303–336.

Boquet, G., Morell, A., Serrano, J., Vicario, J.L., 2020. A variational autoencoder solution for road traffic forecasting systems: Missing data imputation, dimension reduction, model selection and anomaly detection. Transp. Res. C 115, 102622.

Bouzeraib, W., Ghenai, A., Zeghib, N., 2020. A multi-objective genetic GAN oversampling: Application to intelligent transport anomaly detection\. In: 2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems. HPCC/SmartCity/DSS, IEEE, pp. 1142–1149.

Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J., 2000. LOF: Identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. pp. 93–104.

Buckland, M., Gey, F., 1994. The relationship between recall and precision. J. Am. Soc. Inf. Sci. 45 (1), 12–19.

CarPro, 2022. Study reveals where drivers are most reliant on their GPS. URL: https://www.carpro.com/blog/where-drivers-are-most-dependent-on-gps-systems.

Chawla, S., Zheng, Y., Hu, J., 2012. Inferring the root cause in road traffic anomalies. In: 2012 IEEE 12th International Conference on Data Mining. IEEE, pp. 141–150.

CNN, 2020. Artist uses 99 phones to trick Google into traffic jam alert kernel description. URL: https://www.cnn.com/style/article/artist-google-traffic-jam-alert-trick-scli-intl/index.html.

Comert, G., Rahman, M., Islam, M., Chowdhury, M., 2021. Change point models for real-time cyber attack detection in connected vehicle environment. IEEE Trans. Intell. Transp. Syst..

Dang, T.T., Ngan, H.Y., Liu, W., 2015. Distance-based k-nearest neighbors outlier detection method in large-scale traffic data. In: 2015 IEEE International Conference on Digital Signal Processing. DSP, IEEE, pp. 507–510.

Davis, N., Raina, G., Jagannathan, K., 2020. A framework for end-to-end deep learning-based anomaly detection in transportation networks. Transp. Res. Interdiscip. Perspect. 5, 100112.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. 6 (2), 182–197.

Dias, M.L., Mattos, C.L.C., da Silva, T.L., de Macêdo, J.A.F., Silva, W.C., 2020. Anomaly detection in trajectory data with normalizing flows. In: 2020 International Joint Conference on Neural Networks. IJCNN, IEEE, pp. 1–8.

Fei, X., Mahmassani, H.S., 2011. Structural analysis of near-optimal sensor locations for a stochastic large-scale network. Transp. Res. C 19 (3), 440–453.

Fei, X., Mahmassani, H.S., Murray-Tuite, P., 2013. Vehicular network sensor placement optimization under uncertainty. Transp. Res. C 29, 14–31.

Feng, Y., Huang, S.E., Wong, W., Chen, Q.A., Mao, Z.M., Liu, H.X., 2022. On the cybersecurity of traffic signal control system with connected vehicles. IEEE Trans. Intell. Transp. Syst..

Figueiras, P., Gonçalves, D., Costa, R., Guerreiro, G., Georgakis, P., Jardim-Gonçalves, R., 2019. Novel big data-supported dynamic toll charging system: Impact assessment on Portugal's shadow-toll highways. Comput. Ind. Eng. 135, 476–491.

Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., Vázquez, E., 2009. Anomaly-based network intrusion detection: Techniques, systems and challenges. Comput. Secur. 28 (1–2), 18–28.

Guerrero-Ibáñez, J., Zeadally, S., Contreras-Castillo, J., 2018. Sensor technologies for intelligent transportation systems. Sensors 18 (4), 1212.

Haklay, M., 2010. How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance survey datasets. Environ. Plan. B: Plann. Des. 37 (4), 682–703.

Huang, S.E., Feng, Y., Liu, H.X., 2021. A data-driven method for falsified vehicle trajectory identification by anomaly detection. Transp. Res. C 128, 103196.

Ji, Y., Wang, L., Wu, W., Shao, H., Feng, Y., 2020. A method for LSTM-based trajectory modeling and abnormal trajectory detection. IEEE Access 8, 104063–104073.

Kerns, A.J., Shepard, D.P., Bhatti, J.A., Humphreys, T.E., 2014. Unmanned aircraft capture and control via GPS spoofing. J. Field Robotics 31 (4), 617–636.

Khan, Z., Chowdhury, M., Islam, M., Huang, C.-Y., Rahman, M., 2020. Long short-term memory neural network-based attack detection model for in-vehicle network security. IEEE Sensors Lett. 4 (6), 1–4.

Laszka, A., Abbas, W., Vorobeychik, Y., Koutsoukos, X., 2019. Detection and mitigation of attacks on transportation networks as a multi-stage security game. Comput. Secur. 87, 101576.

Laxhammar, R., Falkman, G., 2013. Online learning and sequential anomaly detection in trajectories. IEEE Trans. Pattern Anal. Mach. Intell. 36 (6), 1158–1173.

Lee, C., Iesiev, A., Usher, M., Harz, D., McMillen, D., 2020. IBM X-force threat intelligence index 2020. United States of America.

Li, X., Yu, Y., Sun, G., Chen, K., 2018. Connected vehicles' security from the perspective of the in-vehicle network. IEEE Netw. 32 (3), 58–63.

Li, X., Zhang, H., Miao, Y., Ma, S., Ma, J., Liu, X., Choo, K.-K.R., 2021. Can bus messages abnormal detection using improved SVDD in Internet of Vehicle. IEEE Internet Things J..

Liu, F.T., Ting, K.M., Zhou, Z.-H., 2012. Isolation-based anomaly detection. ACM Trans. Knowl. Discov. Data (TKDD) 6 (1), 1–39.

Liu, Z., Wang, Y., Cheng, Q., Yang, H., 2022. Analysis of the information entropy on traffic flows. IEEE Trans. Intell. Transp. Syst..

Ma, W., Qian, Z.S., 2018. Estimating multi-year 24/7 origin-destination demand using high-granular multi-source traffic data. Transp. Res. C 96, 96–121.

MSN, 2022. Hackers caused a massive traffic jam in Moscow using a ride-hailing app kernel description. URL: https://www.msn.com/en-us/news/technology/hackers-caused-a-massive-traffic-jam-in-moscow-using-a-ride-hailing-app/ar-AA11pGeR.

Mudge, R., Mahmassani, H.S., Haas, R., Talebpour, A., Carroll, L., et al., 2013. Work Zone Performance Measurement Using Probe Data. Technical Report, United States. Federal Highway Administration. Office of Operations.

Münz, G., Li, S., Carle, G., 2007. Traffic anomaly detection using k-means clustering. In: GI/ITG Workshop MMBnet. pp. 13–14.

Ngan, H.Y., Yung, N.H., Yeh, A.G., 2015. Outlier detection in traffic data based on the Dirichlet process mixture model. IET Intell. Transp. Syst. 9 (7), 773–781.

Nguyen, H., Liu, W., Chen, F., 2016. Discovering congestion propagation patterns in spatio-temporal traffic data. IEEE Trans. Big Data 3 (2), 169–180.

Oh, M.-h., Iyengar, G., 2019. Sequential anomaly detection using inverse reinforcement learning. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1480–1490.

Oucheikh, R., Fri, M., Fedouaki, F., Hain, M., 2020. Deep real-time anomaly detection for connected autonomous vehicles. Procedia Comput. Sci. 177, 456–461.

Ran, B., Boyce, D.E., LeBlanc, L.J., 1993. A new class of instantaneous dynamic user-optimal traffic assignment models. Oper. Res. 41 (1), 192–202.

Rousseeuw, P.J., Driessen, K.V., 1999. A fast algorithm for the minimum covariance determinant estimator. Technometrics 41 (3), 212–223.

Salari, M., Kattan, L., Lam, W.H., Lo, H., Esfeh, M.A., 2019. Optimization of traffic sensor location for complete link flow observability in traffic network considering sensor failure. Transp. Res. B 121, 216–251.

Shannon, C.E., 2001. A mathematical theory of communication. ACM SIGMOBILE Mob. Comput. Commun. Rev. 5 (1), 3–55.

Shirazi, S.N., Simpson, S., Gouglidis, A., Mauthe, A., Hutchison, D., 2016. Anomaly detection in the cloud using data density. In: 2016 IEEE 9th International Conference on Cloud Computing. CLOUD, IEEE, pp. 616–623.

Stefanovitch, N., Alshamsi, A., Cebrian, M., Rahwan, I., 2014. Error and attack tolerance of collective problem solving: The darpa shredder challenge. EPJ Data Sci. 3, 1–27.

Sun, R., Gui, R., Neema, H., Chen, Y., Ugirumurera, J., Severino, J., Pugliese, P., Laszka, A., Dubey, A., 2021. TRANSIT-GYM: A simulation and evaluation engine for analysis of bus transit systems. In: 2021 IEEE International Conference on Smart Computing. SMARTCOMP, IEEE, pp. 69–76.

Thajchayapong, S., Barria, J.A., 2010. Anomaly detection using microscopic traffic variables on freeway segments. Transp. Res. Board Natl. Acad. 10–2393.

Wang, Q., Lv, W., Du, B., 2018a. Spatio-temporal anomaly detection in traffic data. In: Proceedings of the 2nd International Symposium on Computer Science and Intelligent Control. pp. 1–5.

Wang, G., Wang, B., Wang, T., Nika, A., Zheng, H., Zhao, B.Y., 2018b. Ghost riders: Sybil attacks on crowdsourced mobile mapping services. IEEE/ACM Trans. Netw. 26 (3), 1123–1136.

Wang, Y., Wong, J., Miner, A., 2004. Anomaly intrusion detection using one class SVM. In: Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop, 2004. IEEE, pp. 358–364.

Wang, Y., Xu, J., Xu, M., Zheng, N., Jiang, J., Kong, K., 2016. A feature-based method for traffic anomaly detection. In: Proceedings of the 2Nd ACM SIGSPATIAL Workshop on Smart Cities and Urban Analytics. pp. 1–8.

Williams, C.K., 2006. Gaussian Processes for Machine Learning, Vol. 2, no. 3.

Xu, X., Lo, H.K., Chen, A., Castillo, E., 2016. Robust network sensor location for complete link flow observability under uncertainty. Transp. Res. B 88, 1–20.

Zhang, S., Guo, H., Zhu, K., Yu, S., Li, J., 2017. Multistage assignment optimization for emergency rescue teams in the disaster chain. Knowl.-Based Syst. 137, 123–137.

Zhang, J., Paschalidis, I.C., 2017. Statistical anomaly detection via composite hypothesis testing for Markov models. IEEE Trans. Signal Process. 66 (3), 589–602.

Zhou, X., List, G.F., 2010. An information-theoretic sensor location model for traffic origin-destination demand estimation applications. Transp. Sci. 44 (2), 254–273.

Zhu, S., Yuchi, H.S., Zhang, M., Xie, Y., 2021. Sequential adversarial anomaly detection with deep Fourier kernel. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 3345–3349.