Consistent Valid Physically-Realizable Adversarial Attack Against Crowd-Flow Prediction Models

Hassan Ali, Muhammad Atif Butt, Fethi Filali, *Senior Member, IEEE*,
Ala Al-Fuqaha, *Senior Member, IEEE*, and Junaid Qadir, *Senior Member, IEEE*

***Abstract*— Recent works have shown that deep learning (DL) models can effectively learn city-wide crowd-flow patterns, which can be used for more effective urban planning and smart city management. However, DL models have been known to perform poorly on inconspicuous adversarial perturbations. Although many works have studied these adversarial perturbations in general, the adversarial vulnerabilities of deep CFP models in particular have remained largely unexplored. In this paper, we perform a rigorous analysis of the adversarial vulnerabilities of DL-based CFP models under multiple threat settings, making three-fold contributions; 1) we propose *CaV-detect* by formally identifying two novel properties—Consistency and Validity—of the CFP inputs that enable the detection of standard adversarial inputs with 0% false acceptance rate (FAR); 2) we leverage universal adversarial perturbations and an adaptive adversarial loss to present adaptive adversarial attacks to evade *CaV-detect* defense; 3) we propose *CVP*, a Consistent, Valid and Physically-realizable adversarial attack, that explicitly inducts the consistency and validity priors in the perturbation generation mechanism. We find out that although the crowd-flow models are vulnerable to adversarial perturbations, it is extremely challenging to simulate these perturbations in physical settings, notably when *CaV-detect* is in place. We also show that *CVP* attack considerably outperforms the adaptively modified standard attacks in FAR and adversarial loss metrics. We conclude with useful insights emerging from our work and highlight promising future research directions.**

***Index Terms*— Deep neural networks, CFP, adversarial ML.**

I. Introduction

THE CFP (CFP) problem aims to predict the CFS (CFS) at some future time, given a set of CFS at previous times. CFP has significance in diverse fields including modeling and understanding human behavior [1], transportation management [2], and smart-city planning [3]. Deep Neural Networks (DNNs) represent a promising technique for solving the cFP problem [2], [3], [4] and related tasks [1], [5], [6]. However, the performance of a DNN highly depends on its training data, which causes it to be vulnerable to adversarial perturbations—undetectable noise, intentionally induced in the input in order to change the DNN output [7], [8], [9], [10], [11]. Although several CFP models based on diverse architectures have been proposed, to the best of our knowledge, the adversarial vulnerabilities of these models remain largely unexplored.

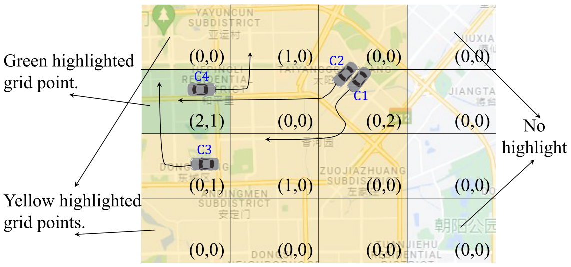


Fig. 1. An illustration of gridding a region and computing the inflow and the outflow matrices from the flow of crowd between adjacent regions (grid points). We typically assume the adjacency within the 2^{nd} neighborhood—the adjacent grid points of green highlighted area are highlighted yellow.

In this paper, we bridge this gap by studying the worst-case performance of three popular and diverse CFP models—Multi-Layer Perceptron (MLP) [3], Spatio-Temporal Resnet (STResnet) [2] and Temporal Graph Convolutional Neural Network (TGCN) [12]—under multiple attack settings. For evaluation, we consider the TaxiBJ dataset, which is one of the most commonly used datasets for CFP. TaxiBJ divides a city into 32×32 grid points (regions) and records the region-wise crowd inflow[1] and outflow[2] at half-hourly intervals [2], [3], [13], as illustrated in Fig. 1.

A. Challenges

Firstly, input structure of different CFP models in literature vary significantly. For example, the TGCN [12] takes the CFS history of a pre-defined length at half-hourly intervals as input. In contrast, STResnet [2] takes three sets of inputs representing hourly, daily, and weekly histories of the pre-defined length. For a fair evaluation of different architectures, the models should be evaluated under similar input settings.

Secondly, recent years have witnessed an arms race between adversarial attackers and defenders—most of the attacks and

Manuscript received 5 March 2023; revised 29 June 2023 and 12 October 2023; accepted 27 November 2023. Date of publication 28 December 2023; date of current version 31 May 2024. This work was supported by the Qatar National Research Fund (a member of Qatar Foundation) through National Priorities Research Program (NPRP) under Grant 13S-0206-200273. The Associate Editor for this article was A. Jolfaei. *(Corresponding author: Junaid Qadir.)*

Hassan Ali and Muhammad Atif Butt were with the Department of Electrical Engineering, Information Technology University (ITU), Lahore, Punjab 54600, Pakistan.

Fethi Filali is with the Qatar Mobility Innovations Center (QMIC), Doha, Qatar.

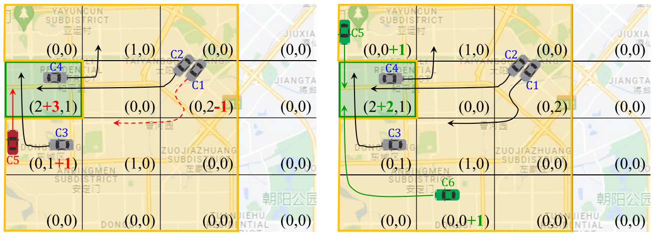
Ala Al-Fuqaha is with the Information and Computing Technology (ICT) Division, College of Science and Engineering (CSE), Hamad Bin Khalifa University (HBKU), Doha, Qatar.

Junaid Qadir is with the Department of Computer Science and Engineering, College of Engineering, Qatar University, Doha, Qatar (e-mail: junaid.qadir@qu.edu.qa).

Digital Object Identifier 10.1109/TITS.2023.3343971

[1] Total devices flowing into a grid point from its adjacent grid points.

[2] Total devices flowing out of a grid point into its adjacent grid points.



(a) Perturbed by PGD attack (b) Perturbed by *CVP* attack

Fig. 2. An illustration of *invalid* and *valid* perturbations generated by the standard PGD (red digits).

defenses were proven ineffective by more adaptive defenses and attacks, respectively, within a few months after they were proposed. Therefore, developing novel adaptive defense and attack strategies to comprehensively analyze adversarial attacks (and their limitations) on DNNs is both extremely important and challenging [14], [15].

B. Findings and Contributions

We first analyze different CFP models against three standard adversarial attacks—FGSM, i-FGSM, and PGD attacks—and show that the CFP models, much like other deep learning models, are vulnerable to these attacks under several design choices. However, we note that these vulnerabilities are mainly limited to the digital attack setting, which assumes a worst-case attacker who has access to the digital input pipeline of the CFP model.

We then identify two properties—*consistency* and *validity*—that natural crowd-flow inputs must satisfy. Although these properties are natural and intuitive, to the best of our knowledge, they have not been emphasized or used in previous works. The property of *consistency* requires that the CFS history at any time must be consistent with the CFS at the previous times. In relation to *validity*, the inflow to and outflow from a particular grid point at any time, by definition, must always be less than the accumulative outflows from and inflows to the adjacent grid points respectively. As illustrated in Fig. 2(a) with example, adversarial perturbations of standard attacks contradict these relationships, and therefore, can be easily invalidated at test time. Noting that the adversarial inputs generated by the standard attacks are *inconsistent* and *invalid*, we show the usefulness of these properties by proposing ***CaV-detect***, a novel consistency and validity check mechanism to detect adversarial inputs at test time by analyzing input consistency and validity. Results show that *CaV-detect* can detect standard adversarial inputs with 0% FAR (FRR $\leq$ 0.5%).

Assuming an expert attacker, we adaptively modify standard adversarial attacks to evade *CaV-detect* by combining universal adversarial perturbations [16] and adaptive adversarial loss. Compared to non-adaptive standard attacks with FAR of 0%, the adaptive attacks typically achieved FAR of $\geq$80% (FRR $\leq$ 0.5%).

We then propose ***CVP* attack, a Consistent, Valid, and Physically-realizable** adversarial attack that explicitly inducts the consistency and validity priors in the adversarial input generation mechanism to find consistent and valid adversarial perturbations (see Fig. 2(b)), and outperforms the standard and the adaptive attacks in both the FAR ($\approx$100%) and the adversarial loss against *CaV-detect*.

We then analyze the physical realizability of adaptive PGD and *CVP* attacks. Our findings highlight that realizing adversarial perturbations under the physical setting requires an impractically large number of adversarially controlled devices, particularly, when *CaV-detect* is in place.

Although the proposed *CVP* attack can be implemented in the physical world, we have chosen in this work to rely on simulation of the attacker in our Python framework. The pragmatic choice of opting for simulation for our analysis instead of relying on a real-world testbed arises from the fact that developing a CFP real-world testbed would be prohibitive from the perspective of cost and time. However, assuming a simulation model that implements the assumed environmental model, we can execute the adversarial attack in the physical world. To foster further research and enable empirical testing of our framework, we have made our code openly available at *https://github.com/hassanalikhatim/CVP-Attack/*.

Finally, our qualitative evaluations show that the CFP models exhibit limited expressiveness—the resulting models, despite showing small test errors, are incapable of producing certain outputs. We attribute this to TaxiBJ data comprising clustered and highly similar CFS [3].

Our main contributions are listed below:

- We are the first to study the adversarial vulnerabilities of the CFP models.
- We formalize two novel properties —*consistency* and *validity*—of CFP inputs and show their usefulness by proposing a novel defense method named *CaV-detect* that achieves 0% FAR with $\leq$0.5% false rejection rate (FRR).
- We combine adaptive loss with universal adversarial perturbation to exhaustively test *CaV-detect*.
- We induct the consistency and validity priors in the adversarial input generation mechanism to propose *CVP* attack that addresses several shortcomings of the standard attacks.

II. Related Work

Owing to the recent developments in intelligent transportation systems (ITS), road traffic congestion forecasting is becoming one of the key steps in curtailing delays and associated costs in traffic management [17]. In the following, we highlight some of the notable and recent works on CFS prediction.

A. CFS Prediction

Depending on the characteristics, structure and quality of the data, various kinds of machine learning (ML) techniques are employed to develop road traffic congestion models. In the literature, these CFP techniques are widely categorized into three main branches—probabilistic and statistical reasoning-based crowd-flow models [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], shallow ML techniques [31], [32], [33], [34], [35], [36], [37], [38], [39] and deep learning (DL) models [17], [24], [40], [41], [42], [43]. Our work focuses on studying the adversarial vulnerabilities of DL-based CFS prediction models. More specifically, we choose three CFP models—MLP model by Jiang et al. [3], STResnet model by

Zhang et al. [2] built over the spatio-temporal residual unit modeling both the spatial dependencies using convolutional layers and the temporal dependencies by concatenating CFS from the recent past into a tuple, and T-GCN model by Zhao et al. [12] using graph convolutional networks (GCN) to model spatial and temporal dependacies. All our CFP models are of notably different architectures for the robustness evaluation. Our choices are based on the recency, diversity, relevance to the problem, and popularity of the architecture. All of these model architectures were trained and evaluated on TaxiBJ dataset in their original papers.

B. Adversarial Attacks on DL Models

Adversarial attacks are small imperceptible changes to the input to fool DNNs. Since they were discovered by Szegedy et al. [9] for image classification, many works have shown that DNNs are generally vulnerable to these attacks in a range of applications including Computer Vision (CV) [7], [44], [45], audio processing [46], networking [47], [48], [49] and Natual Language Processing(NLP) [50], [51], [52]. Adversarial attacks may assume a black-box [7], [53], [54] or a white-box threat model [55], [56], [57]. Several defense techniques have been proposed [55], [58], [59], [60], [61], ranging from model alteration [59], [60]) to input transformation [58], [62], to defend against adversarial attacks. However, most of these defenses were proven ineffective by the later works [14], [63], [64]. More specifically, Athalye et al. [14] note that most of the proposed adversarial defenses can be easily circumvented by the adaptive adversarial attacks customized to render the defense ineffective, leaving adversarial training as the only reliable empirical adversarial defense that survives the test of customized adaptive attacks, in addition to the certified defenses.

In this work, we assume a white-box threat model assuming an attacker knowledgeable of the model architecture and weights. Let an input $x \in \mathcal{X}$, where $\mathcal{X}$ denotes the valid input feature space, produce a true output $\mathcal{F}^*_\theta(x)$, where θ^* denotes the optimized parameters of $\mathcal{F}$. The goal of an attack is to compute an adversarial perturbation δ^*, in order to get the desired output, y_{target} from the model when the perturbation is added to the input.

$$\delta^* = \underset{\delta \in \mathcal{B}(\epsilon)}{\operatorname{argmin}} \left(\mathcal{F}_{\theta^*}(x+\delta) - y_{target}\right)^2 \tag{1}$$

where $\mathcal{B}(\epsilon)$ denotes a pre-defined bounded set of allowed perturbations. One of the most common choices for $\mathcal{B}(\epsilon)$ is an l_∞ ball, defined as, $\boldsymbol{\delta} \in \mathcal{B}_\infty(\epsilon) := -\epsilon \leq \boldsymbol{\delta} \leq \epsilon$. Eq-(1) is iteratively optimized depending on the attack algorithm [9]. In this paper we use three standard adversarial attacks—Fast Gradient Sign Method (FGSM) [55], iterative-FGSM (i-FGSM) and Projected Gradient Descent (PGD) [56]—for evaluation.

III. Methodology

We first formulate the CFP problem in the context of the TaxiBJ dataset and formally define the consistency and validity properties of CFS inputs. Based on these properties, we propose *CaV-detect*, to detect adversarially perturbed inputs by analyzing their consistency and validity. Finally, we present our novel algorithm of *CVP* attack for generating consistent, valid, and physically realizable adversarial perturbations.

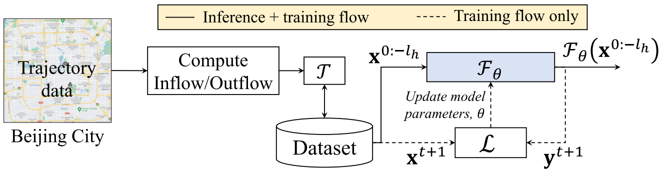


Fig. 3. The training setup of CFP models for the TaxiBJ dataset. The trajectory data collected from the city is first converted into the inflow/outflow matrices, saved in the memory, concatenated with the history set and input to the CFP model.

A. CFS Prediction

1) Task Formulation: The TaxiBJ crowd-flow dataset [3] divides the city into a 2-D grid of size $l_1 \times l_2$, where each grid point physically spans an area of 1000 meters square. Each data sample is a tuple of the city-wide inflow and outflow matrices at 30 minutes interval. At any time t, the integer CFS $\mathbf{n}^t \in \mathbb{Z}^{2\times l_1 \times l_2}$ is defined as a tuple of the integer inflow and outflow matrices, denoted $\mathbf{n}^t_{in} \in \mathbb{Z}^{l_1 \times l_2}$ and $\mathbf{n}^t_{out} \in \mathbb{Z}^{l_1 \times l_2}$, defining the number of devices ($\approx$ persons [2]) flowing into and out of the grid points in the $l_1 \times l_2$ city grid, respectively, where $\mathbb{Z}$ denotes the integer set. Formally,

$$\mathbf{n}^t = (\mathbf{n}^t_{in}, \mathbf{n}^t_{out}) \tag{2}$$

More specifically, for $p_1 \in [0..l_1)$, $p_2 \in [0..l_2)$, $\mathbf{n}^t_{in}(p_1, p_2)$ denotes the total number of devices flowing into the grid point-(p_1, p_2) from the adjacent grid points, and $\mathbf{n}^t_{out}(p_1, p_2)$ denotes the total number of devices outflowing from the grid point-(p_1, p_2) to the adjacent grid points. For example, in Fig. 1, $\mathbf{n}^t_{in}(1, 0) = 2$ (C2 and C3) and $\mathbf{n}^t_{out}(1, 0) = 1$ (C4).

Following prior works, the integer CFS $\mathbf{n}^t$ is transformed into the floating CFS $\mathbf{x}^t \in \mathbb{R}^{2\times l_1 \times l_2}$ (also referred to as the CFS in the future) using a transformation function $\mathcal{T}(\cdot)$.

$$\mathbf{x}^t = (\mathbf{x}^t_{in}, \mathbf{x}^t_{out}) = \mathcal{T}(\mathbf{n}^t) = (\mathcal{T}(\mathbf{n}^t_{in}), \mathcal{T}(\mathbf{n}^t_{out})) \tag{3}$$

$\mathcal{T}(\cdot)$ is chosen such that it is (somewhat) reversible and $\forall \mathbf{n}^t \in [0..\infty], \mathbf{x}^t \in [0, 1]$.[3] Following the prior arts [2], [3], we use $\mathcal{T}(\mathbf{n}^t) = \min(\mathbf{n}^t/1000, 1)$ in our experiments.

Our goal is to learn a model $\mathcal{F}^*_\theta$ that predicts $\mathbf{y}^{t+1}$—the CFS in the immediate future $t+1$—given the current and the previous states $\mathbf{X}_h(t) = \bigcup_{i=0}^{h} \mathbf{x}^{t-i}$, where h is the history length denoting the total number of previous CFS concatenated together with the current state as a tuple input to $\mathcal{F}_\theta$. Following previous works, we solve the above problem as a regression task to learn a model $\mathcal{F}_{\theta^*}$ as formalized below,

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\ \mathbb{E}_{x\sim\mathcal{D}}[(\mathcal{F}_\theta(\mathbf{X}_h(t)) - \mathbf{x}^{t+1})^2] \tag{4}$$

The training setup that we use for training the CFP models $\mathcal{F}_\theta$ is shown in Fig. 3.

[3] Following standard notation, we use $[a..b]$ to denote a set of all integers from a to b, and $[a, b]$ to denote a set of real numbers $\{x,$ such that, $a \leq x \leq b\}$.

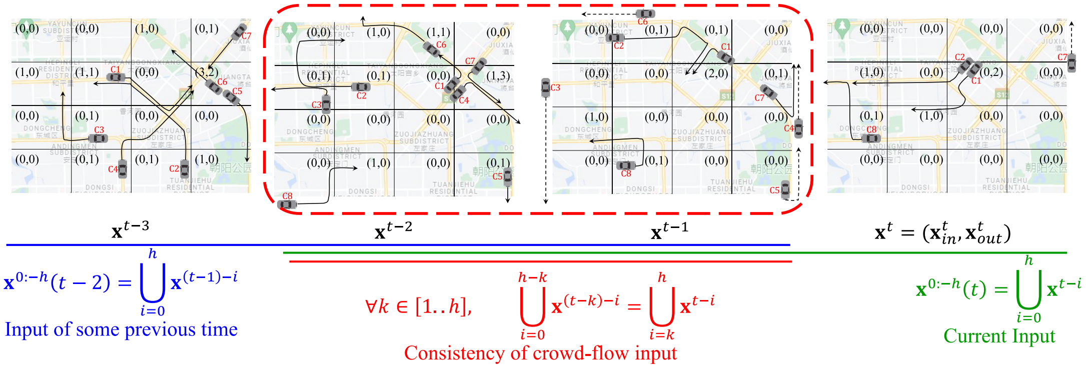

$\mathbf{x}^{t-3}$
$\mathbf{x}^{t-2}$
$\mathbf{x}^{t-1}$
$\mathbf{x}^{t} = (\mathbf{x}^t_{in}, \mathbf{x}^t_{out})$
$\mathbf{x}^{0:-h}(t-2) = \bigcup_{i=0}^{h} \mathbf{x}^{(t-1)-i}$
Input of some previous time
$\forall k \in [1..h], \bigcup_{i=0}^{h-k} \mathbf{x}^{(t-k)-i} = \bigcup_{i=k}^{h} \mathbf{x}^{t-i}$
Consistency of crowd-flow input
$\mathbf{x}^{0:-h}(t) = \bigcup_{i=0}^{h} \mathbf{x}^{t-i}$
Current Input

Fig. 4. An illustration of the *consistency* property of CFS inputs. The CFS history at any time t, must be consistent with the CFS recorded at the previous times.

2) Adversarial Formulation: Given $\mathbf{X}_h(t)$ and $\mathcal{F}^*_\theta(\cdot)$, a typical targeted adversarial attack aims to learn the perturbations, $\mathbf{\Delta}_h(t) = \bigcup_{i=0}^{h} \boldsymbol{\delta}^{t-i} = \bigcup_{i=0}^{h}(\boldsymbol{\delta}^{t-i}_{in}, \boldsymbol{\delta}^{t-i}_{out})$, to $\mathbf{X}_h(t)$ that bring $\mathcal{F}^*_\theta(\mathbf{X}_h(t) + \mathbf{\Delta}_h(t))$ close to the adversarially desired target output, $\mathbf{y}^{t+1}_{target}$.

$$\mathcal{L}_{adv}(\mathbf{X}_h(t), \mathbf{\Delta}_h(t)) = \left|\mathcal{F}_{\theta^*}(\mathbf{X}_h(t) + \mathbf{\Delta}_h(t)) - \mathbf{y}^{t+1}_{target})\right|,$$
$$\mathbf{\Delta}_h(t) = \underset{\mathbf{\Delta}_h(t)\in\mathcal{B}(\epsilon)}{\operatorname{argmax}} \; -\mathcal{L}_{adv}(\mathbf{X}_h(t), \mathbf{\Delta}_h(t)) \quad (5)$$

where $\mathcal{B}(\epsilon)$ denotes a pre-defined bounded set of allowed perturbations.

B. Properties of the CFS Inputs

Here we formally define two key properties of CFP inputs, consistency and validity, which enable the development of *CaV-detect*. We also formally analyze eq-(1) (in specific regards to the aforementioned properties) to highlight the limitations of adversarial attacks against the CFP inputs.

Consistency: We consider a sequence of CFS $\bigcup_{i=0}^{-2h} \mathbf{x}^{t-i}$, recorded at time intervals from $t - 2h$ to t,

$$\bigcup_{i=0}^{2h} \mathbf{x}^{t-i} = \{\mathbf{x}^{t-2h}, \ldots, \mathbf{x}^{t-h}, \ldots, \mathbf{x}^{t}\} \quad (6)$$

The data preprocessing step concatenates the history set containing h previous CFS $\bigcup_{i=1}^{h} \mathbf{x}^{t-i}$ with the current CFS $\mathbf{x}^t$ as tuple input to the CFP model. We note that, for $1 \le k \le h$, the history set at time, t, is a union of a subset of the history set, $\bigcup_{i=1}^{l_h-k} \mathbf{x}^{(t-k)-i}$, and the CFS, $\mathbf{x}^{(t-k)}$, of the model input at time, $t - k$. An input is consistent, if and only if,

$$\forall k \in [1..h], \bigcup_{i=0}^{h-k} \mathbf{x}^{(t-k)-i} = \bigcup_{i=k}^{h} \mathbf{x}^{t-i} \quad (7)$$

which leads to the consistency check mechanism that we develop later. Simply, the history set at any time, t, should be consistent with the CFS at the previous times as illustrated in Fig. 4 with an example.

Remark: For each t, a standard adversarial attack learns a new perturbation $\bigcup_{i=0}^{h} \boldsymbol{\delta}^{t-i}$, independent (and therefore, different) from the perturbation $\bigcup_{i=0}^{h} \boldsymbol{\delta}^{(t-k)-i}$ learned for some previous time $t - k$. Formally, for standard adversarial attacks,

$$\forall k \in [1..h], \bigcup_{i=k}^{h} \boldsymbol{\delta}^{t-i} \neq \bigcup_{i=0}^{h-k} \boldsymbol{\delta}^{t-k-j} \quad (8)$$

Stated simply, the adversarial perturbations (and hence, the adversarially perturbed inputs), generated by the standard adversarial attacks are *inconsistent*, and therefore, can be detected effectively.

Validity: Consider a 4 × 4 grid shown in Fig. 1. For a grid point-(1, 0), (shaded green) the total number of devices entering into the grid point from its adjacent grid points (shaded yellow) is 2 (C2 and C3). As these devices must outflow from the adjacent grid points, validity requires that the total outflow from the adjacent grid points must atleast be 2. In Fig. 1, the total outflow from the adjacent grid points is 3 (C1, C2 and C3), which is greater than 2 (the inflow). Similarly, the total inflow to the adjacent grid points is 2 (C1 and C4), which is greater than 1—outflow from the grid point-(1, 0) (C4).

More generally, given a specific grid point-(p_1, p_2), let $A_n(p_1, p_2)$ denote a set of grid points adjacent to the grid point-(p_1, p_2) in the n^{th} neighborhood,

$$A_n(p_1, p_2) = \bigcup_{i=-n}^{n} \bigcup_{\substack{j=-n \\ |i|+|j|\neq 0}}^{n} (p_1 - i, p_2 - j) \quad (9)$$

where n is the size of neighborhood considered for adjacency. In our experiments, we heuristically choose $n = 2$ as detailed later. By definition, at any time t, the inflow to the grid point-(p_1, p_2), is the total number of devices entering into that grid point from its adjacent grid points $A_n(p_1, p_2)$. Therefore, the total outflow from $A_n(p_1, p_2)$ must be atleast equal to the total inflow to (p_1, p_2). Let $\mathbf{x}^t_{in}(p_1, p_2)$ and $\mathbf{x}^t_{out}(p_1, p_2)$ respectively denote the inflow to and outflow from the grid point-(p_1, p_2) at time, t. Any input to the CFP model is only *valid*, if,

$$\mathbf{x}^t_{in}(p_1, p_2) \le \sum_{(p_1', p_2')\in A_n(p_1,p_2)} \mathbf{x}^t_{out}(p_1', p_2') \quad (10a)$$
$$\mathbf{x}^t_{out}(p_1, p_2) \le \sum_{(p_1', p_2')\in A_n(p_1,p_2)} \mathbf{x}^t_{in}(p_1', p_2') \quad (10b)$$

Although, there can be inflow to (and outflow from) $A_n(p_1, p_2)$ from (and to) outside $A_n(p_1, p_2)$, this only adds to the total inflow to (and outflow from) $A_n(p_1, p_2)$ (right hand sides of eq-(10)), and therefore does not affect the generality of our formulation.

Remark: While generating perturbations, standard adversarial attacks formalized in eq-(1) do not respect the validity

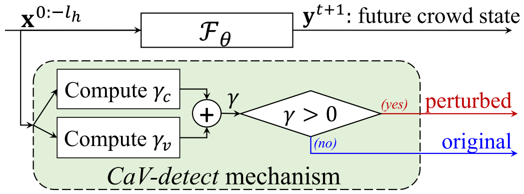


Fig. 5. Illustrating our newly proposed *CaV-detect* methodology integrated with the CFP model to detect adversarial inputs at run-time. For any input, *CaV-detect* checks the consistency, $\gamma_c > 0$ and the validity $\gamma^n_v > 0$ of the input. The input is marked adversarial if any of the checks fail. CaV-detect *does not require retraining the model and can be integrated with an off-the-shelf model.*

between the inflow and outflow matrices, and therefore, can be detected at run-time.

C. CaV-detect*: Consistency and Validity Check Mechanism to Detect Adversarially Perturbed Inputs*

Here we utilize the previously defined two properties of crowd-flow inputs to propose *CaV-detect*, a novel input validation mechanism to detect adversarially perturbed inputs to the CFP models. To summarize, our *CaV-detect* methodology comprises two main steps—*consistency* check mechanism and *validity* check mechanism. An input to the model is considered adversarially perturbed if it fails to satisfy either of the aforementioned checks. Step-by-step details of *CaV-detect* are given below.

1) Consistency Check Mechanism: At any time t, the input to the CFP model $\mathcal{F}_{\theta^*}$ is $\mathbf{X}_h(t) = \bigcup_{i=0}^{h} \mathbf{x}^{t-i}$. Our consistency check mechanism works in three steps:

1) Firstly, we keep all the model inputs, $\mathbf{x}^{(t-k)}$, received at the previous times, $t-k$, saved in the memory, $\forall k \in [1..h]$.
2) Noting that the model inputs received at the previous times, $t-k$, reappear in the history set of the input received at the current time, t, we compute the difference between appropriately cropped model inputs at different times

$$\gamma_c = \sum_{k=1}^{h} \left| \bigcup_{i=k}^{h} \mathbf{x}^{t-i} - \bigcup_{i=0}^{h-k} \mathbf{x}^{(t-k)-i} \right| \geq 0 \tag{11}$$

where γ_c is the inconsistency score—the closer γ_c is to zero, the more consistent the input.
3) The input is marked as adversarial if $\gamma_c > 0$.

Remark: Although the consistency check mechanism essentially compares variables at different times, it is necessary to include the consistency (and validity) check mechanism in the CFP model itself. This is because in the digital setting, a digital attacker may directly influence the originally consistent input to the CFP model (See Fig. 7).

2) Validity Check Mechanism: Our validity check mechanism works in four steps described below.

1) Given n^{th} neighborhood, we first define a filter, $\mathbf{f}^n \in \mathbb{Z}^{(2n+1)\times(2n+1)}$, such that $\forall p_1, p_2 \in [0..2n]$,

$$\mathbf{f}^n(p_1, p_2) = \begin{cases} 1, & p_1 = n, p_2 = n \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

2) Secondly, we compute the inflow and outflow invalidity scores, denoted $\gamma^n_{v_i}$ and $\gamma^n_{v_o}$ respectively, by simultaneously analyzing both the inflow and outflow matrices in the input.

$$\gamma^n_{v_i} = \mathbf{x}^{t-i}_{in} \circledast \mathbf{f}^n - \mathbf{x}^{t-i}_{out} \circledast (\mathbf{1} - \mathbf{f}^n) \leq \mathbf{0} \tag{13}$$

$$\gamma^n_{v_o} = \mathbf{x}^{t-i}_{out} \circledast \mathbf{f}^n - \mathbf{x}^{t-i}_{in} \circledast (\mathbf{1} - \mathbf{f}^n) \leq \mathbf{0} \tag{14}$$

where $\circledast$ denotes a 2-D convolution operation.
3) Finally, we compute the input invalidity score, γ^n_v, based on the inflow and outflow invalidity scores computed in step 2.

$$\gamma^n_v = \text{relu}(\gamma^n_{v_i} + \gamma^n_{v_o}) \tag{15}$$

where relu denotes the rectified linear unit function commonly used in DL literature.
4) The input is marked as adversarial if $\gamma^n_v > 0$.

Note that both the check mechanisms used by *CaV-detect* are model agnostic. Therefore, *CaV-detect* can be incorporated with the pre-trained CFP models of varying architectures without undermining their efficacy.

D. CVP *Attack: Consistent Valid and Physically-Realizable Adversarial Attack Against CFP Models*

In light of the previously formalized practical limitations of standard adversarial attacks, in this section, we propose *CVP* attack—a *consistent*, *valid* and *physically realizable* adversarial attack. Recall that at any time t, we consider an input $\mathbf{X}_h(t) = \bigcup_{i=0}^{h} \mathbf{x}^{t-i}$ to the model $\mathcal{F}^*_\theta$. Our goal is to generate perturbations $\boldsymbol{\Delta}_h(t) = \bigcup_{i=0}^{h} \boldsymbol{\delta}^{t-i}$ to the input in order to bring the model output closer to the adversarial target $\mathbf{y}^{t+1}_{target}$.

Consistency: To ensure consistency in the perturbations, we leverage universal adversarial perturbations to regulate $\boldsymbol{\Delta}_h(t)$, such that, $\forall i \in [0..h]$, $\boldsymbol{\delta}^{t-i} = \boldsymbol{\delta}^u$.

$$\boldsymbol{\Delta}_h(t) = \bigcup_{i=0}^{h} \boldsymbol{\delta}^u = \bigcup_{i=0}^{h} (\boldsymbol{\delta}^u_{in}, \boldsymbol{\delta}^u_{out}) \tag{16}$$

Validity: To ensure validity, we introduce a novel mechanism to generate the *perturbation outflow matrix* $\boldsymbol{\delta}^u_{out}$, given a *perturbation inflow matrix* $\boldsymbol{\delta}^u_{in}$. More specifically, given $\boldsymbol{\delta}^u_{in}$, a specific grid point-$(p_1, p_2)$, and a set of its adjacent grid points in the n^{th} neighborhood $A_n(p_1, p_2)$, we learn a perturbation distribution matrix $\mathbf{W} \in \mathbb{R}^{l_1 \times l_2 \times (2n+1)^2 - 1}$ to first distribute the (perturbed) inflow to grid point-(p_1, p_2) among $A_n(p_1, p_2)$. To achieve this, we first process $\mathbf{W}$ with a sigmoid function $\sigma(\cdot)$ to differentiably remove the negative values of $\mathbf{W}$. The processed matrix $\sigma(\mathbf{W})$ is then normalized so that it sums to 1 for $A_n(p_1, p_2)$. The normalized matrix is then multiplied with the inflow of grid point-(p_1, p_2) to compute the outflow of each adjacent grid point.

$$\boldsymbol{\delta}^*_{out} = \boldsymbol{\delta}^u_{in} \odot \frac{\sigma(\mathbf{W})}{\sum_i \sigma(\mathbf{W})[:, :, i]} \tag{17}$$

where $\odot$ denotes the element-wise (Hadamard) multiplication and $\boldsymbol{\delta}^*_{out} \in \mathbb{R}^{l_1 \times l_2 \times (2n+1)^2 - 1}$ is a set of distributed perturbation outflow matrices for $\boldsymbol{\delta}^*_{in}$ satisfying the validity condition of crowd-flow inputs. The total perturbation outflow for the grid point-(p_1, p_2) is then computed by accumulating relevant distributed outflows,

$$\boldsymbol{\delta}^u_{out}(p_1, p_2) = \sum_{i=-n}^{n} \sum_{\substack{j=-n \\ |i|+|j| \neq 0}}^{n} \boldsymbol{\delta}^*_{out}(p_1 - i, p_2 - j, k) \tag{18}$$

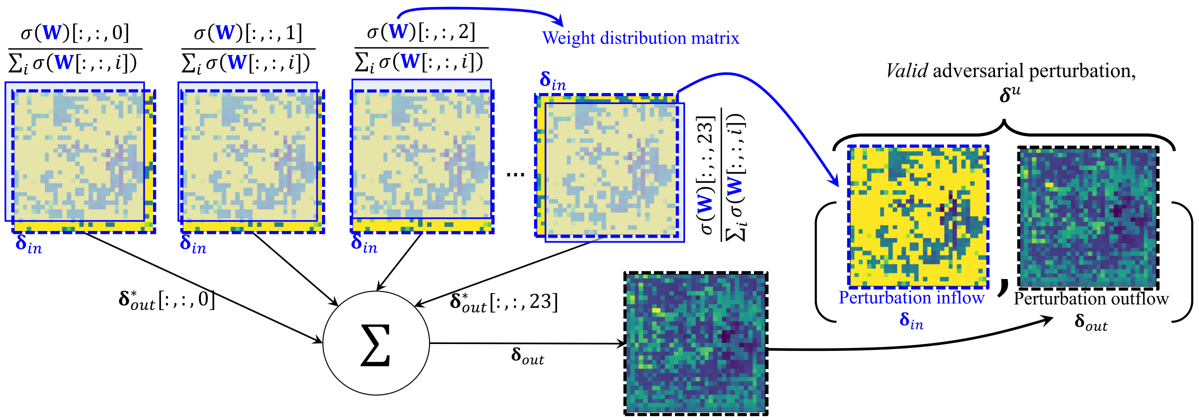

$\frac{\sigma(\mathbf{W})[:,:,0]}{\sum_i \sigma(\mathbf{W}[:,:,i])}$
$\frac{\sigma(\mathbf{W})[:,:,1]}{\sum_i \sigma(\mathbf{W}[:,:,i])}$
$\frac{\sigma(\mathbf{W})[:,:,2]}{\sum_i \sigma(\mathbf{W}[:,:,i])}$
Weight distribution matrix
$\boldsymbol{\delta}_{in}$
$\frac{\sigma(\mathbf{W})[:,:,23]}{\sum_i \sigma(\mathbf{W}[:,:,i])}$
...
Valid adversarial perturbation,
$\boldsymbol{\delta}^u$
$\boldsymbol{\delta}^*_{out}[:,:,0]$
$\boldsymbol{\delta}^*_{out}[:,:,23]$
Σ
$\boldsymbol{\delta}_{out}$
Perturbation inflow
$\boldsymbol{\delta}_{in}$
Perturbation outflow
$\boldsymbol{\delta}_{out}$

Fig. 6. Illustrating the newly proposed *CVP* attack methodology of generating valid adversarial perturbations. Terms highlighted in blue denote the variables that are updated during attack to optimize the adversarial loss. Given a *perturbation inflow matrix*, $\boldsymbol{\delta}_{in}$, a set of *distributed perturbation outflow matrices*, $\boldsymbol{\delta}^*_{out}$ is computed by element-wise application of $\boldsymbol{\delta}_{in}$ and the appropriately shifted normalized perturbation distribution matrices, $\mathbf{W}$. Finally, the *perturbation outflow matrix*, $\boldsymbol{\delta}_{out}$, is computed by adding all the slices of $\boldsymbol{\delta}^*_{out}$.

where k is defined as,

$$k = \begin{cases} (2n+1)(i+n)+j+n-1, & i>0, j>0 \\ (2n+1)(i+n)+j+n, & \text{otherwise} \end{cases} \tag{19}$$

In other words, $\boldsymbol{\delta}_{out}$ is computed as a function, f, of $\boldsymbol{\delta}_{in}$ and $\mathbf{W}$ as illustrated in Fig. 6. Eq-(16) can then be re-written as,

$$\boldsymbol{\Delta}_h(t) = \bigcup_{i=0}^{h}(\boldsymbol{\delta}^u_{in}, \boldsymbol{\delta}^u_{out}) = \bigcup_{i=0}^{h}(\boldsymbol{\delta}^u_{in}, f(\boldsymbol{\delta}^u_{in}, \mathbf{W})) \tag{20}$$

Physical Realizability: Perturbation $\boldsymbol{\Delta}_h(t) = \bigcup_{i=0}^{h} \boldsymbol{\delta}^{t-i}$ learned by standard adversarial attacks at time t is different for each i. In practice, such attacks are only feasible under the digital attack setting. Realizing such attacks under the physical setting requires an attacker to precisely control the number of devices in each grid point, which is challenging because an attacker has to either repeatedly relocate the adversarial devices or have a sufficient number of adversarial devices repeatedly switched on and off to simulate $\boldsymbol{\Delta}_h(t)$. Universal adversarial perturbation naturally addresses this by generating a single most effective perturbation for each time interval.

Additionally, generating $\boldsymbol{\delta}^u \in \mathcal{B}_\infty(\epsilon)$ ball only works under the digital setting. For physical setting, an attacker can only realize $\boldsymbol{\delta} > 0$ perturbations (for example, by physically adding a certain number of adversarial devices). For example, in Fig. 2(a) PGD attacker tries to reduce the inflow to the grid point-(2, 1) by stopping C1—a real device, not controlled by the attacker—from flowing into the grid point, which is impractical. Therefore, for physical attacks, we optimize the perturbations for $\mathcal{B}_\infty(0, \epsilon)$ bound—an attacker may add controllable physical devices into the crowd, but may not control the original devices. To summarize, a physical *CVP* attacker may move a preset number of adversarial devices repeatedly along the computed optimal paths to simulate consistent, valid, positive and universal adversarial perturbation in the physical world.

While generating the perturbations $\boldsymbol{\delta}^u$, we iteratively update $\boldsymbol{\delta}_{in}$ and $\mathbf{W}$ to find the optimal perturbations. More specifically, we optimize the following loss function,

Algorithm 1 *CVP* Attack Algorithm

```
Input:
    $\mathbf{X}_h(t) \leftarrow$ history of CFS
    $\mathcal{F}_{\theta^*} \leftarrow$ trained model
    $\mathbf{y} \leftarrow$ output CFS
    $N \leftarrow$ # of iterations
    $\epsilon \leftarrow$ maximum perturbation budget
Output:
    $\boldsymbol{\delta}^u \leftarrow (\boldsymbol{\delta}^u_{in}, \boldsymbol{\delta}^u_{out}) \leftarrow$ universal adversarial perturbations
Define $i \leftarrow 1, \boldsymbol{\delta}_{in} \leftarrow \mathbf{0}, \mathbf{W} \leftarrow \mathbf{-5}$
Define $\eta \leftarrow (5 \times \epsilon)/N$
repeat
    $\boldsymbol{\delta}^u \leftarrow (\boldsymbol{\delta}^u_{in}, f(\boldsymbol{\delta}^u_{in}, \mathbf{W}))$
    $\mathcal{L}_{adv} \leftarrow \left(\mathcal{F}_{\theta^*}\left(\mathbf{X}_h(t) + \bigcup_{i=0}^{h} \boldsymbol{\delta}^u\right) - \mathbf{y}^{t+1}_{target}\right)^2$
    $\boldsymbol{\delta}^u_{in} \leftarrow \boldsymbol{\delta}^u_{in} - \eta \times \text{sign}\left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{\delta}^u_{in}}\right)$
    $\mathbf{W} \leftarrow \mathbf{W} - \eta \times \text{sign}\left(\frac{\partial \mathcal{L}}{\partial \mathbf{W}}\right)$
    $\boldsymbol{\delta}^u_{in} \leftarrow \text{clip}(-\epsilon, \epsilon)$ —— ($l_\infty$ bound)
    $\boldsymbol{\delta}^u_{out} \leftarrow f(\boldsymbol{\delta}^u_{in}, \mathbf{W})$
    $i \leftarrow i + 1$
until $i \leq N$
```

$$\mathcal{L}_{upa}\left(\mathbf{X}_h(t), (\boldsymbol{\delta}^u_{in}, \mathbf{W})\right) = \left|\mathcal{F}_{\theta^*}\left(\mathbf{X}_h(t) + \bigcup_{i=0}^{h} \boldsymbol{\delta}^u\right) - \mathbf{y}^{t+1}_{target}\right|,$$
$$\boldsymbol{\delta}^u_{in}, \mathbf{W} = \underset{\boldsymbol{\delta}^u_{in} \in \mathcal{B}(\epsilon), \mathbf{W}}{\text{argmax}} -\mathcal{L}_{upa}\left(\mathbf{X}_h(t), (\boldsymbol{\delta}^u_{in}, \mathbf{W})\right) \tag{21}$$

where $\boldsymbol{\delta}^u = (\boldsymbol{\delta}^u_{in}, \boldsymbol{\delta}^u_{out})$ denotes the universal adversarial perturbations that remain constant for all $\mathbf{x}^t \in \mathcal{D}$, ϵ is the maximum perturbation budget as discussed previously. For physical realizability (limitation 3), we repeatedly clip the negative values and project $\boldsymbol{\delta}^u$ on $\mathcal{B}(\epsilon)$ ball while maximizing $\mathcal{L}_{adv}$ using gradient-descent. Details are given in Algirthm 1.

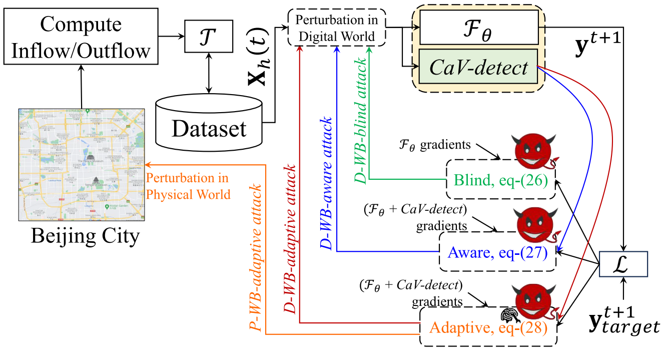


Fig. 7. Illustration and comparison of different white-box threat models used in our experiments. D-WB denotes a digital attack setting under the white-box threat model. P-WB denotes a physical attack setting under white-box threat model. An adaptive attacker adapts the perturbation generation mechanism to fool *CaV-detect*.

IV. Experimental Setup

A. Threat Models

For all the experiments in this paper, we assume a white-box threat model in which the attacker has complete knowledge of the CFP model architecture and its learned parameters θ^*. Further, we always assume a targeted attack scenario where the goal of a white-box attacker is to learn the perturbations $\mathbf{\Delta}_h(t) \in \mathcal{B}(\epsilon)$, that, when added to the inputs, produce the maximum relevance to an attacker's defined target state, $\mathbf{y}^{t+1}_{target}$.

$$\mathcal{L}_{adv}\left(\mathbf{X}_h(t), \mathbf{\Delta}_h(t)\right) = \left|\mathcal{F}_{\theta^*}(\mathbf{X}_h(t) + \mathbf{\Delta}_h(t)) - \mathbf{y}^{t+1}_{target})\right| \quad (22)$$

We experiment with three different white-box threat configurations as detailed below, and illustrated in Fig. 7.

1) WB-Blind Threat Model: In this white-box threat model, the attacker is assumed to be unaware of the *CaV-detect* mechanism deployed in the pipeline. The goal of WB-blind attacker is formalized below,

$$\mathbf{\Delta}_h(t) = \underset{\mathbf{\Delta}_h(t) \in \mathcal{B}_\infty(\epsilon)}{\operatorname{argmax}} \; -\mathcal{L}_{adv}\left(\mathbf{X}_h(t), \mathbf{\Delta}_h(t)\right) \quad (23)$$

where $\mathcal{B}_\infty(\epsilon)$ is an l_∞ ball, defined as, $\boldsymbol{\delta} \in \mathcal{B}_\infty(\epsilon) := -\epsilon \leq \boldsymbol{\delta} \leq \epsilon$.

2) WB-Aware Threat Model: This white-box threat model assumes that the attacker is fully aware of *CaV-detect* mechanism in the pipeline, and tries to evade the detection by *CaV-detect* while simultaneously trying to produce the target state at the model output. Formally, we define a Lagrange function,

$$\mathbf{\Delta}_h(t) = \underset{\mathbf{\Delta}_h(t) \in \mathcal{B}(\epsilon)}{\operatorname{argmax}} \; -\mathcal{L}_{adv}\left(\mathbf{X}_h(t), \mathbf{\Delta}_h(t)\right) - \lambda \times \left(\gamma_c + \gamma_v^n\right) \quad (24)$$

where λ is the Lagrange multiplier. We set $\lambda = 10^{10}$, to strictly meet the validity and the consistency condition.

Note: We conduct experiments under the WB-aware threat model and discover that a WB-aware attacker is unable to cause any considerable change in the model output. We conjecture that the strict consistency and validity check mechanism leads to contradicting gradient updates when optimizing eq-(24). Therefore, we do not report the quantitative results in the paper.

3) WB-Adaptive Threat Model: Similar to WB-aware, this threat model also assumes an attacker fully aware of *CaV-detect* mechanism in the pipeline and tries to evade the detection by *CaV-detect* while changing the output towards the target state *by adaptively modifying the attack algorithm*. More specifically, an adaptive attacker modifies the attack to make it easier for eq-(24) to be optimized. In order to achieve this, our adaptive attacker leverages the algorithm of universal adversarial perturbations to naturally evade the consistency check mechanism, while optimizing the following adaptive loss function,

$$\boldsymbol{\delta}^u = \underset{\boldsymbol{\delta}^u \in \mathcal{B}(\epsilon)}{\operatorname{argmax}} \; -\mathcal{L}_{adv}\left(\mathbf{X}_h(t), \bigcup_{i=0}^{h} \boldsymbol{\delta}^u\right) - \lambda \times \gamma_v^n \quad (25)$$

where we set $\lambda = 10^{10}$.

4) Digital and Physical Settings: In addition to the threat models mentioned above, we consider two different attack settings—digital and physical. The digital attack setting (D-WB) depicts a typical white-box scenario where an attacker can hack into the inference pipeline to directly perturb the digital input to the model. On the contrary, the physical attack setting (P-WB) depicts a more realistic white-box scenario where an attacker knows $\mathcal{F}_{\theta^*}$, but cannot directly perturb the digital input to the model. Instead, a physical attacker has to optimally control a specific number of devices, called adversarial devices, at specific grid points in order to realize adversarial perturbations. P-WB restricts an attacker by only allowing physical perturbations, which makes it more practical than D-WB.

B. Adversarial Attacks

For D-WB settings, we evaluate three standard adversarial attacks—FGSM, i-FGSM, and PGD attacks—on our trained models. FGSM attack is simple, fast, and generates transferable adversarial perturbations, which makes it a good choice for our evaluation. On the other hand, PGD is among the strongest adversarial attacks found in literature against non-obfuscated models such as the ones we use in our evaluations. For P-WB settings, we compare the aforementioned attacks with our newly proposed *CVP* attack on different model architectures.

C. Evaluation Metrics

1) Test Loss: To evaluate $\mathcal{F}_{\theta^*}$ on some test data $\mathcal{D}_{test}$, we use a commonly used metric, the mean square error (MSE), that captures the distance of the model output from the ground truth $\mathbf{x}^{t+1}$, as defined below,

$$\mathcal{L}(\mathcal{D}_{test}) = \frac{1}{|\mathcal{D}_{test}|} \sum_{\forall \mathbf{X}_h(t) \in \mathcal{D}_{test}} \left(\mathcal{F}_{\theta^*}(\mathbf{X}_h(t)) - \mathbf{x}^{t+1}\right)^2 \quad (26)$$

A smaller value of $\mathcal{L}(\mathcal{D}_{test})$ indicates a better learned model.

2) Adversarial Loss: For the adversarial evaluation, we let $\mathcal{D}^*_{test}$ denote the adversarially perturbed test data and compute an adversarial MSE, denoted $\mathcal{L}^*(\mathcal{D}^*_{test})$, as a measure of the model's robustness.

$$\mathcal{L}^*(\mathcal{D}^*_{test}) = \frac{1}{|\mathcal{D}^*_{test}|} \sum_{\forall \mathbf{X}^*_h(t) \in \mathcal{D}^*_{test}} \left(\mathcal{F}_{\theta^*}(\mathbf{X}^*_h(t)) - \mathbf{y}^{t+1}_{target}\right)^2 \quad (27)$$

where $\mathbf{y}^{t+1}_{target}$ denotes the target output desired by an attacker, and $\mathbf{X}^*_h(t) = \mathbf{X}_h(t) + \mathbf{\Delta}_h(t)$ is the perturbed input. A larger $\mathcal{L}^*(\mathcal{D}^*_{test})$ indicates that $\mathcal{F}_{\theta^*}(\mathbf{X}^*_h(t))$ is considerably different from $\mathbf{y}^{t+1}_{target}$, and therefore $\mathcal{F}_{\theta^*}$ is more robust to the perturbations in the input. On the contrary, a smaller $\mathcal{L}^*(\mathcal{D}^*_{test})$ indicates that $\mathcal{F}_{\theta^*}(\mathbf{X}^*_h(t))$ closely matches $\mathbf{y}^{t+1}_{target}$ (target output achieved), signifying that $\mathcal{F}_{\theta^*}$ is less robust to the perturbations in the input.

3) False Acceptance Rate (FAR): False Rejection Rate is defined as the percentage of original (unperturbed) inputs marked as perturbed by the detector to the total number of original inputs. Formally, given n,

$$\text{FRR}(n) = \frac{|\mathbf{X}_h(t) \in \mathcal{D}_{test}, \text{ s.t. } \gamma_c > 0, \gamma^n_v > 0|}{|\mathbf{X}_h(t) \in \mathcal{D}_{test}|} \times 100\% \tag{28}$$

To evaluate the efficacy of *CaV-detect* in capturing adversarial inputs, we use a widely used metric called FAR defined as the percentage of adversarial inputs marked unperturbed by the detector to the total number of adversarial inputs generated by the attacker. Formally, given n, FAR(n) is defined as,

$$\text{FAR}(n) = \frac{|\mathbf{X}^*_h(t) \in \mathcal{D}^*_{test}, \text{ s.t. } \gamma_c = 0, \gamma^n_v = 0|}{|\mathbf{X}^*_h(t) \in \mathcal{D}^*_{test}|} \times 100\% \tag{29}$$

In our experiments, we heuristically choose the minimum value of n (the adjacency number) such that the FRR$(n) \leq 0.5\%$ (See Table I). In our experiments, we use $n = 2$. Additionally, we also use FAR(n) to quantify the efficacy of adversarial attacks to evade the detection by *CaV-detect*. A greater FAR(n) indicates that the attack is stealthier and appears more benign to *CaV-detect*.

D. Hyperparameters

In this subsection, we report key hyperparameters that we analyze to understand the performance of the model under both, the standard and the adversarial scenarios.

Data: While preparing the dataset, we use the history length, $h \in \{2, 5, 10, 15, 20\}$.

Models: We train different models based on the MLP and STResnet architectures by varying the number of hidden layers of each model. Specifically, our MLP-l architecture is defined as: **Input () – {FC (512) – ReLU ()}**$\times l$ **– FC (output shape) – Sigmoid** (), where **FC()** denotes a fully connected layer. We use l in $\{3, 5, 10\}$ for MLP and denote the models as MLP-3, MLP-5 and MLP-10 respectively. Similarly, our STResnet-l architecture from Zhang et al. [2] is defined as: **Input () – Conv2D (64, (7, 7)) – ReLU () – {residual block()}**$\times l$ **– Conv2D (10, (7, 7)) – ReLU () – FC (output shape) – Sigmoid** (), where each **residual block()** contains two Conv2D layers. We use l in $\{1, 2, 3\}$ for STResnet and denote the models as STResnet-1, STResnet-2 and STResnet-3 respectively. As such, our STResnet-2 contains a total of 9 layers—1 input layer, 6 2D convolution layers (+6 activation layers), 1 fully connected layer and 1 output layer with the sigmoid activation. Similarly, STResnet-1 and STResnet-3 are comprised of 7 and 11 layers. For a TGCN model, we experiment with different dimensions of the hidden messages in $\{1, 3, 5, 10\}$ (See [12] for the definition of hidden messages in the TGCN model) and study the effect of changing the number of neighbors, $d_A \in \{1, 3, 5, 10\}$ on the accuracy of TGCN models, where d_A denotes the number of adjacent nodes of TGCN model assumed to be able to communicate with each other. For future reference, we denote *tgcn*-(m, d_A) as a TGCN model with m dimensional hidden messages and d_A node connectivity.

Attacks: For adversarial evaluation, we experiment with $\epsilon \in \{0.01, 0.03, 0.05, 0.07, 0.1\}$ (for digital attack settings) and the adversarial device budget $b_d \in \{1000, 3000, 5000, 7000, 10000\}$ (for physical attack settings). We also analyze the effects of changing the number of attack iterations, $N \in \{100, 250, 500, 750, 1000\}$ on the performance of the aforementioned model. For future references, we denote a specific attack setting as Attack-(ϵ, N). For example, PGD-$(0.1, 1000)$ denotes a PGD attacker with the maximum allowed perturbation of 0.1 and an iteration budge of 1000.

V. Results

We first establish the baselines by reporting mean square loss over the original/unperturbed inputs. We then evaluate these models under the standard adversarial attacks and the newly proposed *CVP* attack. Finally, we show the efficacy of *CVP* attack over the standard adversarial attacks by comparing the number of adversarial devices required by each to achieve the adversarial goal.

A. Performance of Prediction Models

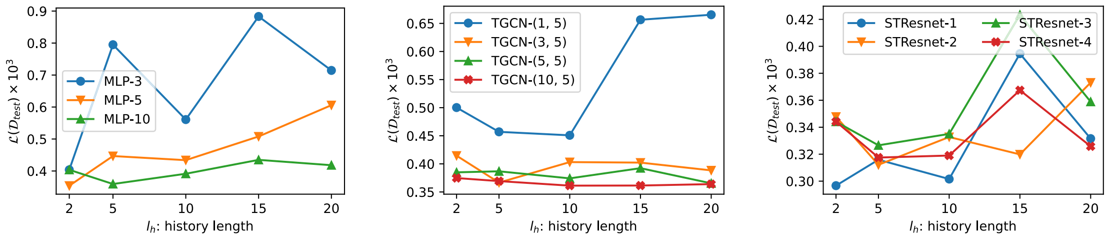
Fig. 8(a, b and c) compare $\mathcal{L}(\mathcal{D}_{test})$ over unperturbed test inputs of MLP, TGCN and STResnet models trained on TaxiBJ-16 dataset with different history lengths. We do not observe any strict relationship between the complexity of a model and its performance over unperturbed test inputs. STResnet models generally perform better than MLP and TGCN models which can be attributed to their spatio-temporal architecture. Although TGCN models also capture spatio-temporal patterns in data, they have far fewer parameters as compared to STResnet models.

In Fig. 8, as h is increased, $\mathcal{L}(\mathcal{D}_{test})$ typically slightly increases for all architectures, with occasional exceptions. We conjecture that a greater h increases the input information to the model, which may lead to mutually contradicting gradient updates during training, causing the resulting model to underfit. For relatively simpler models that are already vulnerable to underfitting, the increase in $\mathcal{L}(\mathcal{D}_{test})$ with the increase in h is more significant, which further validates our hypothesis.

Performance of CaV-detect on Original Inputs: Table I shows the FRR of original inputs by *CaV-detect* as function of n (see eq-(9)). For $n = 1$, 42.5% of original inputs are marked as adversarial by *CaV-detect*, while for $n = 2$ FRR(n) drops to 0%. This indicates that for the TaxiBJ-16 dataset, $n = 1$ is too small to give a good performance. On the other hand, $n = 2$ successfully satisfies the validity of all the inputs in the dataset. Overall, increasing n makes the definition of validity looser, which makes *CaV-detect* more tolerant towards perturbations in the input.

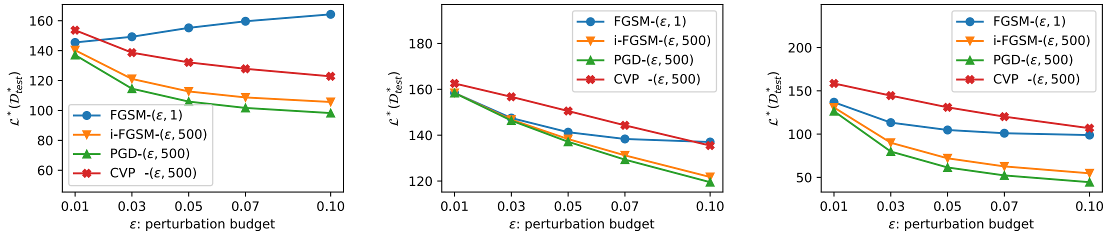
B. D-WB-Blind Adversarial Attacks

Fig. 9(a-c) summarizes our results of adversarial attacks on the CFP models of different architectures for



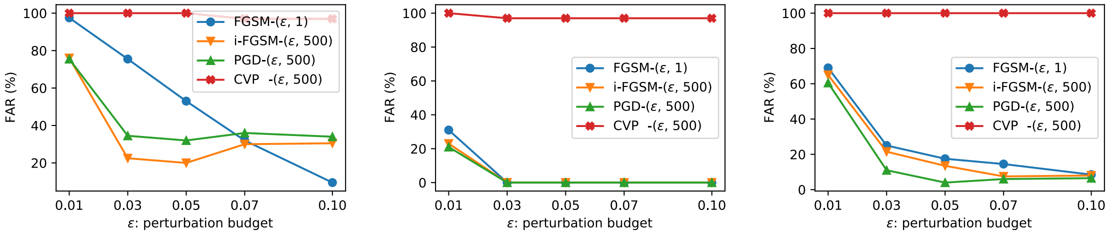
(a) Multi-Layer Perceptron (b) Temporal Graph Convolution Net (c) Spatio-Temporal ResNet

Fig. 8. A comparison of **the model loss**, $\mathcal{L}(\mathcal{D}_{test})$ (eq-(26)), over **the original/unperturbed test set**, $\mathcal{D}_{test}$, for different model complexities as the predefined history length, h, is increased. (Settings: Dataset is TaxiBJ-16). *No strict relationship between the model complexity and its performance over $\mathcal{D}_{test}$ is observed. When the input history length is increased, the $\mathcal{L}(\mathcal{D}_{test})$ increases in a slight manner, indicating a decrease in model performance. Of the three architectures, STResnet performs best.*



(a) MLP-5 (b) TGCN-(5,5) (c) STResnet-2

Fig. 9. Comparing **the adversarial loss**, $\mathcal{L}^*(\mathcal{D}^*_{test})$ (eq-(27)), over **the perturbed dataset**, $\mathcal{D}^*_{test}$, by different attacks for different model architectures as ϵ is increased assuming **a D-WB-blind attacker**. (Settings: Dataset is TaxiBJ-16; h is 5.). *Deep CFP models are vulnerable to adversarial attacks. Increasing ϵ makes the attack stronger. TGCN-(5,5) is the most robust of the considered architectures.*



(a) MLP-5 (b) TGCN-(5,5) (c) STResnet-2

Fig. 10. False acceptance rate (FAR) of *CaV-detect* mechanism against the perturbed inputs, $\mathcal{D}^*_{test}$, generated by **a D-WB-blind attacker**. (Settings: Dataset is TaxiBJ-16. h is 5). *The adversarial perturbations become increasingly invalid as ϵ increases. FAR of the consistency check mechanism is always 0%, so we only report FAR of the validity-check mechanism.*

TABLE I

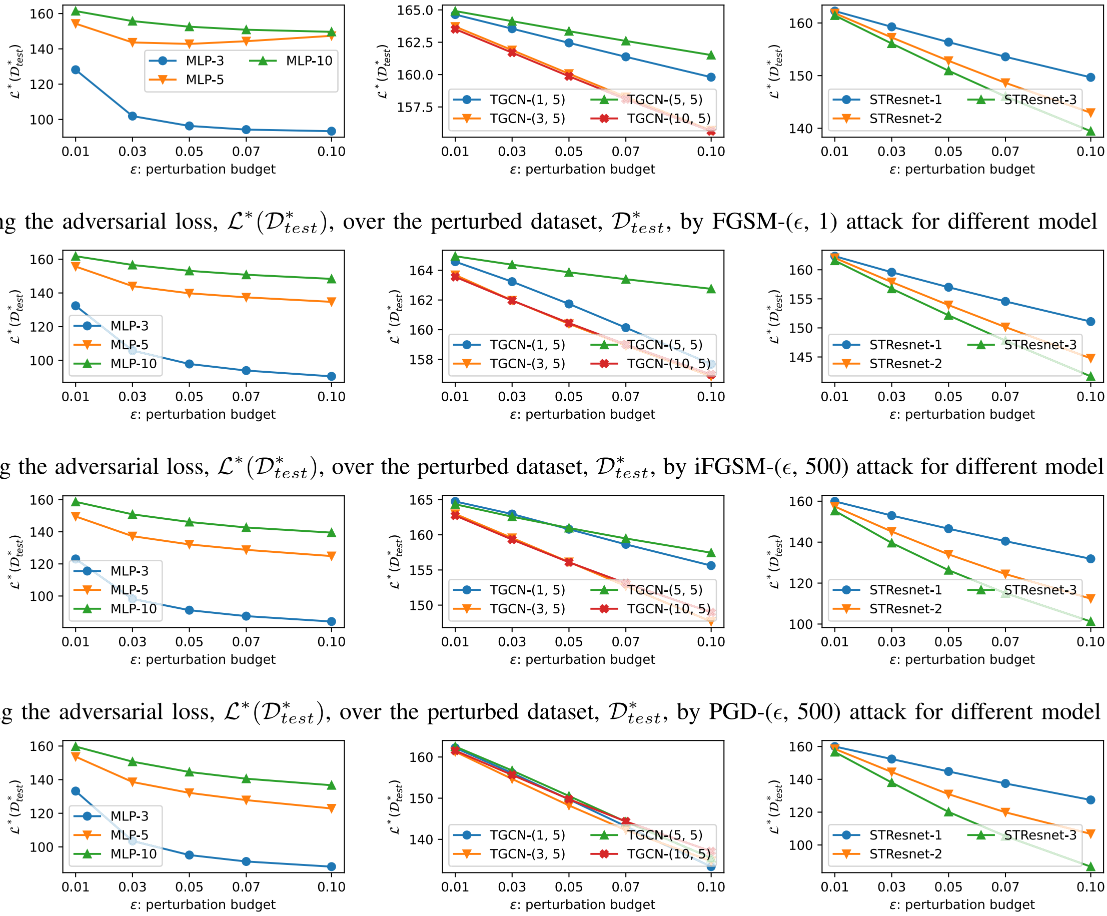
COMPARING *CaV-detect* MECHANISM'S FALSE REJECTION RATE (FRR) ON ORIGINAL INPUTS AND FALSE ACCEPTANCE RATE (FAR) ON PERTURBED INPUTS BY DIFFERENT ATTACK ALGORITHMS ASSUMING D-WB-BLIND AND D-WB-ADAPTIVE ATTACKERS AS A FUNCTION OF n, WHERE n DENOTES THE ASSUMED NEIGHBORHOOD IN EQ-(9). (SETTINGS: ALL ATTACKS ASSUME $\epsilon = 0.1$, $N = 500$)

Model	n	**FRR(n)** %	**D-WB-blind attacker, FAR(n) %**				**D-WB-adaptive attacker, FAR(n) %**			
			FGSM	**i-FGSM**	**PGD**	***CVP***	**FGSM**	**i-FGSM**	**PGD**	***CVP***
MLP-5	1	42.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	2	0.0	0.0	0.0	0.0	100.0	80.5	90.0	93.5	100.0
	3	0.0	0.0	0.0	0.0	100.0	100.0	100.0	100.0	100.0
TGCN-(5, 5)	1	42.5	0.0	0.0	0.0	0.0	0.0	24.0	11.0	0.0
	2	0.0	0.0	0.0	0.0	86.5	0.0	80.0	72.5	97.0
	3	0.0	0.0	0.0	0.0	100.0	36.0	89.0	78.5	100.0
STResnet-2	1	42.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	2	0.0	0.0	0.0	0.0	100.0	0.0	72.0	84.0	100.0
	3	0.0	0.0	0.0	0.0	100.0	100.0	100.0	100.0	100.0

$\epsilon \in \{0.01, 0.03, 0.05, 0.07, 0.1\}$. Overall, we note that deep CFP models, like other deep learning models, are significantly vulnerable to adversarial attacks illustrated by considerably smaller values of $\mathcal{L}^*(\mathcal{D}^*_{test})$ for $\epsilon > 0$ compared to those for $\epsilon = 0$.

Increasing ϵ makes the attack stronger (decrease in the value of $\mathcal{L}^*(\mathcal{D}^*_{test})$), which is consistent with our intuitions. PGD attack appears to be the strongest, while *CVP* attack seems the weakest. However, as we see later, the perturbations generated by the three standard attacks are 100% detectable by *CaV-detect* mechanism, while the perturbations generated by the *CVP* attack remain undetected.

As compared to MLP-5 and TGCN-(5,5), STResnet-2 exhibits smaller values of $\mathcal{L}^*(\mathcal{D}^*_{test})$, suggesting that



(a) Comparing the adversarial loss, $\mathcal{L}^*(\mathcal{D}^*_{test})$, over the perturbed dataset, $\mathcal{D}^*_{test}$, by FGSM-(ϵ, 1) attack for different model architectures.

(b) Comparing the adversarial loss, $\mathcal{L}^*(\mathcal{D}^*_{test})$, over the perturbed dataset, $\mathcal{D}^*_{test}$, by iFGSM-(ϵ, 500) attack for different model architectures.

(c) Comparing the adversarial loss, $\mathcal{L}^*(\mathcal{D}^*_{test})$, over the perturbed dataset, $\mathcal{D}^*_{test}$, by PGD-(ϵ, 500) attack for different model architectures.

(d) Comparing the adversarial loss, $\mathcal{L}^*(\mathcal{D}^*_{test})$, over the perturbed dataset, $\mathcal{D}^*_{test}$, by *CVP*-(ϵ, 500) attack for different model architectures.

Fig. 11. Comparing **the adversarial loss**, $\mathcal{L}^*(\mathcal{D}^*_{test})$, over $\mathcal{D}^*_{test}$, of different attack algorithms adapted with consistency and validity. (a)-(c), with $\mathcal{L}^*(\mathcal{D}^*_{test})$ of *CVP* attack (d) for different model architectures as ϵ is increased assuming **a D-WB-adaptive attacker**. (Settings: Dataset is TaxiBJ-16; l_h is 5). *Of the three architectures assumed in the paper, the TGCN-(5,5) model shows the greatest adaptive adversarial robustness against different attacks followed by the STResnet-2 model.*

STResnet-2 is relatively more vulnerable to adversarial perturbations. This is surprising considering the relatively superior performance of STResnet-2 on the unperturbed dataset. These observations hint at the possibility of an accuracy-robustness tradeoff in the CFP models, as has been commonly observed in other DL models [50], [65].

Unlike other architectures, for MLP-5, $\mathcal{L}^*(\mathcal{D}^*_{test})$ of FGSM slightly increases as ϵ is increased. This is because FGSM is a single-shot attack, and the gradients computed by the attacker only locally estimate the loss surface. Larger perturbations render these local gradients imprecise, thus, degrading the efficacy of the attack. On the other hand, an i-FGSM attacker iteratively computes these gradients after small perturbation steps, which significantly increases the strength of the attack.

Detecting D-WB-Blind Adversarial Perturbations: In this experiment, we evaluate the efficacy of *CaV-detect* to detect adversarial attacks by a D-WB-blind attacker as illustrated in Fig. 7. Overall, with the False Rejection Rate (FRR) set to $\leq$0.5%, our *CaV-detect* mechanism shows a False Acceptance Rate (FAR) of 0% against standard adversarial attacks and FAR of >99.7% against our proposed *CVP* attack.

In Fig. 10, we only report the FAR of the validity check mechanism, as the FAR of the consistency check mechanism is always 0% against standard adversarial attacks and 100% against *CVP* attack. Consequently, the overall FAR of *CaV-detect* is 0% for standard attacks and equal to the FAR of validity check mechanism for *CVP* attack. This can also been seen in Table I in tabular form, where the FAR of adversarially perturbed inputs is 0% for all the attacks.

As ϵ increases, the adversarial perturbations become increasingly invalid. This is not surprising because $\mathbf{X}_h(t)$ is initially valid, and introducing invalid perturbations of larger magnitude more significantly affects the validity of the perturbed inputs. For MLP-5, we observe that the generated perturbations are relatively more valid as compared to TGCN-(5,5) and STResnet-2. We conjecture that because of its relatively simpler architecture, MLP-5 gradients are mostly linear. By definition in eq-(10), if $\mathbf{X}_h(t)$ is valid, its linear multiple is also valid.

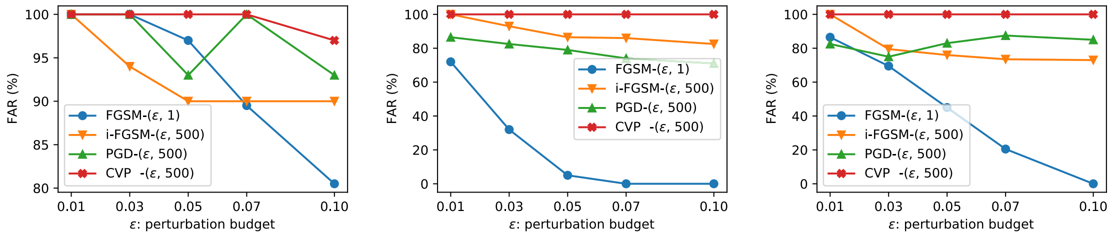
C. D-WB-Adaptive Adversarial Attacks

Fig. 11(a-d) summarizes our results of four different adversarial attacks on the CFP models for $\epsilon \in \{0.01, 0.03, 0.05, 0.07, 0.1\}$. Table II gives a summary of Fig. 11 in tabular form. As observed previously, increasing

TABLE II

Comparison of the Adversarial Loss $\mathcal{L}^*(\mathcal{D}^*_{test})$ (↓= Better Attack) Achieved by Different Adaptive Attacks (Adapted With Our Consistency and Validity, See eq-(25)) With the CVPR Attack for Different Model Depths and Architectures in the Tabular Form. Best Results Are Underlined. CVP *Attack Notably and Consistently Outperforms the Other attacks*

Attack Algorithm	**MLP-l**			**TGCN-(m, d_A)**				**STResnet-l**		
	3	**5**	**10**	**(1, 5)**	**(3, 5)**	**(5, 5)**	**(10, 5)**	**1**	**2**	**3**
FGSM-(0.05, 1)	96.31	142.8	152.55	162.45	160.07	163.35	159.87	156.36	152.82	150.93
i-FGSM-(0.05, 500)	97.88	139.8	153.12	161.73	160.39	163.87	160.45	156.99	153.93	152.16
PGD-(0.05, 500)	91.17	132.15	146.12	160.8	156.13	160.97	156.1	146.57	134.03	126.31
CVP-(0.05, 500)	95.15	132.15	144.57	149.71	148.24	150.53	149.76	144.77	130.95	120.14



(a) MLP-5 (b) TGCN-(5,5) (c) STResnet-2

Fig. 12. False acceptance rate (FAR) of *CaV-detect* mechanism against the perturbed inputs, $\mathcal{D}^*_{test}$, generated by **a D-WB-adaptive attacker**. (Settings: Dataset is TaxiBJ-16. *h* is 5). *The adversarial perturbations become increasingly invalid as ϵ increases. FAR of the consistency check mechanism is always 100%, so we only report FAR of the validity-check mechanism.*

ϵ increases the strength of the attack. Contrary to our previous observation (where STResnet-2 was least robust), against adaptive attacks, the robustness of STResnet models is on par with or better than MLP. Of the three architectures, TGCN models show the greatest adaptive adversarial robustness. Also, our proposed *CVP* attack performs significantly better than other adaptive attacks.

MLP-10 is more robust as compared to MLP-3 and MLP-5, owing to the greater model complexity [50]. All the standard attacks considered in this paper give a comparable adversarial performance on MLP, which appears counter-intuitive (as i-FGSM and PGD are generally considered stronger than FGSM), but can be attributed to the simplicity of MLP architecture, which makes it equally vulnerable to relatively simpler attacks.

For the TGCN architecture, we do not observe a definitive effect of increasing a model's complexity on its adversarial robustness. However, STResnet-1 shows a considerably greater adversarial robustness, respectively followed by STResnet-2 and STResnet-3, which can be attributed to the increased adversarial vulnerability of latent DNN layers [66].

Detecting D-WB-Adaptive Adversarial Perturbations: In this experiment, we evaluate how effectively do the adversarial inputs perturbed adaptively are detected by *CaV-detect*. Fig. 12 only reports the FAR of the validity check mechanism under adaptive attack settings, because the FAR of the consistency check mechanism is always 100% against all adaptive adversarial attacks due to the universal adversarial perturbations. Consequently, the overall FAR of *CaV-detect* is equal to the FAR of the validity check mechanism. To summarize our results, with the False Rejection Rate (FRR) set to ≤0.5%, adaptive attacks have considerably higher FAR (≈80%-100%) against *CaV-detect* modified by our novel objective (eq-(25)) as compared to FAR of D-WB blind/aware attacks. However, despite their high FAR, adaptive attacks also achieve a much higher $\mathcal{L}^*(\mathcal{D}^*)$ as compared to D-WB-blind attacks and *CVP* attack.

As previously observed, increasing ϵ considerably decreases FAR. Contrary to the *CaV-detect*-blind attacks, D-WB-adaptive attacks can evade *CaV-detect* with around 80% FAR for standard adaptive attacks. We specifically attribute this to the newly proposed adaptive modifications to the standard attacks—*universalizing* the adversarial perturbations and adaptive Lagrange optimization. However, despite its D-WB-adaptive algorithm, FGSM fails to perform well against *CaV-detect*, particularly notable for TGCN-(5,5) and STResnet-2 where FAR drops to 0% when $\epsilon = 0.1$ (also see Table I, which can again be attributed to the imprecise local gradients.

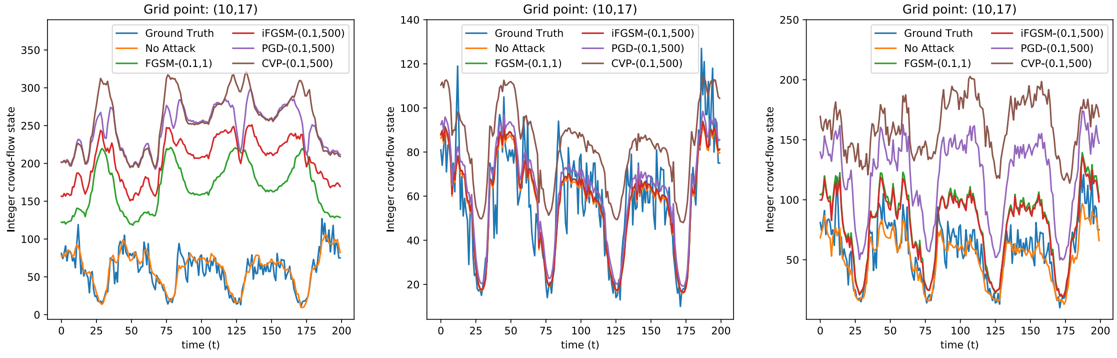
We also compare FAR(n) of adaptive adversarial attacks for a range of values of n in Table I for $\epsilon = 0.1$. As observed previously for the original inputs, increasing n makes *CaV-detect* more tolerant towards adversarial perturbations resulting in the adaptive attacks achieving 100% FAR against MLP-5 and STResnet-2 for $n \geq 3$.

VI. Discussions

A. Visualizing CFPs

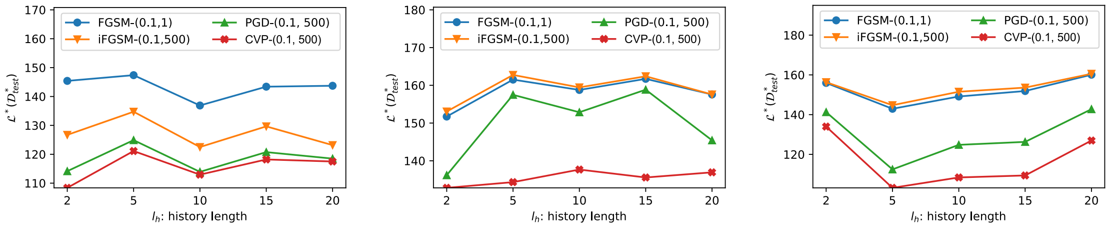
Fig. 13 compares the predicted inflow states of different CFP models with the ground truths recorded in the future for both the original and the perturbed inputs generated by different adaptive attacks, where the goal of the attacks is to increase the predicted inflow state as much as possible while keeping $\boldsymbol{\delta} \in \mathcal{B}_\infty(\epsilon)$. The qualitative analysis shows that the *CVP* attack outperforms other attacks, as the predicted inflow state for *CVP* attacked inputs typically exhibits the highest value, irrespective of the model architecture. Furthermore, TGCN-(5,5) is more robust to the adaptive attacks as compared to other models. We also note that the predicted inflow state is more affected by the *CVP* perturbations when the originally predicted inflow state is relatively small.

Interestingly, the predicted inflow states for the adversarial inputs are, in general, highly correlated—either positively (Fig. 13(b,c) or negatively (FGSM-(0.1,1) and *CVP*-(0,1,500)



(a) MLP-5 (b) TGCN-(5,5) (c) STResnet-2

Fig. 13. Visualizing the predicted inflow states of the CFP models of different architectures with the actual inflow states (recorded in the future). “No Attack” denotes the predicted inflow states for the original/unperturbed inputs assuming **a D-WB-adaptive attacker**. (Settings: Dataset is TaxiBJ-16; h is 5; ϵ is 0.1). CVP *attack outperforms the other attacks. TGCN-(5,5) is more robust to consistent and valid adversarial attacks than the other two models.*



(a) MLP-5 (b) TGCN-(5,5) (c) STResnet-2

Fig. 14. A comparison of **the adversarial loss**, $\mathcal{L}^*(\mathcal{D}^*_{test})$, over **the perturbed test inputs** generated by different attacks for different models (of varying architectures) trained for different history length, h. (Settings: Dataset is TaxiBJ-16; ϵ is 0.1). *Typically, when the input history length is increased, the* $\mathcal{L}^*(\mathcal{D}^*_{test})$ *slightly increases indicating that the models trained on a larger history length are slightly more robust to the adversarial perturbations.*

on MLP-5 in Fig. 13(a)—with the originally predicted inflow states, for all the models considered in this experiment. Based on these observations, we conjecture that the CFP models have limited expressiveness—the models are incapable to produce certain outputs irrespective of the inputs.

B. Effect of History Length, h, on Adversarial Loss

We analyze the effect of changing h on the adversarial loss of different attacks. We train MLP-5, TGCN-(5,5) and STResnet-2 for $h \in \{2, 5, 10, 15, 20\}$ and then attack the models. Note that for each h, we train a new model as the inputs of the models trained for different h values are incompatible with each other. For this experiment, we set ϵ=0.1 and N=500 for all the attacks. Fig. 14(a-c) reports $\mathcal{L}^*(\mathcal{D}^*_{test})$ values of attacks on the aforementioned three models respectively.

We observe no strict relationship between the adversarial robustness and h. However, increasing h typically makes the model slightly more robust, which appears counter-intuitive as a greater history length allows an attacker to add more perturbations to the input. However, recalling from Section V-A, the increasing robustness night be due to the decreased performance of the models when h is increased hinting the accuracy-robustness tradeoff [50], [65].

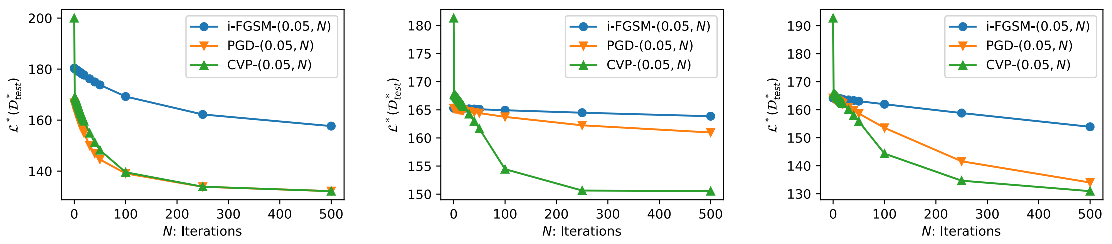
C. Speed of Adversarial Attacks

Fig. 15(a-c) compares the speed of different attacks to optimize the adversarial loss along the number of iterations for MLP-5, TGCN-(5,5) and STResnet-2. As the attacks progress, the generated perturbations become better indicated by the decreasing adversarial loss. *CVP* attack consistently outperforms the other two attacks with a significant margin, specifically notable for TGCN-(5,5) and STResnet-2.

Fig. 15(a),(b) also hint that the adaptive PGD attack performs slightly better than *CVP* attack for lesser iterations (more notable for MLP-5). This can be attributed to the additional constraints imposed by the consistent and valid perturbation generation mechanism of *CVP* attack.

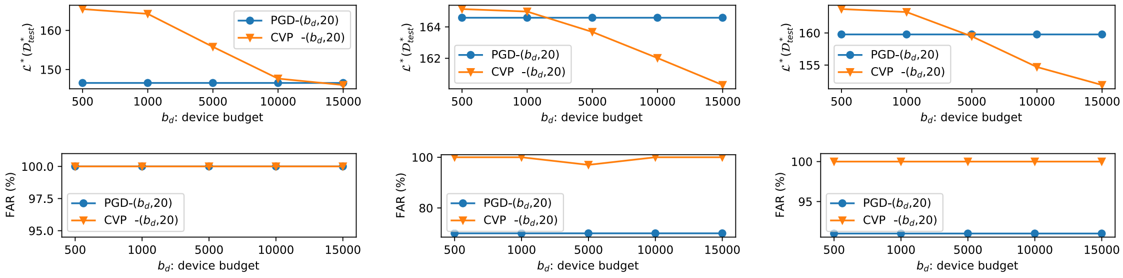
D. On Physical-Realizability of Adversarial Attacks

In order to study the potential impact of the proposed adversarial attack in the real world, we assume a physical threat model with a physical attacker who cannot control the input to the CFP model, but may influence it by controlling physical adversarial devices (e.g., the mobile phones or the GPS modules communicating with the pipeline) up to a certain device budge b_d as perturbations in the real world. The aim of the physical attacker is to change the model's prediction about the future CFS by physically controlling the available b_d adversarial devices in the most optimal way. P-WB-adaptive attack in Fig. 7 illustrates such a threat model. In addition, we further limit our attacker by assuming a limited query setting [7] that limits our attacker to be only able to query the model 20 times at maximum. We further assume that our attacker has a limited device budget, b_d, defining the number of devices, that we refer to as the adversarial devices, which our attacker can physically control. The goal of our attacker is to fool the crowd-prediction model by physically moving the adversarial devices (to simulate adversarial perturbations). We vary the $b_d \in \{500, 1000, 5000, 10000, 15000\}$, and report the adversarial loss of two attacks—*adaptive*



(a) MLP-5 (b) TGCN-(5,5) (c) STResnet-2

Fig. 15. Comparing the decline of adversarial loss, $\mathcal{L}^*(\mathcal{D}^*_{test})$, over the perturbed dataset, $\mathcal{D}^*_{test}$, by different attack algorithms for different model architectures as the attack progresses assuming **a D-WB-adaptive attacker**. (Settings: Dataset is TaxiBJ-16; h is 5; ϵ is 0.05). CVP *attack consistently outperforms other attacks in terms of the adversarial loss and speed, given a perturbation budget.*



(a) MLP-5 (b) TGCN-(5,5) (c) STResnet-2

Fig. 16. Comparing the physical plausibility of the PGD attack and the *CVP* attack for different model architectures at different device budgets in terms of the adversarial loss, $\mathcal{L}^*(\mathcal{D}^*_{test})$, and FAR of *CaV-detect* assuming **a D-WB-adaptive attacker**. (Settings: Dataset is TaxiBJ-16; h is 5; N is 20; b_d is the maximum number of devices physically controllable by the attacker).

PGD-(b_d, 20) (with consistency and validity priors formalized in eq-(25)) and *CVP*-(b_d, 20)—in Fig. 16 for different values of b_d.

Interestingly, we note that for smaller b_d the *adaptive* PGD attack outperforms the *CVP* attack in terms of $\mathcal{L}^*(\mathcal{D}^*_{test})$ for all the three architectures considered in this experiment. We attribute this to two reasons. Firstly, the adversarial perturbations generated by *adaptive* PGD attack are relatively more invalid as compared to those generated by the *CVP* attack, observable in the last row of Fig. 16, which reports FAR of the *CaV-detect* mechanism. Secondly, the outflow perturbation generating mechanism proposed in eq-(18) implicitly imposes additional constraints on δ_{in} and δ_{out}, which makes it difficult to find optimal perturbations in fewer iterations. This observation also aligns with that observed in Fig. 15 in the long run (for larger b_d and N). However, given enough iterations, $\mathcal{L}^*(\mathcal{D}^*_{test})$ achieved by the *CVP* attack is notably smaller than the PGD attack. Therefore, we believe that *CVP* attack is the strongest crowd-flow prediction attack yet, and is much more practical than the PGD attack.

Compared to $\mathcal{L}^*(\mathcal{D}^*_{test})$ values in Fig. 11, $\mathcal{L}^*(\mathcal{D}^*_{test})$ in Fig. 16 are notably larger, which can simply be attributed to the limited query budget and device budget of the attacker. This shows that although the CFP models are vulnerable to consistent and valid adversarial perturbations under physical setting, realizing the targeted outputs is considerably more challenging than the digital attack setting—$b_d = 15000$ is still a very large number. A low power GPS module typically costs ∼1$. Therefore, each adversarial device not only incurs an additional cost of ∼1$, but the attacker might have to use an additional vehicle if the computed path in not similar to the previous ones. Given b_d, an attacker may use our openly available code to compute an optimal perturbation that is consistent, valid and physically-realizable.

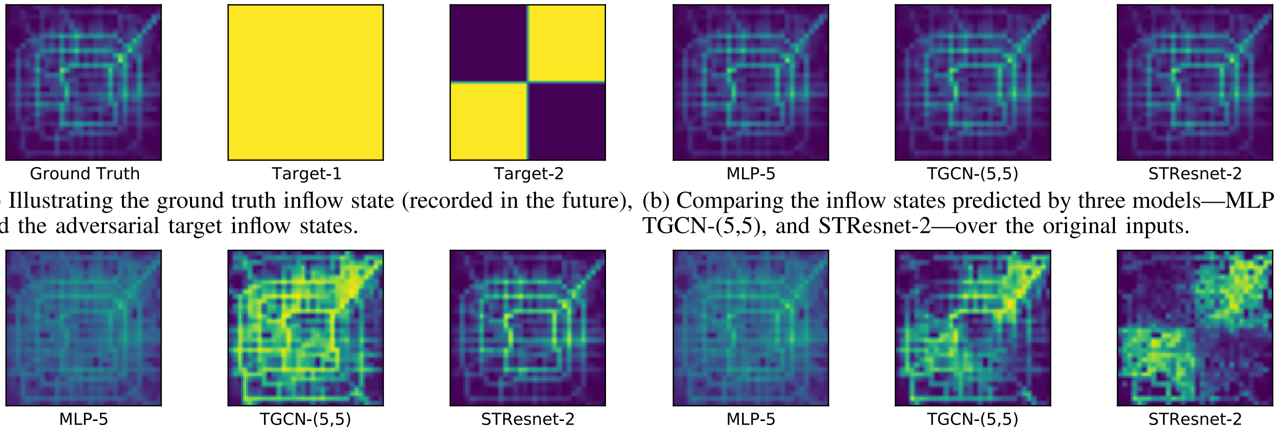
E. Limited Expressiveness of CFP Models

In this experiment, we show that MLP-5, TGCN-(5,5), and STResnet-2 exhibit limited expressiveness. We define expressiveness as the ability of a model to produce a certain output given an appropriate input. Fig 17(b) shows that all the three models perfrom well on the original inputs. To show the limited expressiveness of the models, we assume a strong PGD-(1,500) adversary with $\epsilon = 1$ so that the adversary can make any change to the input with *CaV-detect*-blind threat model—the adversary does not have to care about the *CaV-detect*. Additionally, we assign two adversarial target states—Target-1 and Target-2 in Fig. 17(a)—for the adversary to produce at the models' outputs. Fig. 17(c,d) report the output predictions of models on adversarial inputs generated by PGD-(1,500) attack for Target-1 and Target-2 respectively.

Ideally, assuming a complete control over the inputs, the adversary should be able to manipulate the model into producing any desired output. However, results in Fig. 17(c,d) show that this is not the case with the CFP models. For example, the outputs of MLP-5 are significantly different from the targets, which concludes that MLP-5 is incapable of producing the target outputs. We conjecture that MLP-5 has very limited expressiveness. Although TGCN-(5,5) and STResnet-2 get significantly closer to the target outputs as compared to MLP-5, they still lack sufficient expressiveness to exactly produce the target output. We attribute this to the mostly clustered and highly similar outputs in the TaxiBJ dataset. Overall, we observe that STResnet-2 is the most expressive, which also explains why STResnet models are adversarially less robust compared to TGCN models (as observed in Fig. 11).

F. Challenges and Future Directions

Real world scenarios are challenging and diverse. White-box (WB) attacks help us understand the behavior and stability



(a) Illustrating the ground truth inflow state (recorded in the future), and the adversarial target inflow states.

(b) Comparing the inflow states predicted by three models—MLP-5, TGCN-(5,5), and STResnet-2—over the original inputs.

(c) Comparing the inflow states predicted by three models over the PGD-blind attacked inputs optimized for Target-1.

(d) Comparing the inflow states predicted by three models over the PGD-blind attacked inputs optimized for Target-2.

Fig. 17. Illustrating limited *expressiveness* (the ability of a model to produce the desired output given an infinite control over the input) of CFP models of different architectures. (Settings: Dataset is TaxiBJ-16; h is 5; Attack is PGD-(1,500); ϵ is 1—indicating infinite control over the inputs). *STResnet-2 is the most expressive of the three models considered in this experiment, while MLP-5 is the least expressive*.

of the subject model in the worst-case scenarios, and give us further insights into the potential fail cases where the model behaves unexpectedly. Previous works have shown that white-box adversarial attacks can be a real threat, even in the black-box scenario, due to additional challenges like insider threats, network breach and viruses. Several open research questions remain unanswered, such as, how a hybrid threat setting targeting multiple such vulnerabilities make adversarial concerns graver, and how to counter the adversarial threat in such a hybrid setting. In the following, we state some of these challenges for future researchers.

Model Stealing Attacks: Previous works have shown that given query access to the model, it is often possible for an expert attacker to perform the model stealing attack by either repeatedly querying the model using adversarial data or monitoring its input and the corresponding output, given a network security breach (which is not uncommon in today's systems). Gradients from the stolen model can then be used to estimate the original gradients and optimize the perturbation.

Insider Threat: Insider threat may let an adversary steal a copy of the deployed model, compute gradients, and optimize the perturbations in order to maximize the adversarial reward.

Black-box Adversarial Attacks: Zeroth-order optimization, decision-based attacks and their advanced alterations can be used to simulate almost all of the white-box attacks (including the *CVP* attack) in the black-box setting.

Limitation of Our Study: One of the major limitations of our study is that, although *CVP* attack is physically realizable, our experiments do not include its real-world implementation due to the lack of sufficient resources for a real-world study. We only simulate a physical attacker in our python framework.

VII. Conclusion

In this paper we studied the adversarial vulnerabilities of the CFP models of three different architectures—Multi-Layer Perceptron, Temporal Graph Convolution Neural Network and Spatio-Temporal ResNet. We extensively analyze the effects of changing the model complexity and crowd-flow data history length on the performance and the adversarial robustness of the resulting models, and find that the CFP models, like other deep learning models, are significantly vulnerable to adversarial attacks. Secondly, we identified and normalized two novel properties—consistency and validity—of the crowd-flow inputs that can be used to detect adversarially perturbed inputs. We therefore propose *CaV-detect* that can detect adversarial inputs with FAR of 0% by analyzing their consistency and validity—a model input is considered unperturbed if it is both consistent and valid. We then adaptively modify the standard adversarial attacks to evade *CaV-detect* with an FAR of ∼80-100%. Finally, by encoding the consistency and validity priors in the adversarial perturbation generating mechanism, we propose *CVP* attack, a consistent, valid and physically-realizable adversarial attack that outperforms the adaptive standard attacks in terms of both the target adversarial loss and the FAR of *CaV-detect*. Lastly, insightfully discuss the adversarial attacks on CFP models and show that CFP models exhibit limited expressiveness and can be physically realized by simulating universal perturbations.

Acknowledgment

The statements made herein are solely the responsibility of the authors. The authors would also like to thank the Qatar National Library for providing Open Access funding.

References

[1] S. Saadatnejad, M. Bahari, P. Khorsandi, M. Saneian, S.-M. Moosavi-Dezfooli, and A. Alahi, "Are socially-aware trajectory prediction models really socially-aware?" *Transp. Res. C, Emerg. Technol.*, vol. 141, Aug. 2022, Art. no. 103705.

[2] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1655–1661.

[3] W. Jiang, "TaxiBJ21: An open crowd flow dataset based on Beijing taxi GPS trajectories," *Internet Technol. Lett.*, vol. 5, no. 2, p. e297, Mar. 2022.

[4] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. 6th Int. Conf. Learn. Represent.*, Vancouver, BC, Canada, 2018, pp. 1–16. [Online]. Available: https://openreview.net/forum?id=SJiHXGWAZ

[5] S. Liu et al., "Harnessing perceptual adversarial patches for crowd counting," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2022, pp. 2055–2069.

[6] F. Liu, H. Liu, and W. Jiang, "Practical adversarial attacks on spatiotemporal traffic forecasting models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 1–13.

[7] F. Khalid, H. Ali, M. A. Hanif, S. Rehman, R. Ahmed, and M. Shafique, "FaDec: A fast decision-based attack for adversarial machine learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8.

[8] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.

[9] C. Szegedy et al., "Intriguing properties of neural networks," in *Proc. 2nd Int. Conf. Learn. Represent.*, Banff, AB, Canada, Y. Bengio and Y. LeCun, Eds., 2014, pp. 1–10.

[10] M. Liu, Z. Zhang, Y. Chen, J. Ge, and N. Zhao, "Adversarial attack and defense on deep learning for air transportation communication jamming," *IEEE Trans. Intell. Transp. Syst.*, early access, 2023, doi: 10.1109/TITS.2023.3262347.

[11] J. Tian, B. Wang, R. Guo, Z. Wang, K. Cao, and X. Wang, "Adversarial attacks and defenses for deep-learning-based unmanned aerial vehicles," *IEEE Internet Things J.*, vol. 9, no. 22, pp. 22399–22409, Nov. 2022.

[12] L. Zhao et al., "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020.

[13] L. Mourad, H. Qi, Y. Shen, and B. Yin, "ASTIR: Spatio-temporal data mining for crowd flow prediction," *IEEE Access*, vol. 7, pp. 175159–175165, 2019.

[14] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 274–283.

[15] F. Tramer, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1633–1645.

[16] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1765–1773.

[17] M. Chen, X. Yu, and Y. Liu, "PCNN: Deep convolutional networks for short-term traffic congestion prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 11, pp. 3550–3559, Nov. 2018.

[18] E. Onieva, V. Milanés, J. Villagrá, J. Pérez, and J. Godoy, "Genetic optimization of a vehicle fuzzy decision system for intersections," *Expert Syst. Appl.*, vol. 39, no. 18, pp. 13148–13157, Dec. 2012.

[19] X. Zhang, E. Onieva, A. Perallos, E. Osaba, and V. C. S. Lee, "Hierarchical fuzzy rule-based system optimized with genetic algorithms for short term traffic congestion prediction," *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 127–142, Jun. 2014.

[20] E. Onieva, P. Lopez-Garcia, A. D. Masegosa, E. Osaba, and A. Perallos, "A comparative study on the performance of evolutionary fuzzy and crisp rule based classification methods in congestion prediction," *Transp. Res. Proc.*, vol. 14, pp. 4458–4467, Dec. 2016.

[21] P. Lopez-Garcia, E. Onieva, E. Osaba, A. D. Masegosa, and A. Perallos, "A hybrid method for short-term traffic congestion forecasting using genetic algorithms and cross entropy," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 557–569, Feb. 2016.

[22] Y. Zheng, Y. Li, C.-M. Own, Z. Meng, and M. Gao, "Real-time predication and navigation on traffic congestion model with equilibrium Markov chain," *Int. J. Distrib. Sensor Netw.*, vol. 14, no. 4, Apr. 2018, Art. no. 155014771876978.

[23] J. F. Zaki, A. Ali-Eldin, S. E. Hussein, S. F. Saraya, and F. F. Areed, "Traffic congestion prediction based on hidden Markov models and contrast measure," *Ain Shams Eng. J.*, vol. 11, no. 3, pp. 535–551, Sep. 2020.

[24] S. Sun, J. Chen, and J. Sun, "Traffic congestion prediction based on GPS trajectory data," *Int. J. Distrib. Sensor Netw.*, vol. 15, no. 5, May 2019, Art. no. 155014771984744.

[25] Y. Qi and S. Ishak, "A hidden Markov model for short term prediction of traffic conditions on freeways," *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 95–111, Jun. 2014.

[26] S. Yang, "On feature selection for traffic congestion prediction," *Transp. Res. C, Emerg. Technol.*, vol. 26, pp. 160–169, Jan. 2013.

[27] L. Zhu, R. Krishnan, F. Guo, J. W. Polak, and A. Sivakumar, "Early identification of recurrent congestion in heterogeneous urban traffic," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 4392–4397.

[28] S. Sun, C. Zhang, and G. Yu, "A Bayesian network approach to traffic flow forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 124–132, Mar. 2006.

[29] G. Asencio-Cortés, E. Florido, A. Troncoso, and F. Martínez-Álvarez, "A novel methodology to predict urban traffic congestion with ensemble learning," *Soft Comput.*, vol. 20, no. 11, pp. 4205–4216, Nov. 2016.

[30] J. Kim and G. Wang, "Diagnosis and prediction of traffic congestion on urban road networks using Bayesian networks," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2595, no. 1, pp. 108–118, Jan. 2016.

[31] X. Yang, S. Luo, K. Gao, T. Qiao, and X. Chen, "Application of data science technologies in intelligent prediction of traffic congestion," *J. Adv. Transp.*, vol. 2019, pp. 1–14, Apr. 2019.

[32] K. M. Nadeem and T. P. Fowdur, "Performance analysis of a real-time adaptive prediction algorithm for traffic congestion," *J. Inf. Commun. Technol.*, vol. 17, no. 3, pp. 493–511, 2018.

[33] J. Lee, B. Hong, K. Lee, and Y.-J. Jang, "A prediction model of traffic congestion using weather data," in *Proc. IEEE Int. Conf. Data Sci. Data Intensive Syst.*, Dec. 2015, pp. 81–88.

[34] P. Zhang and Z. Qian, "User-centric interdependent urban systems: Using time-of-day electricity usage data to predict morning roadway congestion," *Transp. Res. C, Emerg. Technol.*, vol. 92, pp. 392–411, Jul. 2018.

[35] S. Jain, S. S. Jain, and G. Jain, "Traffic congestion modelling based on origin and destination," *Proc. Eng.*, vol. 187, pp. 442–450, Jan. 2017.

[36] F.-H. Tseng, J.-H. Hsueh, C.-W. Tseng, Y.-T. Yang, H.-C. Chao, and L.-D. Chou, "Congestion prediction with big data for real-time highway traffic," *IEEE Access*, vol. 6, pp. 57311–57323, 2018.

[37] X. Wang, K. An, L. Tang, and X. Chen, "Short term prediction of freeway exiting volume based on SVM and KNN," *Int. J. Transp. Sci. Technol.*, vol. 4, no. 3, pp. 337–352, Sep. 2015.

[38] Y. Liu and H. Wu, "Prediction of road traffic congestion based on random forest," in *Proc. 10th Int. Symp. Comput. Intell. Design (ISCID)*, vol. 2, Dec. 2017, pp. 361–364.

[39] Z. Chen, Y. Jiang, and D. Sun, "Discrimination and prediction of traffic congestion states of urban road network based on spatio-temporal correlation," *IEEE Access*, vol. 8, pp. 3330–3342, 2020.

[40] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, Apr. 2017.

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

[42] Y. Xing, X. Ban, X. Liu, and Q. Shen, "Large-scale traffic congestion prediction based on the symmetric extreme learning machine cluster fast learning method," *Symmetry*, vol. 11, no. 6, p. 730, May 2019.

[43] L. Lin, J. C. Handley, and A. W. Sadek, "Interval prediction of short-term traffic volume based on extreme learning machine and particle swarm optimization," Transp. Res. Board, Washington, DC, USA, Tech. Rep. 17-04796, 2017.

[44] M. A. Butt, A. Qayyum, H. Ali, A. Al-Fuqaha, and J. Qadir, "Towards secure private and trustworthy human-centric embedded machine learning: An emotion-aware facial recognition case study," *Comput. Secur.*, vol. 125, Feb. 2023, Art. no. 103058.

[45] X. Yang, W. Liu, S. Zhang, W. Liu, and D. Tao, "Targeted attention attack on deep learning models in road sign recognition," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4980–4990, Mar. 2021.

[46] S. Latif, R. Rana, and J. Qadir, "Adversarial machine learning and speech emotion recognition: Utilizing generative adversarial networks for robustness," 2018, *arXiv:1811.11402*.

[47] M. Usama, A. Qayyum, J. Qadir, and A. Al-Fuqaha, "Black-box adversarial machine learning attack on network traffic classification," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2019, pp. 84–89.

[48] M. Usama, J. Qadir, and A. Al-Fuqaha, "Adversarial attacks on cognitive self-organizing networks: The challenge and the way forward," in *Proc. IEEE 43rd Conf. Local Comput. Netw. Workshops (LCN Workshops)*, Oct. 2018, pp. 90–97.

[49] M. Usama, M. Asim, S. Latif, J. Qadir, and Ala-Al-Fuqaha, "Generative adversarial networks for launching and thwarting adversarial attacks on network intrusion detection systems," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2019, pp. 78–83.

[50] H. Ali et al., "All your fake detector are belong to us: Evaluating adversarial robustness of fake-news detectors under black-box settings," *IEEE Access*, vol. 9, pp. 81678–81692, 2021.

[51] J. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin, and Y. Qi, "TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP," in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstrations*, Q. Liu and D. Schlangen, Eds. Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 119–126, doi: 10.18653/v1/2020.emnlp-demos.16.

[52] H. Ali, M. S. Khan, A. Al-Fuqaha, and J. Qadir, "Tamp-X: Attacking explainable natural language classifiers through tampered activations," *Comput. Secur.*, vol. 120, Sep. 2022, Art. no. 102791.

[53] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, Nov. 2017, pp. 15–26.

[54] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," in *Proc. 6th Int. Conf. Learn. Represent.*, Vancouver, BC, Canada, 2018, pp. 1–12. [Online]. Available: https://openreview.net/forum?id=SyZI0GWCZ

[55] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. 3rd Int. Conf. Learn. Represent.*, San Diego, CA, USA, Y. Bengio and Y. LeCun, Eds., 2015, pp. 1–11.

[56] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. 6th Int. Conf. Learn. Represent.*, Vancouver, BC, Canada, 2018, pp. 1–23. [Online]. Available: https://openreview.net/forum?id=rJzIBfZAb

[57] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 39–57.

[58] F. Khalid et al., "QuSecNets: Quantization-based defense mechanism for securing deep neural network against adversarial attacks," in *Proc. IEEE 25th Int. Symp. On-Line Test. Robust Syst. Design (IOLTS)*, Jul. 2019, pp. 182–187.

[59] H. Ali, F. Khalid, H. A. Tariq, M. A. Hanif, R. Ahmed, and S. Rehman, "SSCNets: Robustifying DNNs using secure selective convolutional filters," *IEEE Design & Test*, vol. 37, no. 2, pp. 58–65, Apr. 2020.

[60] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2016, pp. 582–597.

[61] G. S. Dhillon et al., "Stochastic activation pruning for robust adversarial defense," 2018, *arXiv:1803.01442*.

[62] D. Meng and H. Chen, "MagNet: A two-pronged defense against adversarial examples," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 135–147.

[63] N. Carlini and D. Wagner, "Defensive distillation is not robust to adversarial examples," 2016, *arXiv:1607.04311*.

[64] N. Carlini and D. Wagner, "MagNet and 'efficient defenses against adversarial attacks' are not robust to adversarial examples," 2017, *arXiv:1711.08478*.

[65] B. Wu, J. Chen, D. Cai, X. He, and Q. Gu, "Do wider neural networks really help adversarial robustness?" in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 7054–7067.

[66] N. Kumari, M. Singh, A. Sinha, H. Machiraju, B. Krishnamurthy, and V. N. Balasubramanian, "Harnessing the vulnerability of latent layers in adversarially trained models," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Macao, China, S. Kraus Ed., Aug. 2019, pp. 2779–2785, doi: 10.24963/ijcai.2019/385.

Hassan Ali received the M.S. degree (Hons.) from the School of Electrical Engineering and Computer Sciences, National University of Science and Technology (NUST), Pakistan. He is currently with the Department of Electrical Engineering, Information Technology University (ITU), Lahore, Pakistan. His research interests include embedded systems, machine learning, artificial intelligence, and security.

Muhammad Atif Butt received the master's degree in computer science from the School of Electrical Engineering and Computer Science (SEECS), National University of Science and Technology (NUST), Islamabad, Pakistan. He is currently pursuing the Ph.D. degree with the University of Barcelona and the Computer Vision Center (CVC). He was a Research Assistant with the Department of Electrical Engineering, Information Technology University (ITU), Lahore, Pakistan. Before joining ITU in August 2021, he was a Research Associate with the Control, Automotive and Robotics Laboratory, National Centre of Robotics and Automation (NCRA), Pakistan. His research interests include computer systems, intelligent information systems, applied artificial intelligence (AI), and adversarial machine learning (AML).

Fethi Filali (Senior Member, IEEE) received the Ph.D. degree in computer science and the Habilitation degree from the University of Nice Sophia Antipolis, France, in 2002 and 2008, respectively. He is currently the Director of Technology and Research with the Qatar Mobility Innovations Center (QMIC). He is also leading the development of innovating and applied technologies in the areas of artificial intelligence, geospatial data analytics, embedded sensing, scalable and distributed algorithms, and communication systems and protocols. Prior to joining QMIC, in 2010, he was with the Mobile Communications Department, EURECOM, France, as an Assistant Professor and an Associate Professor for eight years. His research grants include 17 competitive awards from several funding agencies (European Commission, The French National Research Agency, and Qatar National Research Fund). He was the Ph.D. Director of ten Ph.D. students in the areas of intelligent transportation, wireless sensor and mesh networks, vehicular communications, big data analytics, the Internet of Things, and mobility management. He has coauthored more than 130 research papers in international peer-reviewed conferences and journals and (co-)filed more than ten patent applications.

Ala Al-Fuqaha (Senior Member, IEEE) received the Ph.D. degree in computer engineering and networking from the University of Missouri-Kansas City, Kansas City, MO, USA, in 2004. His research interests include the use of machine learning, in general, and deep learning, in particular, in support of the data- and self-driven management of large-scale deployments of the Internet of Things and smart city infrastructure and services, wireless vehicular networks (VANETs), cooperation and spectrum access etiquette in cognitive radio networks, and management and planning of software-defined networks (SDNs).

Junaid Qadir (Senior Member, IEEE) is currently a Professor with the Department of Computer Science and Engineering, Qatar University, Doha, Qatar. He is also the former Chairperson of the Department of Electrical Engineering, Information Technology University (ITU), Lahore, Punjab, Pakistan, where he also directs the IHSAN Research Laboratory. He has published more than 150 peer-reviewed articles at various high-quality research venues including journal publications at top international research journals, including *IEEE Communication Magazine*, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (JSAC), IEEE COMMUNICATIONS SURVEYS AND TUTORIALS (CST), and IEEE TRANSACTIONS ON MOBILE COMPUTING (TMC). His primary research interests include computer systems and networking, applied machine learning, ICT for development (ICT4D), and engineering education. He is a Senior Member of ACM. He was awarded the Highest National Teaching Award in Pakistan and the Higher Education Commission's (HEC) Best University Teacher Award from 2012 to 2013. He has been appointed as an ACM Distinguished Speaker for a three-year term starting from 2020.