Adversarial Attacks and Defenses on Cyber–Physical Systems: A Survey

Jiao Li, *Student Member, IEEE*, Yang Liu, *Student Member, IEEE*, Tao Chen, *Student Member, IEEE*, Zhen Xiao, *Student Member, IEEE*, Zhenjiang Li, *Member, IEEE*, and Jianping Wang, *Senior Member, IEEE*

***Abstract*—Cyber-security issues on adversarial attacks are actively studied in the field of computer vision with the camera as the main sensor source to obtain the input image or video data. However, in modern cyber–physical systems (CPSs), many other types of sensors are becoming popularly used, such as surveillance sensors, microphones, and textual interfaces. A series of recent works investigates the adversarial attacks and the potential defenses in these noncamera sensor-based CPSs. Therefore, this article provides a systematic discussion on these existing works and serves as a complimentary summary of the adversarial attacks and defenses for CPSs beyond the field of computer vision. We first introduce a general working flow for adversarial attacks on CPSs. On this basis, a clear taxonomy is provided to organize existing attacks effectively and indicate where the defenses can be potentially performed in CPSs as well. Then, we discuss these existing attacks and defenses with detailed comparison studies. Finally, we point out concrete research opportunities to be further explored along this research direction.**

***Index Terms*—Adversarial attacks and defense, cyber–physical systems (CPSs), cyber-security.**

I. Introduction

THE RAPID advancement in the Internet of Things (IoT) [8] and machine learning [52], two promising techniques have made cyber–physical systems (CPSs) [8] increasingly sophisticated, intelligent, and autonomous. In particular, the IoT uses various useful sensors, which can enable novel ways for CPSs to conduct sensing and collect rich sensory data inputs. Learning algorithms, such as deep learning, will further process the sensory data to obtain the desired information for controlling the system. With such enriched input data sources and capable data processing ability, several useful applications have been developed in practice, such as smart home, industrial monitoring, robotic systems, intelligent transportation, and e-health, which can benefit aspects of people's daily life.

Although CPSs are advancing to be more functional and capable than before, many works have reported that these systems are vulnerable to a series of malicious attacks, among which the *adversarial attack* draws considerable attention due to the recent great success of the deep learning [2]. In particular, the adversarial attack targets at the neural network of deep learning in CPSs. The attacker can add a small "noise," which is known as *perturbation*, to the normal data input. In this way, people or certain deployed data monitors in the system will not detect the added perturbation. However, the composed input (normal input plus perturbation) can cause incorrect output of the neural network, and the output can be precisely determined by the selection of perturbation by the attacker in advance. For example, in a smart home, an audio command still sounds similar to "good afternoon my friends" to us, whereas the speech recognition system may decode it as "switch off all surveillance cameras" due to the adversarial attack. In other words, this attack can arbitrarily manipulate the output of the neural network without people's perception, which is thus a serious security risk in practice.

In the literature, adversarial attacks (and potential countermeasures) have been actively studied and comprehensively surveyed in the field of computer vision [2]. In this article, we find that a systematic study on the adversarial attacks and the potential defenses for CPSs beyond the field of computer vision is still lacking, but it is highly desired. In addition to the camera, current CPSs utilize various useful sensors (e.g., surveillance sensors) to collect the system and environmental monitoring data, microphone to receive audios, and textual interface to accept the text inputs. These noncamera sensor-based CPSs are already widely adopted in practice.

Existing attacks on these noncamera sensor-based CPSs can unveil the vulnerability of the system designs and alarm people the potential security risks when they are using such systems. By being aware of these potential risks, people naturally further want to determine effective defenses against these adversarial attacks. As a result, this article provides a systematic discussion for this specific and crucial cyber-security issue. This article can serve as a complementary summary of adversarial attacks and defenses for the CPSs beyond the field of computer vision. To this end, we make the following contributions.

1) *General Working Flow:* We summarize a general working flow to describe the adversarial attacks in CPSs. On this basis, a clear taxonomy is provided to structure and

Manuscript received November 2, 2019; revised February 2, 2020; accepted February 14, 2020. Date of publication February 21, 2020; date of current version June 12, 2020. This work was supported in part by GRF grant from the Research Grants Council of Hong Kong under Project CityU 11217817, and in part by the Science Technology and Innovation Committee of Shenzhen Municipality Under Project JCYJ20170818095109386. *(Corresponding author: Zhenjiang Li.)*

The authors are with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: jli242-c@my.cityu.edu.hk; yliu562-c@my.cityu.edu.hk; tachen6-c@my.cityu.edu.hk; zxian4-c@my.cityu.edu.hk; zhenjiang.li@cityu.edu.hk; jianwang@cityu.edu.hk).

Digital Object Identifier 10.1109/JIOT.2020.2975654

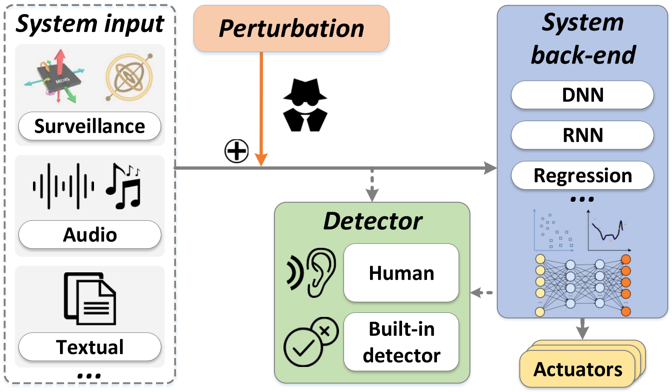

Fig. 1. General working flow of adversarial attacks in CPSs.

organize existing attacks effectively and indicate where the defenses can be potentially performed in CPSs.

2) *Systematic and Comparable Studies:* From the above taxonomy, we classify the existing attacks according to three popular sensor data types, namely, surveillance sensor, audio, and textual data. We also provide a detailed comparison among them from six detailed technical aspects. Moreover, we identify and summarize three feasible directions to defend the adversarial attacks in CPSs.
3) *Research Opportunities:* We point out several concrete research opportunities that are meaningful to be explored in the future for inspiring and attracting additional follow-up works in this research area.

The remainder of this article is organized as follows. We introduce the preliminary and our proposed taxonomy in Section II. We review the adversarial attacks and the defense strategies in Sections III and IV, respectively. We further discuss the potential research opportunities in Section V before the conclusion in Section VI.

II. Preliminary and Taxonomy

A. General Working Flow of Adversarial Attacks

To facilitate our discussion about the working flow in Fig. 1, we temporarily ignore the attack first and only focus on two inherent modules in CPSs as follows.

1) *System Inputs:* The system input data are collected from various sensors. In addition to the camera, other popular sensors are utilized, namely, surveillance sensor to monitor environments or other systems, speaker to send audio commands, and textual interface to accept text inputs.
2) *System Back-End:* The system back-end applies certain learning techniques [e.g., deep neural networks, recurrent neural networks (RNNs), and regressions] to process the collected input sensor data and generate the desired information to control the system (e.g., control the actuators).

Application Scenarios of Studied Sensors: Before introducing the adversarial attacks that can be launched on the CPS sensors stated above, we first brief the importance and the application scenarios of these sensors in practice.

1) Surveillance Sensors: The surveillance sensors play a vital role in industrial safety, such as smart grids, nuclear power plants, and other monitoring systems. For example, in smart grids, sensors can obtain the physical or chemical states of the key nodes in the system, and these measurements play a decisive role in the estimation of the state of the entire grid, so as to monitor the state of the grid in real time. Another emerging application scenario of surveillance sensors is in the vehicular networks to enable the application of autonomous driving, e.g., the LiDAR sensors [7] are deployed on the vehicle to obtain the distance between itself and potential obstacles in the autonomous driving. Thus, the surveillance sensors are important to industrial and vehicular systems that are closely related to our daily life. Compromising these sensors could pose a direct threat to the industry and driving safety.

2) Audio Speakers: The audio speaker is playing an increasingly important role in the speech recognition technology, widely used in the aspects of our daily life. More specifically, most speech recognition applications rely on the inputs from audio speakers, including the voice assistants on the mobile phones, the voice controller on air conditioners and lighting systems, the voice interaction in vehicle systems and smart wearables, etc. While audio speakers bring great convenience to us, they could also lead to severe privacy and security issues in practice. For example, human–vehicle interaction becomes an important application for vehicles' in-car controls [68]. Recent research works have revealed the possibility of the adversarial attacks through speakers, e.g., the adversarial examples can bypass human perception and fool the neural network of the speech recognition system, which could cause unsafe driving issues.

3) Textual Interfaces: The textual interfaces are used in various scenarios due to the rapid development of text recognition. The textual interfaces are utilized to obtain the textual data in many applications, such as the handwriting input function on mobile devices, the street view text translation on augmented reality (AR) glasses, the user comments on the online shopping platform, and the text input on robots. Therefore, the textual interfaces are commonly available in these daily applications.

4) Other Sensors: In addition to the three types of sensors stated above, some other sensors have also been utilized in the CPS and IoT systems, e.g., the inertial measurement unit (IMU) sensors for smart health [41] and the activity recognition [31], [49] applications. The IMU sensors can provide time-series data about the user, while recent works [20], [35] show that such time-series data can be misclassified in applications by adding perturbations once the attackers obtain the access to such sensor data [30]. As the studies in this category are few in the literature, we thus still focus on the surveillance sensors, audio sensors, and textual interfaces in the rest of this article.

Adversarial Attack: When the adversarial attack is launched, two additional relevant modules need to be further considered.

1) *Perturbation:* To launch the adversarial attack, the attacker first needs to compromise one common input data as victim [9]. Then, for these victim data, the attacker can select an appropriate noise data (perturbation) depending on the desired output of the attacker

from the system back-end, which is the target of the adversarial attack. This perturbation is then added to this compromised input data to fool the neural network in the system back-end. We assume that the generated perturbation works only when it is added to these compromised input data.

2) *Detector:* In addition to fool the neural network of the system for generating the desired output selected by the attacker, the added perturbation should bypass the perception of a detector in the system. In particular, a detector may be a certain deployed data monitor that is responsible for checking the unusual input data and avoiding such data entering the system back-end, such as in the monitoring systems [33]. In some cases, the detector can be users. For example, if the perturbation is added to a voice command, then the intrusion should not be perceived and noticed by humans [9].

Potential Defense Strategies: Fig. 1 shows that only the perturbation module belongs to the attacker. Therefore, in principle, we can defend the adversarial attacks in three modules. From our review of the literature on existing defense designs, we find that they can be appropriately mapped to one of the three modules: 1) defenses from front-end input data; 2) defenses from the middle-end detector; and 3) defenses from the back-end network. This working flow motivates the taxonomy to organize the defense studies in this article, which is briefed in Section II-C and detailed in Section IV.

B. Technical Terms

We explain related technical terms in this article here.

1) *Adversarial Example:* The adversarial example is the compromised common system input data plus the generated perturbation. After processing an adversarial example, the system will produce the desired output of the attacker.
2) *White-Box Attack:* In this attack scenario, the attacker knows the particulars about the neural network of the system to be attacked, such as the structure and the parameters of the network.
3) *Black-Box Attack:* In this attack scenario, the attacker can only access the neural network as a black box to provide input data and observe the corresponding outputs, without knowing any particular of the neural network.
4) *Gray-Box Attack:* This attack is between white-box and black-box attacks, which requires partial knowledge of the victim system's neural network only (it cannot be the full knowledge; otherwise it becomes a white-box attack). The specific form of the partial knowledge[1] could be different in different attacks [11], [27], [37], [54].
5) *Untargeted Attack:* In this type of attack, the goal of the attacker is simply to cause a system error, such as making the actual output of the system different from the original output if no adversarial attack occurs.
6) *Targeted Attack:* In this type of attack, the attacker will not only cause a system error but also specify which particular error to be caused in advance by selecting different perturbations.

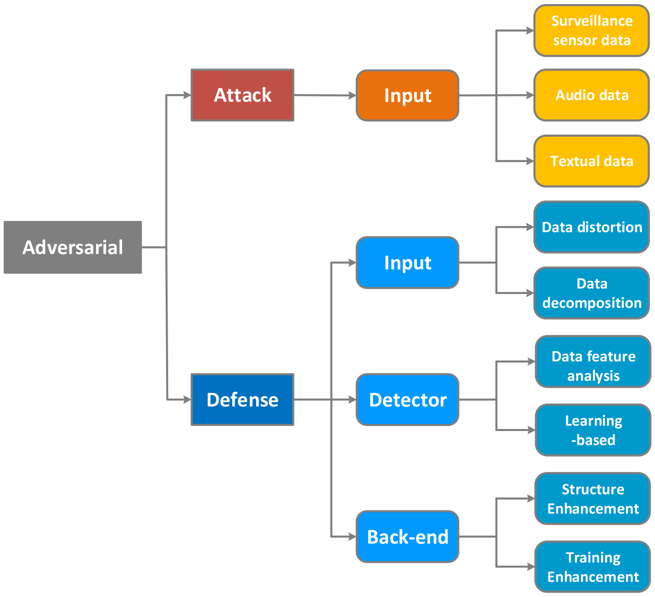

Fig. 2. Taxonomy of adversarial attacks and defense strategies.

C. Taxonomy

At a high level, the taxonomy of this article contains two parts: 1) *attack* and 2) *defense* as illustrated in Fig. 2.

For the adversarial attacks (Section III), we review recent research and find that these attacks are all launched by modifying the system input (Fig. 2). Hence, we first categorize the existing attacks according to the types of their targeted sensor data—surveillance sensor data, audio data, and textual input. We further compare them through six detailed technical aspects.

For the defense strategies (Section IV), we find the existing designs can be mapped to one of the three modules plotted in Fig. 1, namely, defenses from the front-end input data, the middle-end detector, and the back-end network. In particular, data distortion and data decomposition are the major strategies for the first module. The feature analysis and the learning-based method are mainly used in the second module. For the back-end network, existing defenses can be further grouped into two categories: 1) structure enhancement and 2) training algorithm enhancement (Fig. 2).

III. Adversarial Attacks

In this section, we review the existing adversarial attacks for the noncamera-based CPSs from other popular three input sensory data types (Fig. 2). We also compare these data types on the basis of six detailed technical perspectives: 1) white/gray/black box; 2) untargeted/targeted type; 3) learning model; 4) learning task; 5) access to sensors;

[1] For example, the gray-box attack proposed in [27] requires the knowledge used to estimate the gradient of the neural network in the victim speech recognition system merely. They find a way to conduct this estimation without the need to know how the neural network of the victim system is implemented, while most audio adversarial attacks are white box and require to know how the neural network of the victim system is implemented.

and 6) performance. The first two perspectives have been explained in Section II-B. The learning model reflects the complexity of the model, such as the linear or nonlinear model. The learning task describes activities in the system, such as the regression or classification task. In addition, some attacks require accessing the sensor directly, whereas other attacks may not need such a requirement. Finally, the performance of these adversarial attacks is presented and compared.

1) Attacks on Surveillance Sensor Data: The surveillance sensors can monitor different factors, such as temperatures, pressures, and system parameters, and are usually utilized to build surveillance CPSs to monitor environments, fields, or other systems (e.g., forests, nuclear plants, factories, electric power grids, and transportation networks). In such monitoring CPSs, the input data from the surveillance sensors can be viewed as the samples of the targeted environments or systems that are being monitored. The CPS back-end usually applies certain techniques, such as regression, to recover the instant "state" of the targeted system under monitoring. The detector aims to detect any abnormal system states for alarming the system administrator. To attack such a surveillance CPS, the attacker can inject the perturbation into the raw sensory data from the surveillance sensors. The perturbation should be carefully designed to ensure that the detector will not treat it as an abnormal system state when the recovered system state by the system back-end is different from the actual state.

According to this article, we find that most of the adversarial attacks on the surveillance sensor data are under the white-box setting [7], [8], [10], [17], [21], [32], [33], [38], [55]–[57], [60], as the attacker needs this information to know the instant system state when the attacker is searching for the appropriate perturbation. In addition, these works all assume that the attacker can access a subset of sensors. In particular, the attacks proposed in [8], [10], [21], [33], and [55]–[57] are untargeted and their learning models are linear while the learning models of the attacks proposed in [38] and [60] are nonlinear. Furthermore, the attacks proposed in [17] and [32] can be both targeted and untargeted.

For example, to analyze the impact of adversarial attacks on the linear-time-invariant Gaussian systems, Mo and Sinopoli [33] assumed that the system is equipped with a controller, a Kalman filter, and a failure detector. They provided a sufficient condition under which the attacker could modify the system state estimation and fool the detector at the same time. Cárdenas *et al.* [8] introduced a general attack that can be used to launch both integrity attacks and denial-of-service attacks. Xie *et al.* [55] presented an attack against the state estimation in the electricity markets, circumventing the detection of the detector equipped in the system. Similar to [8], [33], and [55], Kosut *et al.* [21] proposed the adversarial attack and the defense strategies for smart grids, which can minimize the knowledge that the attacker must know to launch an attack by a graph-theory-based design. Cui *et al.* [10] designed a normal white-box attack, and Xie *et al.* [56] introduced an integrity attack and formalized the economic impact of this attack. Yadegar *et al.* [57] proposed an attack where the attacker can maliciously manipulate the controller's commands to the actuators and destroy the system. Different from the above works, more advanced nonlinear attacking models were proposed in [38] and [60] and the attacks of these two works are untargeted. Liu *et al.* and Hao *et al.* further proposed both untargeted and targeted adversarial attacks on system state estimation in [32] and [17], respectively. On the other hand, a novel adversarial sensor attack in autonomous driving was proposed in [7], wherein the LiDAR sensor can obtain the distance between itself and potential obstacles by firing laser pulses and obtaining their reflections, thereby helping the vehicle to know the current surrounding road conditions and make a correct driving decision. However, by adding perturbations to the same physical channel of the LiDAR laser pulses, the sensor could capture the erroneous data and cause the vehicle to make wrong driving decisions without a human perception, which could thus raise a serious road safety risk for the autonomous driving.

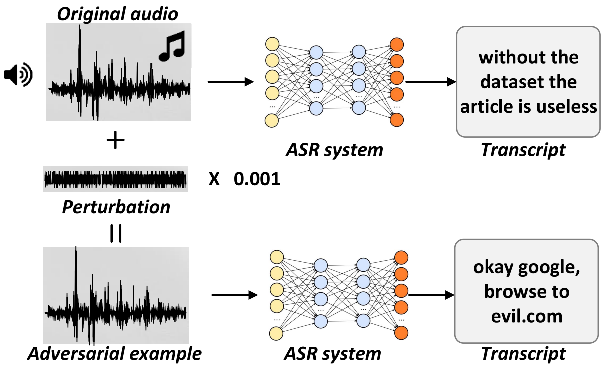

Fig. 3. Illustration of adversarial attacks on the audio data. One audio clip with the transcript of "without the data set this article is useless." To minimize the perception of humans, the amplitude of the added perturbation is very small, and the adversarial example will be nearly the same as the original audio clip to still sounds like the original transcript. However, ASR will recognize it as another one, which is determined on the basis of the selection of the added perturbation. The original audio and the adversarial example are from [9].

In addition to the above white-box attacks, Rahman and Mohsenian-Rad [37] proposed a gray-box attack, in which the adversary has incomplete information of the smart grids. Yu and Chin [63] further proposed a black-box attack where the attacker uses the principle component analysis (PCA) to construct the adversarial attack without the knowledge of the topology matrix of the smart power grids. Moreover, Feng *et al.* [12] found that the neural network can predict the system state more accurately. Based on the deep learning, they proposed a black box and targeted attack in the industrial control systems.

2) Attacks on Audio Data: The microphone sensor is popularly used on various devices, such as mobiles, wearables, voice assistants, and vehicles. This sensor can receive audio commands from the ambient speakers or humans to enable a novel human–computer interaction in the smart-home or smart-driving CPSs by using the speech recognition techniques, such as RNN. In this attack scenario, the received audio data are processed by the system back-end, to convert the audio input to the corresponding transcript. In this case, the detector is the user or the owner of the microphone device. The attacker thus aims to minimize the chance that humans can detect the perturbation added to the audio input as shown in Fig. 3.

For the audio adversarial attacks, people are more interested in the targeted attack, and most existing designs are under the white-box setting [9], [23], [29], [36], [45], [47], [58], [64]. Moreover, the learning models of modern automatic speech recognition (ASR) systems usually adopt neural networks, e.g., RNN. Among those attacks, Carlini and Wagner [9] introduced the targeted audio adversarial examples by applying the iterative optimization algorithm, but the generated adversarial example cannot be transmitted over the air. To make this attack more practical, Yuan *et al.* [64] proposed CommanderSong. When songs are played, the hidden commands can be recognized effectively by the ASR without a human's perception, but this design is mainly effective for songs. Moreover, Yakura and Sakuma [58] enabled the over-the-air audio adversarial attacks. This attack does not require any specific type of audio inputs, but its attacking distance is limited, e.g., about 0.5 m. In addition, Schönherr *et al.* [45] proposed an adversarial attack based on the psychoacoustic hiding where a psychoacoustic model is constructed to control the acoustic signal below the threshold of the human perception. Similar to [45] and [58], Qin *et al.* [36] first developed the imperceptible audio adversarial examples using the psychoacoustic principle and then constructed perturbations to make the audio play over-the-air. Different from the above works, Liu *et al.* [29] introduced a weighted-sampling algorithm to accelerate the generation speed of the audio adversarial examples. Szurley and Kolter [47] proposed an adversarial attack based on the psychoacoustic principle as in [36], [45], and [58]. Kwon *et al.* [23] designed selective audio adversarial examples. They are suitable for the situation, wherein there are one classifier to be protected and another classifier to be attacked. In this situation, the victim classifier will misclassify the adversarial examples, while the protected classifier will classify the adversarial examples correctly. For example, in military applications, the adversarial examples are applied to the enemy's eavesdropping device for decoding the message incorrectly, while the legal device is not affected.

In addition to the white-box and targeted attacks above, there are also untargeted white-box attacks proposed in [1], [14], [18], and [34]. In particular, Gong and Poellabauer [14] designed adversarial examples for the speech paralinguistics applications to make the system accept an illegal user. Iter *et al.* [18] generated adversarial examples for speech recognition to make the recognition system output incorrect results. Neekhara *et al.* [34] and Abdoli *et al.* [1] proposed universal adversarial attacks on the audio data. Neekhara *et al.* [34] presented an attack, where an imperceptible perturbation is constructed to convert any arbitrary audio signal to an adversarial example. Abdoli *et al.* [1] proposed an attack and constructed more advanced audio perturbations, which can be utilized to launch both targeted and untargeted attacks.

Apart from white-box attacks, there are gray-box attacks proposed in [27] and [54]. Li *et al.* [27] proposed an untargeted real-time attack against voice assist's wake-word detection system leveraging the domain transferability of the perturbation. Wu *et al.* [54] added adversarial perturbations in music and voice commands using genetic algorithms with partial knowledge of the speech recognition's structure.

In addition, the black-box attacks against ASR systems were also proposed in [4], [16], [22], and [48]. Specifically, Alzantot *et al.* [4] proposed a targeted attack without the knowledge of model architecture and parameters. They utilized a genetic algorithm in the absence of gradients and this method can be used to construct black-box attacks. Similarly, Taori *et al.* [48] designed a targeted black-box attack leveraging genetic algorithms as well. Han *et al.* [16] presented a black-box and untargeted attack against Google Cloud Speech-to-Text API utilizing genetic algorithms as in [54]. Recently, different from the above white-box or black-box attacks, Kreuk *et al.* [22] presented both black-box and white-box attacks against automatic speaker verification systems. For a white-box attack, the fast gradient sign method (FGSM) was used, while for a black-box attack, the adversarial examples generated by model A were used to attack model B and *vice versa*.

3) Attacks on Textual Data: The textual data plays a vital role in providing CPSs with textual input interfaces, such as recommended systems, machine translation, and sentiment analysis. The textual input data are processed by the neural network of the system back-end to extract useful information, such as the translated language and precise sentiment analysis results. The detector in such an attack scenario is inherently used to check the correctness of the textual data such as spelling typos. To launch such attacks, the added perturbation is also text based. For example, the adversary can manipulate the textual data of movie reviews to ensure that the rating system misclassifies this movie from five stars to one star. A similar attack can be performed to the online shopping by modifying the comments of the product to affect its sale. Given that the goal of this series of attacks is to ensure misclassification of the deep neural network, their targeted learning tasks are classification problems.

In particular, Samanta and Mehta [44] proposed a method to generate adversarial text examples by replacing or deleting some important words or by introducing new words in the text examples. They first calculated the contribution of each word to the classification, and then built a candidate pool for each word, and finally proposed three strategies to generate adversarial examples. This attack is a white-box and untargeted attack, which has been evaluated on two real-world data sets. In addition to [44], Behjati *et al.* [6], Lei *et al.* [25], Yuan *et al.* [65], and Zhu and Woo [70] also presented attacks in white-box setting except that attacks in the first three works are targeted and the last one takes both targeted and untargeted attacks into consideration. Lei *et al.* [25] proposed a gradient-guided greedy algorithm to construct the paraphrase combinations within the candidate set and then conducted word replacements to generate adversarial examples. Yuan *et al.* [65] presented an adaptive adversarial attack against scene text recognition systems without manually searching hyperparameters. Zhu and Woo [70] constructed an adversarial product review using the heuristics algorithm and

TABLE I

Summary of the Adversarial Attacks Launched for Various CPS Sensor Data. In the Third Column, "W/G/B" Refers to White-Box, Gray-Box, and Black-Box Attacks, Respectively. In the Second Last Column, "Access" Means the Need to Access Sensors. In the Last Column, by Default, the Value Indicates the Success Rate of the Proposed Attack Design. Some Works Report the AccR or ErrR of the Victim System After the Proposed Attack Is Launched. Some Works Define Profits (Pfit) That the Attacker Can Obtain From the Proposed Attack. Larger AccR, ErrR, or Pfit Values Indicate More Effective of the Proposed Attacks. On the Other Hand, Some Works Measure the Rate That the Adversarial Examples Can be Detected (DecR), Which Should be as Small as Possible. Finally, Some Works Report Whether the Victim System Can Be Attacked or Not Merely, Which Is Denoted as "Success" in the Table

Adversarial Attacks	Sensor Data	W/G/B	Type	Learning Model	Learning Task	Access	Performance
Mo *et al.* [33]	Surveillance	White	Untargeted	Linear Model	Regression	Subset	Success
Cárdenas *et al.* [8]	Surveillance	White	Untargeted	Linear Model	Regression	Subset	100%
Xie *et al.* [55]	Surveillance	White	Untargeted	Linear Model	Regression	Subset	2.5–9.76% (Pfit)
Kosut *et al.* [21]	Surveillance	White	Untargeted	Linear Model	Regression	Subset	0–95%
Cui *et al.* [10]	Surveillance	White	Untargeted	Linear Model	Regression	Subset	50–66.67%
Xie *et al.* [56]	Surveillance	White	Untargeted	Linear Model	Regression	Subset	2.4–9.46% (Pfit)
Yadegar *et al.* [57]	Surveillance	White	Untargeted	Linear Model	Regression	Subset	Success
Rahman *et al.* [38]	Surveillance	White	Untargeted	Nonlinear Model	Regression	Subset	0% or 100%
Yang *et al.* [60]	Surveillance	White	Untargeted	Nonlinear Model	Regression	Subset	0.4–100%
Liu *et al.* [32]	Surveillance	White	Both	Linear Model	Regression	Subset	100%
Hao *et al.* [17]	Surveillance	White	Both	Linear Model	Regression	Subset	25–100%
Rahman *et al.* [37]	Surveillance	Gray	Untargeted	Linear Model	Regression	Subset	0–15% (DecR)
Yu *et al.* [63]	Surveillance	Black	Untargeted	Nonlinear Model	Regression	Subset	0–100%
Feng *et al.* [12]	Surveillance	Black	Targeted	Neural Network	Regression	Subset	40–50%
Cao *et al.* [7]	Surveillance	White	Targeted	Neural Network	Classification	Subset	75%
Carlini *et al.* [9]	Audio	White	Targeted	Neural Network	Classification	No need	100%
Yuan *et al.* [64]	Audio	White	Targeted	Neural Network	Classification	No need	96–100%
chönherr *et al.* [45]	Audio	White	Targeted	Neural Network	Classification	No need	98%
Yakura *et al.* [58]	Audio	White	Targeted	Neural Network	Classification	No need	100%
Qin *et al.* [36]	Audio	White	Targeted	Neural Network	Classification	No need	100%
Liu *et al.* [29]	Audio	White	Targeted	Neural Network	Classification	No need	100%
Szurley *et al.* [47]	Audio	White	Targeted	Neural Network	Classification	No need	100%
Kwon *et al.* [23]	Audio	White	Targeted	Neural Network	Classification	No need	91.67%
Neekhara *et al.* [34]	Audio	White	Untargeted	Neural Network	Classification	No need	89.06%
Gong *et al.* [14]	Audio	White	Untargeted	Neural Network	Classification	No need	31–75% (ErrR)
Iter *et al.* [18]	Audio	White	Untargeted	Neural Network	Classification	No need	93%
Abdoli *et al.* [1]	Audio	White	Both	Neural Network	Classification	No need	83.1–85.4%
Li *et al.* [27]	Audio	Gray	Untargeted	Neural Network	Classification	No need	81.5–82.4% (AccR)
Wu *et al.* [54]	Audio	Gray	Targeted	Neural Network	Classification	No need	20–90%
Alzantot *et al.* [4]	Audio	Black	Targeted	Neural Network	Classification	No need	87%
Taori *et al.* [48]	Audio	Black	Targeted	Neural Network	Classification	No need	35%
Han *et al.* [16]	Audio	Black	Untargeted	Neural Network	Classification	No need	86%
Kreuk *et al.* [22]	Audio	White/Black	Targeted	Neural Network	Classification	No need	48–69.9% (AccR)
Samanta *et al.* [44]	Textual	White	Untargeted	Neural Network	Classification	No need	13.3–42.0% (AccR)
Lei *et al.* [25]	Textual	White	Targeted	Neural Network	Classification	No need	20–90%
Yuan *et al.* [65]	Textual	White	Targeted	Neural Network	Classification	No need	99.99%
Zhu *et al.* [70]	Textual	White	Targeted	Neural Network	Classification	No need	10% (AccR)
Behjati *et al.* [6]	Textual	White	Both	Neural Network	Classification	No need	43% (AccR)
Dasgupta *et al.* [11]	Textual	Gray	Untargeted	Neural Network	Classification	No need	10–80%
Alzantot *et al.* [5]	Textual	Black	Targeted	Neural Network	Classification	No need	70–97%
Ren *et al.* [42]	Textual	Black	Targeted	Neural Network	Classification	No need	33.4–82.9% (AccR)
Vijayaraghavan *et al.* [50]	Textual	Black	Untargeted	Neural Network	Classification	No need	71.45% (AccR)
Wang *et al.* [53]	Textual	Black	Untargeted	Neural Network	Classification	No need	40–75% (AccR)
Li *et al.* [26]	Textual	Black	Untargeted	Neural Network	Classification	No need	20–38% (AccR)
Gao *et al.* [13]	Textual	Black	Untargeted	Neural Network	Classification	No need	59–61% (AccR)
Jin *et al.* [19]	Textual	Black	Untargeted	Neural Network	Classification	No need	95.8–99.7%
Liang *et al.* [28]	Textual	White/Black	Targeted	Neural Network	Classification	No need	80.5%
Wang *et al.* [51]	Textual	White/Black	Both	Neural Network	Classification	No need	69.7% (AccR)

word replacements. Finally, Behjati *et al.* [6] designed universal adversarial perturbations for texts based on the gradient projection to launch the attacks.

Dasgupta *et al.* [11] proposed a gray-box and untargeted attack by integrating gradient-based white-box methods and black-box methods where the attacker can query the victim model to obtain output. Moreover, there are black-box attacks on textual data proposed in [5], [13], [19], [26], [28], [42], [50], [51], and [53]. Similar to [4], a gradient-free genetic algorithm was used to construct natural language adversarial examples under black-box setting in [5].

Ren *et al.* [42] proposed a black-box and targeted attack. They constructed natural language adversarial examples through a word replacement order while maintaining the semantic similarity to the original sentence. Besides, the authors of [13], [19], [26], [50], and [53] proposed black-box and untargeted attacks. Vijayaraghavan and Roy [50] proposed a reinforcement learning framework to generate adversarial examples. Wang *et al.* [53] designed an improved genetic algorithm attack method for synonyms substitution and Li *et al.* [26] proposed universal rules for constructing imperceptible adversarial examples automatically based on the coevolutionary optimization algorithm. Jin *et al.* [19] proposed a black-box and untargeted attack against text classification and entailment. In addition, Gao *et al.* [13] presented an algorithm to generate adversarial text sequences with a black-box setting. In this article, important words are determined first and then modified slightly to control the distance to the original textual input.

Moreover, two more advanced attacks are presented by Liang *et al.* [28] and Wang *et al.* [51]. Liang *et al.* [28] proposed a targeted attack strategy and took both white-box and black-box settings into consideration. The adversarial examples generated by this article can fool both word-level classifiers and character-level classifiers. Wang *et al.* [51] also designed both black-box and white-box attacks, where targeted and untargeted scenarios are further taken into consideration. They proposed a unified framework to generate adversarial text and leverage autoencoder to ensure the grammar correctness.

4) Summary and Comparison: The key attributes of all the attacks reviewed in this section are tabulated and compared in Table I. In addition, we also include their reported performance in the table and summarize it in the following. By default, the value indicates the success rate of the proposed attack design. However, some works report other performance metrics, e.g., accuracy reduction, error rate (ErrR) profit, detection rate, etc., which are explained in the table's caption.

1) Attacks on Surveillance Sensor Data: Most related works [7], [8], [10], [12], [17], [21], [32], [38], [60], [63] report the success rates of the attacks. For example, the success rate of attack in [12] is from 40% to 50%, which means that 40%–50% of attacks can achieve the attack targets. In addition, the success rate of the attack in [7] is improved to 75%. Although the performance of these attacks is not very high, the ratios can satisfy the requirement to successfully attack the target system and bypass the detector. The success rates of the attacks in [8], [10], [17], [21], [32], [38], [60], and [63] are between 50% and 100%, and most of the success rates are above 95%. The profit value (Pfit) of [55] and [56] is between 2% and 10%, which means that the attacker can obtain 2%–10% profits of the victim system. In addition, the "DecR" value of [37] is between 0% and 15%, i.e., the adversarial examples from [37] have a small chance to be detected. Other attacks on surveillance sensors [33], [57] mainly show that they can succeed in their attacks.

2) Attacks on Audio Data: Most attacks in this category [1], [4], [9], [16], [18], [23], [29], [34], [36], [45], [47], [54], [58], [64] have success rates above 85%, and all of the audio adversarial examples generated in [9], [29], [36], [47], [58], and [64] can be classified as the target class. The attack success rate proposed in [48] is relatively low, i.e., 35%. However, this article considers both the success rate of the attack and the similarity of the adversarial examples to the original samples to achieve a good audio quality, i.e., the adversarial examples maintain more than 85% similarity with the original samples. In addition, the ErrR of the attacks in [14] is from 31% to 75%, which means the ErrR of the victim system is relatively high, indicating the success of the attack. The accuracy reduction rates (AccRs) of [27] and [22] are from 81.5% to 82.4% and from 48% to 69.9%, respectively, and these rates indicate that the classification accuracy rates of victim systems are reduced because of the successful attacks.

3) Attacks on Textual Data: The performance of the attacks proposed in [5], [19], [25], and [65] are relatively higher compared with [6], [26], [44], and [70]. Success rates or accuracy reduction of attacks proposed in [11], [13], [28], [42], [50], [51] and [53] are between 10% and 85%. The reason for this difference is that different works use different attack methods in different scenarios and all these adversarial attacks can succeed. For example, the success rate of [19] is between 95.8% and 99.7%, which indicates that more than 95% of the generated adversarial textual examples can be misclassified by the attacked models without the perception of the detector. Although the success rate of the attack in [26] is 20%–38%, the attack is more advanced as the success rate of this article is a universal rule success rate, i.e., universal rule can convert 20%–38% of the textual data to adversarial examples without searching the perturbation for each adversarial example.

IV. Defense Strategies

In this section, we further discuss existing defense strategies in the literature. As stated in Section II-C, these strategies are from three aspects, e.g., the defenses from the front-end the input data, middle-end detector, and the system back-end.

A. Defenses From Input Data

1) Data Distortion: To minimize the chance to be detected by the detector, the attacker needs to minimize the volume of the perturbation bits added to the original input data. But one direct consequence is that the resulting adversarial example will not be reliable and robust enough to the external noises or data distortions. Inspired by this opportunity, some existing works propose to further process the input data to remove its adversarial property before the system back-end processes it.

Following this idea, data compression, data coding, data randomization, and adding additional noises could be candidate methods for this type of defense strategy. One representative example is the defense proposed in [64], wherein Yuan *et al.* proposed to add audio turbulence or conduct audio squeezing to defend the audio adversarial attack CommanderSong, proposed in [64] as well. For the audio turbulence, the authors find that adding noise to adversarial examples can degrade the success rate of the CommanderSong attack. For audio squeezing, downsampling the input audio can also degrade the success rate of the attack. In particular, if the current audio input is an adversarial example, the outputs of the audio recognition system will be different with and without these two defense operations. In addition, Rajaratnam and Kalita [40] proposed a method in which a frequency band of audio signal input is flooded with random noise so that the adversarial examples can be more easily detected. Rajaratnam *et al.* [39] further leveraged the principle of speech coding and decoding to detect the audio adversarial examples on ASR. Kwon *et al.* [24] proposed an audio modification defense method by adding distortion to audio signals and this method is a kind of data turbulence. Zhang et al. [67] proposed a defense mechanism using code modulation and audio compression and these two methods can be categorized into data turbulence and data squeeze, respectively. On the other hand, data squeeze can also be applied to defend against the attacks on the textual data. Rosenberg *et al.* [43] proposed a defense method called sequence squeezing by leveraging the idea that the squeezed adversarial examples can be detected more easily.

2) Data Decomposition: In addition to the data distortion, data decomposition is another kind of defense method from the system input perspective. In particular, Hao *et al.* [17] found that the surveillance sensor measurement data shows the low rank and sparse structure. Thus, leveraging these two features of sensor data, the original sensor measurement and the added perturbation can be recovered, respectively, from the adversarial examples using the decomposition algorithm. Therefore, this could be a defense strategy against adversarial attacks on the surveillance sensor measurement.

B. Defenses From Detector

1) Data Feature Analysis: For the above three kinds of sensor data, researchers find that it is also possible to conduct data feature analysis to recognize adversarial examples directly.

For surveillance data, various data statistics analysis methods can be adopted. For instance, Kosut *et al.* [21] proposed a defense design based on the generalized likelihood ratio test with two hypotheses—a null hypothesis indicating no attack and a nonnull hypothesis indicating a detected attack. Once the system input data is obtained, the data will be utilized to compute a statistic result first and the result is then compared with a threshold to determine whether the current input is an adversarial example or not, i.e., a null or nonnull hypothesis. Similarly, Cárdenas *et al.* [8] proposed two detection methods, i.e., sequential detection and change detection. In particular, the sequential detection was based on the sequential probability ratio test, while the change detection was designed on the basis of the cumulative sum statistic. The sequential detection aims to minimize the system input measurement to calculate the statistic result, while the change detection aims to detect a possible change from a null hypothesis to a nonnull hypothesis to finally determine whether the input is an adversarial example. In addition to [8] and [21], Yang *et al.* [60] proposed spatial-based and temporal-based detection methods similar to [8]. For spatial-based detection, the hypothesis test was applied to decide whether the system is under an attack, while in temporal-based detection, the cumulative sum statistic was computed to be compared with the threshold. Ye and Zhang [62] proposed a summation detector to detect false data injection attacks in control systems. This detector is different from previous ones since it not only uses the current information but also utilizes historical information to predict potential threats.

For audio data, Yang *et al.* [61] identified the audio adversarial examples by inspecting the temporal consistency. In particular, the authors find that the temporal consistency exists between the two halves of a complete audio. However, there is no temporal consistency if the audio is an adversarial example. Utilizing this feature, the audio adversarial examples can be identified according to this design.

For textual data, the data feature analysis-based defense has been applied in [3] and [52]. In particular, Wang *et al.* [52] observed that misspelled words may exist in adversarial examples as they are usually generated by modifying, inserting, or replacing important words or characters. Alshemali and Kalita [3] proposed a spell-checking system using the contextual and frequency information to correct misspellings. Thus, checking the spelling of the textual data by the detector can be regarded as the first line to resist the textual adversarial examples and protect the system from adversarial attacks.

2) Learning-Based: In addition to the data feature analysis defense methods, learning-based defense methods have been proposed recently due to the rapid development of machine learning. This kind of method usually needs to extract features first and then train a decision model to decide whether the input is an adversarial example. For example, Yan *et al.* [59] proposed a machine learning-based attack detection scheme by deriving features and applying extreme learning machine techniques. Zhou *et al.* [69] designed a secure control framework based on reinforcement learning to defend against attacks on the surveillance sensors in CPSs.

C. Defenses From System Back-End

Finally, the defense can be also applied by enhancing the system back-end design by augmenting the system back-end structure itself directly or the network training.

1) Structure Enhancement: In the literature, the system back-end structure can be enhanced from the *network parameter* and *network structure* two aspects.

To improve the network parameter for defending adversarial attacks, one effective way is called *defense distillation* [2]. Generally speaking, the key idea of this technique is to train the neural network twice. The first-round training is a standard training, by using the input data and its corresponding label from the data set to train the network. In the second round, this technique will further use the input data and the corresponding logits from the network (trained in the first round) as the label to train another network. With this design, the authors find that the neural network (trained in the second round) will become more reliable and robust to adversarial examples. However, Soll *et al.* [46] evaluated the defensive distillation for defending against adversarial examples on the textual data and they found that this method can only have a limited impact on improving the robustness of the neural network.

On the other hand, to enhance the neural network's structure, Akhtar and Mian [2] proposed to apply the emerging framework named generative adversarial network (GAN). With this framework, the network structure can be extended with one more component called discriminator. Moreover, to train the network, we will also include certain adversarial examples in advance. Therefore, the purpose of the discriminator is to distinguish these adversarial examples in the training data set. After the network training, the network itself has

TABLE II
SUMMARY OF THE DEFENSE STRATEGIES TO THE ADVERSARIAL ATTACKS. IN THE LAST COLUMN, SOME WORKS REPORT THEIR PERFORMANCE USING THE TARGETED ATTACK'S ACCURACY AFTER THE PROPOSED DEFENSE IS APPLIED, WHICH IS DENOTED AS "(ATTA)." A SMALL ATTA VALUE MEANS THE DEFENSE DESIGN IS EFFECTIVE. SOME WORKS REPORT THE CLASSIFICATION ACCURACY "(CLFA)," ERRR, FPR, FNR, AND TPR OF THE VICTIM SYSTEM AFTER THE PROPOSED DEFENSE IS APPLIED. SOME WORKS REPORT THE DETECTION PERCENTAGE OF THE TARGETED ADVERSARIAL ATTACKS, WHICH ARE DENOTED AS "(DECP)." A LARGE DECP VALUE MEANS THE DEFENSE DESIGN IS EFFECTIVE

Defenses	Modules	Defense Methods	Usage Scope	Performance
Yuan *et al.* [64]	System Input	Data Distortion	Defend against the practical white-box and targeted audio attacks on ASR systems	0–8% (AttA)
Rajaratnam *et al.* [39]	System Input	Data Distortion	Mitigate and detect targeted audio adversarial examples	93–98% (DecP)
Rajaratnam *et al.* [40]	System Input	Data Distortion	Defend against the black-box, targeted and non-complex audio attacks	91.8% (DecP)
Rosenberg *et al.* [43]	System Input	Data Distortion	Defend against black-box, white-box and untargeted adversarial examples for RNN networks	15% (AttA)
Kwon *et al.* [24]	System Input	Data Distortion	Detect the white-box and targeted audio adversarial examples	6.21% (AttA)
Zhang *et al.* [67]	System Input	Data Distortion	Defend against the white-box and targeted attacks on cloud-aided ASR systems	0.02% (FPR) / 4.62–10.69% (FNR)
Hao *et al.* [17]	System Input	Data Decomposition	Defend against the white-box, targeted and untargeted malicious data attacks in smart grids	44.7–100% (DecP)
Kosut *et al.* [21]	Detector	Data Feature Analysis	Defend against the white-box and untargeted malicious data attacks in smart grids	5–100% (DecP)
Cárdenas *et al.* [8]	Detector	Data Feature Analysis	Detect the white-box and untargeted attacks in the process control systems	Not available
Yang *et al.* [60]	Detector	Data Feature Analysis	Defend against white-box and untargeted false data injection attacks in power systems	99.6% (DecP)
Yang *et al.* [61]	Detector	Data Feature Analysis	Mitigate the black-box, white-box, targeted and non-adaptive audio adversarial examples	2.1% (AttA)
Alshemali *et al.* [3]	Detector	Data Feature Analysis	Defend against the black-box adversarial attacks for texts	88.66% (ClfA)
Ye *et al.* [62]	Detector	Data Feature Analysis	Defend against malicious data attacks for surveillance data	Not available
Yan *et al.* [59]	Detector	Learning-based	Detect the adversarial attacks for surveillance sensor data at the physical layer	99.46% (TPR)
Zhou *et al.* [69]	Detector	Learning-based	Mitigate the adversarial attacks for surveillance sensor data	Not available
Zeng *et al.* [66]	System Back-End	Structure Enhancement	Detect the white-box and black-box audio adversarial examples	99.88% (DecP)
Soll *et al.* [46]	System Back-End	Structure Enhancement	Defend against untargeted adversarial attacks for texts	96–98% (AttA)
Goodfellow *et al.* [15]	System Back-End	Training Enhancement	Defend against the adversarial attacks for texts and images	20% (ErrR)

the ability to recognize and be against the adversarial examples. In addition to the GAN-based design, gradient hiding is another effective method to defend this attack [2]. Its key idea is to adjust the network structure to make the gradient of the network be zero (nondifferentiable), e.g., utilizing some nondifferentiable model such as the K-nearest neighbor (KNN) model. By doing this, many existing adversarial example generation methods, such as FGSM [2], will become not effective anymore, as they rely on the gradient descent to generate adversarial examples. In addition to the above methods, an approach to defense against audio adversarial examples based on the multiversion programming (MVP) was proposed in [66], which is designed to defend against software flaws. Different programs can be viewed to be developed independently. Therefore, different versions of the programs may have their own unique flaws. By comparing the output results of different versions of the program, the flaws can be detected, thus the robustness of the whole software system can be improved. Inspired by this idea, multiple ASR systems can be run in parallel and the adversarial example can be regarded as the software flaw. Once the input audio is passed into all the ASRs, a similar metric based on the recognition results will be calculated. Then, the system will show the detection result according to the metric to indicate whether the input audio is an adversarial example.

2) Network Training Enhancement: Even the network structure itself keeps unchanged, researchers find adversarial attacks may still be defended by improving the network training alone. In particular, Goodfellow *et al.* [15] proposed a simple yet effective method by including adversarial examples directly in the training data set. However, the label of each adversarial example is set to be the same as that of the original input without adding the perturbation. After this improvement on the network training, the network becomes aware of various adversarial examples in advance and the added perturbation bits will be simply treated as noises. Goodfellow *et al.* [15]

used the textual input data as a concrete example to investigate the efficacy of this defense strategy. Its limitation is that the network developer needs dedicated efforts to generate a set of appropriate adversarial samples in advance.

D. Summary and Comparison

The key attributes of the defense strategies reviewed in this section are tabulated and compared in Table II. Moreover, we also include their performance in the table and summarize it in the following. Similar to the attack part, some different performance metrics are utilized in the evaluation of these defense designs, which are explained in the caption of Table II.

1) Defense From System Input: As Table II shows, the success rates of [24], [43], and [64] are between 0% and 15%, indicating that when these defense methods are applied, the success rates of the attacks are lower than 15%. In addition, the precision of detecting attacks in [17], [39], and [40] is above 90% and the result indicates that most of the adversarial examples can be detected. The false positive rate (FPR) and false negative rate (FNR) of the victim system in [67] are 0.02% and 4.62%–10.69%, respectively, which indicates that the defense method proposed in this article can also defend against the targeted audio adversarial attacks effectively.

2) Defense From Detector: The success rate of the original attack reduces to 2.1% with the design from [61], which is quite small, indicating that the defense method is effective. The detection percentages of adversarial examples of [21] and [60] are above 90%, showing that most adversarial examples can be detected. Moreover, the classification accuracy of [3] and the true positive rate (TPR) of [59] are 88.66%–99.46%, respectively, indicating the effectiveness of these defense methods. In general, the defense methods proposed from the detector aspect can defend against adversarial attacks effectively.

Defense From System Back-End: The detecting percentage of [66] is relatively high, which indicates that the defense method can effectively defend the audio adversarial attacks. The ErrR of [15] is around 20%, which is also effective. However, the success rate of the adversarial attacks on the textual data is still high, even the defensive distillation method [46] is applied.

V. Discussion and Research Opportunities

A. Comparison With the Field in Computer Vision

In this section, we first summarize some differences of the adversarial attacks and defenses discussed in this article compared with the field in computer vision.

1) Perception of Adversarial Examples: Compared with the images or videos with rich information, the added perturbation for the noncamera data might be easier to be noticed by the detector or the user, like audio clips or textual information. The attacker has to minimize the volume of added perturbation bits, which however can make the adversarial examples less robust and reliable to many external factors in practice.

2) Learning Model: As most noncamera sensory data are the streaming-like data, the adopted learning model is usually RNN that further considers the time dimension. Because of the relatively high computation complexity in RNN, e.g., due to the cyclical computations, the adversarial generation also becomes time consuming (since most attacks are white-box attacks), e.g., it takes more than 1 h to generate only one adversarial example in [9].

3) Learning Task: The adversarial attacks in the computer vision domain mainly focuses on the classification problem. However, in noncamera-based CPSs as summarized in Table I, the problem includes both classification and regression, which could further complicate the attack and defense designs.

B. Research Opportunities

To further explore this research area in the future, we propose two concrete research opportunities as follows.

1) Pushing the Limit of Adversarial Example's Efficacy: Existing attack designs could limit the efficacy of adversarial attack from at least two aspects. First, as stated in the perception of adversarial examples above, existing adversarial examples are vulnerable to the external noise or the manual data preprocessing, which however cannot be avoided in practice. For instance, when an audio adversarial example is played by a speaker, it has to go through the acoustic channel to reach the microphone. Existing studies indeed observe the over-the-air transmission could undermine the adversarial attack [9]. Second, the generated perturbation is only effective when it is added to the specific targeted input data compromised before, which is named as the *targeted input data*. This could cause the generation of adversarial examples less efficient and less scalable.

Although these two limitations appear to be different problems, through this article, we observe that they share the same rationale behind—the generation of perturbation never takes the potential impacts (e.g., from ambient noise, data distortion, or different input versions) into consideration. To overcome this issue, one simple yet effective strategy is to collect different variants of the targeted input data and then generate the perturbation to best fit this series of inputs. For instance, the variants could include audio clips spoken by different people. The attacker can further play and receive these audio clips first, and then use the received audio clips to generate perturbation. In summary, an attacker should factor the possible impacts into perturbation's generation in advance so that generated adversarial examples could be robust against these impacts later. The recent study [58] achieves an initial success to generate over-the-air audio adversarial examples. However, some issues still remain open and we suggest two further possible investigations.

1) Optimize the selection of the targeted input data, to ensure both the perturbation quality and the overhead caused.
2) The generated perturbation may still be tailored for these selected input data set. We need to further remove their specific features to make the perturbation more general.

2) Multimodality Defense Design and Assessment: In literature, existing defense designs are mainly single modality based, i.e., defend from one of the system input, detector, and back-end three components. This limits the defense strategy normally being effective to one type of input data or application scenario in practice. The defense may not be effective anymore when the system is deployed in a new environment or the attacker could easily bypass the proposed defense strategy. Therefore, in the future, we propose a possible future study to explore the multimodality strategy design for achieving a comprehensive and holistic defense design and making CPSs more secure and robust against adversarial examples. Along this direction, we suggest the following investigations.

1) We need to understand the reason why each single modality defense is effective to certain types of the input data, rather than other data types. Afterward, we need to extract the key features or attributes that majorly contribute to the defense for every single modality, so that we can quantitatively measure the efficacy of each modality across different data types.
2) With the research from the first step, we propose to further research the mechanism for finding the best combination to integrate different potential modalities. However, we need to differentiate them in the integration according to the reliability and confidence from different modalities to optimize the overall fusion.
3) Although the comprehensive and holistic defense can be achieved by the multimodality design, as the noncamera-based CPSs designs may have more complicated learning models and tasks stated above, we should also consider the balance between the defense performance and system cost. This tradeoff is expected to be adapted automatically according to the system configuration.

VI. Conclusion

In this article, we surveyed adversarial attacks and defenses for the CPSs using various CPS sensor data, including surveillance sensor data, audio data, and textual data. We first introduced a general working flow of adversarial attacks on these CPSs, which also indicates the possible strategies to perform defenses in the system. We performed a systematic discussion on existing attack and defense designs with detailed comparison studies in this article. Finally, we pointed out several concrete future research opportunities in this research area.

References

[1] S. Abdoli, L. G. Hafemann, J. Rony, I. B. Ayed, P. Cardinal, and A. L. Koerich, "Universal adversarial audio perturbations," 2019. [Online]. Available: arXiv:1908.03173.

[2] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14410–14430, 2018.

[3] B. Alshemali and J. Kalita, "Toward mitigating adversarial texts," *Int. J. Comput. Appl.*, vol. 178, no. 50, pp. 1–7, 2019.

[4] M. Alzantot, B. Balaji, and M. Srivastava, "Did you hear that? Adversarial examples against automatic speech recognition," 2018. [Online]. Available: arXiv:1801.00554.

[5] M. Alzantot, Y. Sharma, A. Elgohary, B.-J. Ho, M. Srivastava, and K.-W. Chang, "Generating natural language adversarial examples," 2018. [Online]. Available: arXiv:1804.07998.

[6] M. Behjati, S.-M. Moosavi-Dezfooli, M. S. Baghshah, and P. Frossard, "Universal adversarial attacks on text classifiers," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2019, pp. 7345–7349.

[7] Y. Cao *et al.*, "Adversarial sensor attack on LIDAR-based perception in autonomous driving," in *Proc. ACM Conf. Comput. Commun. Security*, 2019, pp. 2267–2281.

[8] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry, "Attacks against process control systems: Risk assessment, detection, and response," in *Proc. ACM Symp. Inf. Comput. Commun. Security,*, 2011, pp. 355–366.

[9] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *Proc. IEEE Symp. Security Privacy Workshops*, 2018, pp. 1–7.

[10] S. Cui, Z. Han, S. Kar, T. T. Kim, H. V. Poor, and A. Tajer, "Coordinated data-injection attack and detection in the smart grid: A detailed look at enriching detection solutions," *IEEE Signal Process. Mag.*, vol. 29, no. 5, pp. 106–115, Sep. 2012.

[11] P. Dasgupta, J. B. Collins, and A. Buhman, "Gray-box techniques for adversarial text generation," in *Proc. AAAI Fall Symp. ALEC*, 2018, pp. 17–23.

[12] C. Feng, T. Li, Z. Zhu, and D. Chana, "A deep learning-based framework for conducting stealthy attacks in industrial control systems," 2017. [Online]. Available: arXiv:1709.06397.

[13] J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi, "Black-box generation of adversarial text sequences to evade deep learning classifiers," in *Proc. IEEE Symp. Security Privacy Workshops*, 2018, pp. 50–56.

[14] Y. Gong and C. Poellabauer, "Crafting adversarial examples for speech paralinguistics applications," 2017. [Online]. Available: arXiv:1711.03280.

[15] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014. [Online]. Available: arXiv:1412.6572.

[16] J. K. Han, H. Kim, and S. S. Woo, "Nickel to LEGO: Using Foolgle to create adversarial examples to fool Google cloud speech-to-text API," in *Proc. ACM Conf. Comput. Commun. Security*, 2019, pp. 2593–2595.

[17] J. Hao, R. J. Piechocki, D. Kaleshi, W. H. Chin, and Z. Fan, "Sparse malicious false data injection attacks and defense mechanisms in smart grids," *IEEE Trans. Ind. Informat.*, vol. 11, no. 5, pp. 1–12, Oct. 2015.

[18] D. Iter, J. Huang, and M. Jermann, "Generating adversarial examples for speech recognition," Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, Rep., 2017.

[19] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, "Is bert really robust? Natural language attack on text classification and entailment," 2019. [Online]. Available: arXiv:1907.11932.

[20] F. Karim, S. Majumdar, and H. Darabi, "Adversarial attacks on time series," 2019. [Online]. Available: arXiv:1902.10755.

[21] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on smart grid state estimation: Attack strategies and countermeasures," in *Proc. IEEE SmartGridComm*, 2010, pp. 220–225.

[22] F. Kreuk, Y. Adi, M. Cisse, and J. Keshet, "Fooling end-to-end speaker verification with adversarial examples," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2018, pp. 1962–1966.

[23] H. Kwon, Y. Kim, H. Yoon, and D. Choi, "Selective audio adversarial example in evasion attack on speech recognition system," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 526–538, Jun. 2019.

[24] H. Kwon, H. Yoon, and K.-W. Park, "POSTER: Detecting audio adversarial example through audio modification," in *Proc. ACM Conf. Comput. Commun. Security*, 2019, pp. 2521–2523.

[25] Q. Lei, L. Wu, P.-Y. Chen, A. G. Dimakis, I. S. Dhillon, and M. Witbrock, "Discrete adversarial attacks and submodular optimization with applications to text classification," in *Proc. SysML*, 2019.

[26] D. Li, D. V. Vargas, and S. Kouichi, "Universal rules for fooling deep neural networks based text classification," 2019. [Online]. Available: arXiv:1901.07132.

[27] J. Li, S. Qu, X. Li, J. Szurley, J. Z. Kolter, and F. Metze, "Adversarial music: Real world audio adversary against wake-word detection system," in *Proc. NIPS*, 2019, pp. 11908–11918.

[28] B. Liang, H. Li, M. Su, P. Bian, X. Li, and W. Shi, "Deep text classification can be fooled," 2017. [Online]. Available: arXiv:1704.08006.

[29] X. Liu, K. Wan, and Y. Ding, "Adversarial attack on speech-to-text recognition models," 2019. [Online]. Available: arXiv:1901.10300.

[30] Y. Liu and Z. Li, "ALEAK: Privacy leakage through context-free wearable side-channel," in *Proc. IEEE INFOCOM*, 2018, pp. 1232–1240.

[31] Y. Liu, Z. Li, Z. Liu, and K. Wu, "Real-time arm skeleton tracking and gesture inference tolerant to missing wearable sensors," in *Proc. ACM MobiSys*, 2019, pp. 287–299.

[32] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Trans. Inf. Syst. Security*, vol. 14, no. 1, pp. 1–33, 2011.

[33] Y. Mo and B. Sinopoli, "False data injection attacks in control systems," in *Proc. SCS*, 2010, pp. 1–7.

[34] P. Neekhara, S. Hussain, P. Pandey, S. Dubnov, J. McAuley, and F. Koushanfar, "Universal adversarial perturbations for speech recognition systems," 2019. [Online]. Available: arXiv:1905.03828.

[35] I. Oregi, J. Del Ser, A. Perez, and J. A. Lozano, "Adversarial sample crafting for time series classification with elastic similarity measures," in *Proc. Int. Symp. Intell. Distrib. Comput.*, 2018, pp. 26–39.

[36] Y. Qin, N. Carlini, I. J. Goodfellow, G. Cottrell, and C. Raffel, "Imperceptible, robust, and targeted adversarial examples for automatic speech recognition," 2019. [Online]. Available: arXiv:1903.10346.

[37] M. A. Rahman and H. Mohsenian-Rad, "False data injection attacks with incomplete information against smart power grids," in *Proc. IEEE GLOBECOM*, 2012, pp. 3153–3158.

[38] M. A. Rahman and H. Mohsenian-Rad, "False data injection attacks against nonlinear state estimation in smart power grids," in *Proc. IEEE PESGM*, 2013, pp. 1–5.

[39] K. Rajaratnam, B. Alshemali, K. Shah, and J. Kalita. (2018). *Speech Coding and Audio Preprocessing for Mitigating and Detecting Audio Adversarial Examples on Automatic Speech Recognition*. [Online]. Available: http://cs.uccs.edu/∼jkalita/work/reu/REU2018/07Rajaratnam.pdf

[40] K. Rajaratnam and J. Kalita, "Noise flooding for detecting audio adversarial examples against automatic speech recognition," in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol.*, 2018, pp. 197–201.

[41] D. Ravi, C. Wong, B. Lo, and G.-Z. Yang, "A deep learning approach to on-node sensor data analytics for mobile or wearable devices," *IEEE J. Biomed. Health Informat.*, vol. 21, no. 1, pp. 56–64, Jan. 2017.

[42] S. Ren, Y. Deng, K. He, and W. Che, "Generating natural language adversarial examples through probability weighted word saliency," in *Proc. Assoc. Comput. Linguist.*, 2019, pp. 1085–1097.

[43] I. Rosenberg, A. Shabtai, Y. Elovici, and L. Rokach, "Defense methods against adversarial examples for recurrent neural networks," 2019. [Online]. Available: arXiv:1901.09963.

[44] S. Samanta and S. Mehta, "Towards crafting text adversarial samples," 2017. [Online]. Available: arXiv:1707.02812.

[45] L. Schönherr, K. Kohls, S. Zeiler, T. Holz, and D. Kolossa, "Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding," 2018. [Online]. Available: arXiv:1808.05665.

[46] M. Soll, T. Hinz, S. Magg, and S. Wermter, "Evaluating defensive distillation for defending text processing neural networks against adversarial examples," in *Proc. Int. Conf. Artif. Neural Netw.*, 2019, pp. 685–696.

[47] J. Szurley and J. Z. Kolter, "Perceptual based adversarial audio attacks," 2019. [Online]. Available: arXiv:1906.06355.

[48] R. Taori, A. Kamsetty, B. Chu, and N. Vemuri, "Targeted adversarial examples for black box audio systems," 2018. [Online]. Available: arXiv:1805.07820.

[49] Y. Vaizman, K. Ellis, and G. Lanckriet, "Recognizing detailed human context in the wild from smartphones and smartwatches," *IEEE Pervasive Comput.*, vol. 16, no. 4, pp. 62–74, Oct.–Dec. 2017.

[50] P. Vijayaraghavan and D. Roy, "Generating black-box adversarial examples for text classifiers using a deep reinforced model," 2019. [Online]. Available: arXiv:1909.07873.

[51] B. Wang, H. Pei, H. Liu, and B. Li, "AdvCodec: Towards a unified framework for adversarial text generation," 2019. [Online]. Available: arXiv:1912.10375.

[52] W. Wang, B. Tang, R. Wang, L. Wang, and A. Ye, "A survey on adversarial attacks and defenses in text," 2019. [Online]. Available: arXiv:1902.07285.

[53] X. Wang, H. Jin, and K. He, "Natural language adversarial attacks and defenses in word level," 2019. [Online]. Available: arXiv:1909.06723.

[54] Y. Wu, J. Liu, Y. Chen, and J. Cheng, "Semi-black-box attacks against speech recognition systems using adversarial samples," in *Proc. IEEE DySPAN*, 2019, pp. 1–5.

[55] L. Xie, Y. Mo, and B. Sinopoli, "False data injection attacks in electricity markets," in *Proc. IEEE SmartGridComm*, 2010, pp. 128–138.

[56] L. Xie, Y. Mo, and B. Sinopoli, "Integrity data attacks in power market operations," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 659–666, Dec. 2011.

[57] M. Yadegar, N. Meskin, and W. M. Haddad, "An output-feedback adaptive control architecture for mitigating actuator attacks in cyber-physical systems," *Int. J. Adapt. Control Signal Process.*, vol. 33, no. 6, pp. 943–955, 2019.

[58] H. Yakura and J. Sakuma, "Robust audio adversarial example for a physical attack," 2018. [Online]. Available: arXiv:1810.11793.

[59] W. Yan, L. K. Mestha, and M. Abbaszadeh, "Attack detection for securing cyber physical systems," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8471–8481, Oct. 2019.

[60] Q. Yang, J. Yang, W. Yu, D. An, N. Zhang, and W. Zhao, "On false data-injection attacks against power system state estimation: Modeling and countermeasures," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 3, pp. 717–729, Mar. 2014.

[61] Z. Yang, B. Li, P.-Y. Chen, and D. Song, "Towards mitigating audio adversarial perturbations," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–7.

[62] D. Ye and T.-Y. Zhang, "Summation detector for false data-injection attack in cyber-physical systems," *IEEE Trans. Cybern.*, early access, doi: 10.1109/TCYB.2019.2915124.

[63] Z.-H. Yu and W.-L. Chin, "Blind false data injection attack using PCA approximation method in smart grid," *IEEE Trans. Smart Grid*, vol. 6, no. 3, pp. 1219–1226, May 2015.

[64] X. Yuan *et al.*, "CommanderSong: A systematic approach for practical adversarial voice recognition," in *Proc. USENIX Security*, 2018, pp. 49–64.

[65] X. Yuan, P. He, and X. A. Li, "Adaptive adversarial attack on scene text recognition," 2018. [Online]. Available: arXiv:1807.03326.

[66] Q. Zeng *et al.*, "A multiversion programming inspired approach to detecting audio adversarial examples," in *Proc. IEEE/IFIP DSN*, 2019, pp. 39–51.

[67] J. Zhang, B. Zhang, and B. Zhang, "Defending adversarial attacks on cloud-aided automatic speech recognition systems," in *Proc. Security Cloud Comput.*, 2019, pp. 23–31.

[68] M. Zhou, Z. Qin, X. Lin, S. Hu, Q. Wang, and K. Ren, "Hidden voice commands: Attacks and defenses on the VCS of autonomous driving cars," *IEEE Wireless Commun.*, vol. 26, no. 5, pp. 128–133, Oct. 2019.

[69] Y. Zhou, K. G. Vamvoudakis, W. M. Haddad, and Z.-P. Jiang, "A secure control learning framework for cyber-physical systems under sensor attacks," in *Proc. Amer. Control Conf.*, 2019, pp. 4280–4285.

[70] Y. Zhu and S. S. Woo, "Adversarial product review generation with word replacements," in *Proc. ACM Conf. Comput. Commun. Security*, 2018, pp. 2324–2326.

Jiao Li (Student Member, IEEE) received the B.E. degree from the School of Software Engineering, Xidian University, Xi'an, China, in 2018. She is currently pursuing the Ph.D. degree with the Department of Computer Science, City University of Hong Kong, Hong Kong.

Her current research interests include cyber security, mobile computing, and deep learning.

Yang Liu (Student Member, IEEE) received the B.E. degree from Xi'an Jiaotong University, Xi'an, China, in 2016. She is currently pursuing the Ph.D. degree with the Department of Computer Science, City University of Hong Kong, Hong Kong.

Her research interests include mobile and wearable sensing, mobile computing, deep learning, and cyber security and privacy.

Tao Chen (Student Member, IEEE) received the B.E. degree from Northwestern Polytechnical University, Xi'an, China, in 2017. He is currently pursuing the Ph.D. degree working with Dr. Z. Li with the Department of Computer Science, City University of Hong Kong, Hong Kong.

His research interests include wireless and machine learning, signal processing, and cyber security.

Zhen Xiao (Student Member, IEEE) received the B.E. degree from the Department of Electrical Engineering, City University of Hong Kong, Hong Kong, in 2018, where he is currently pursuing the Ph.D. degree with the Department of Computer Science.

His current research interests include smart sensing and deep learning.

Zhenjiang Li (Member, IEEE) received the B.E. degree from Xi'an Jiaotong University, Xi'an, China, in 2007, and the M.Phil. and Ph.D. degrees from the Hong Kong University of Science and Technology, Hong Kong, in 2009 and 2012, respectively.

He is currently an Assistant Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong. His research interests include wearable and mobile computing, smart health, deep learning, and distributed computing.

Jianping Wang (Senior Member, IEEE) received the B.S. and M.S. degrees in computer science from Nankai University, Tianjin, China, in 1996 and 1999, respectively, and the Ph.D. degree in computer science from the University of Texas at Dallas, Richardson, TX, USA, in 2003.

She is a Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong. Her research interests include cloud computing, edge computing, autonomous driving, Internet of Things, and security.