DeSVig: Decentralized Swift Vigilance Against Adversarial Attacks in Industrial Artificial Intelligence Systems

DeSVig: Decentralized Swift Vigilance Against Adversarial Attacks in Industrial Artificial Intelligence Systems

Gaolei Li, *Student Member, IEEE,* Kaoru Ota, *Member, IEEE,* Mianxiong Dong, *Member, IEEE,* Jun Wu, *Member, IEEE,* and Jianhua Li

***Abstract*—Individually reinforcing the robustness of a single deep learning model only gives limited security guarantees especially when facing adversarial examples. In this paper, we propose DeSVig, a Decentralized Swift Vigilance framework to identify adversarial attacks in an industrial artificial intelligence systems (IAISs), which enables IAISs to correct the mistake in a few seconds. DeSVig is highly decentralized, which improves the effectiveness of recognizing abnormal inputs. We try to overcome the challenges on ultra-low latency caused by dynamics in industries using peculiarly-designated mobile edge computing and generative adversarial networks (GANs). The most important advantage of our work is that it can significantly reduce the failure risks of being deceived by adversarial examples, which is critical for safety-prioritized and delay-sensitive environments. In our experiments, adversarial examples of industrial electronic components are generated by several classical attacking models. Experimental results demonstrate the DeSVig is more robust, efficient, and scalable than some state-of-art defences.**

***Index Terms*—Deep learning, adversarial examples, industrial artificial intelligence systems (IAISs), mobile edge computing, generative adversarial networks (GAN).**

I. Introduction

INDUSTRIAL artificial intelligence systems (IAISs) have great potentials to complete some complex industrial tasks, replacing a large variety of human vigour in many industrial scenarios. As one of the key enabling technologies in IAISs, deep learning with the superiorities of standardizable structure and high accuracy has achieved increasing attention in recent years. Significant performance of deep learning attracts a lot of researchers from different fields and then such interdisciplinary cooperation greatly expands the application scope of deep learning [1, 2, 3]. For example, the power of deep learning in signal compression and signal detection revolutionizes the physical layer communications [4]. We consider that the industrial artificial intelligent system is a promising application scenario, which supports edge computing, deep learning, and 5G networks.

Since many industrial systems are delay-sensitive and confidential, deep learning must process industrial data locally, leading to the bottleneck in terms of local resource exhaustion. At present, in order to overcome such difficulties, both academic and industrial circles are strongly advocating mobile edge computing, which enables context-aware resources offloading [5, 6]. With sufficient computation resources, deploying the deep learning models at edges in a distributed way has been a popular trend in recent years [7, 8, 9].

One common prospect of deep learning in a smart industry what many researchers want to achieve is "Deep Learning of Things (DLoT)", a term we define as a special case of on-edge/device deep learning [10, 11, 12]. Without a good on-edge/device deep learning model, where following the model does not correctively and timely output the corresponding result, smart industry can not succeed thoroughly. Although it is not a difficulty for industrial engineers to train deep learning with high accuracy, many industrial vendors are still on the sidelines and few deep learning models are adopted in real production lines, due to potential security vulnerabilities.

Adversarial example [13], which is specially-designed with iterative optimization, gradually growing into the greatest threat to deep learning. Different from common hacking attacks in computer networks, the goal of generating an adversarial example is to fool the deep learning model rather than to steal some confidential data. Many machine learning and deep learning models have been validated to be susceptible to adversarial examples. There are many interests in constructing defences to enhance the robustness of deep learning, while new powerful optimization-based attacks are also arising [14, 15].

Together with deep learning, the adversarial example is introduced into industrial scenarios, possibly leading to serious accidents or economic losses [16]. Attackers can achieve great attacking effects in industries only by feeding several adversarial examples into the deep learning model with ultra-low frequency [17]. For example, in the semiconductor industry, vendors can use a deep learning model to identify the type of electronic devices (such as capacitor, inductor, and transistor) on circuit boards. A mistake that identifies a capacitor as an inductor may cause important chips to burn out. Fig. 1 shows possible attacks that may feed adversarial examples into IAISs.

To promote the practicality of various deep learning products in the upcoming smart world, exploring the effective defence methods to circumvent adversarial examples is an indispensable and urgent need. However, it is not an easy task to evade the impact of adversarial examples. Moreover, the variability of industrial scenes makes such evasion extremely difficult. We identify three difficulties of evading adversarial examples in industrial scenes:

G. Li is with the School of Cyber Security, Shanghai Jiao Tong University and Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, Shanghai 200240, China. And also, he is a visiting student at Muroran Institution of Technology, Muroran 050-8585, Japan.

Kaoru Ota and Mianxiong Dong are with Muroran Institute of Technology, Muroran 050-8585, Japan. (Corresponding author: Mianxiong Dong, Email: mx.dong@csse.muroran-it.ac.jp)

Jun Wu and Jianhua Li are with School of Cyber Security, Shanghai Jiao Tong University, Shanghai, China, 200240 and the Shanghai Key Laboratory of Integrated Adminstration Techologies for Information Security.

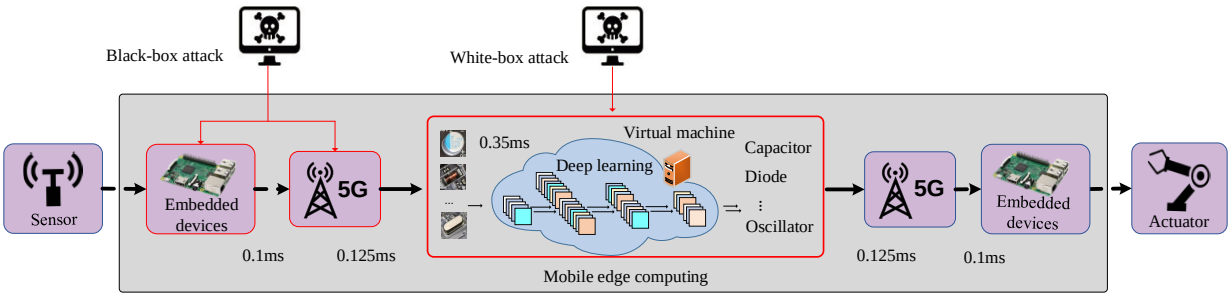


Fig. 1. Possible attacks that feed adversarial examples into industrial artificial intelligence systems.

- *Irregular interferences* are unexpected noises caused either intentionally by privacy protection or unintentionally by situational changes. Coincidentally, some interferences may transfer real inputs into abnormal inputs, which has the same function with adversarial examples.
- *Shattered feeding* is a highly discrete and distributed attack trick, which is effective for attackers to hide their identities during the attacking stage. Common defences are not sensitive to such weak signaling.
- *Ultra-short reaction time* is maximum latency (less than 5ms) of real-time industrial applications. For 5G-enabled Tactile Internet [18, 19], this latency is in range of 1ms, leading to the inapplicability of complex defences.

In this paper, we propose a Decentralized Swift Vigilance (DeSVig) framework, which can circumvent the unknown adversarial examples in industrial artificial intelligence systems (IAISs). We consider the DeSVig framework with several deep learning models and more than one conditional generative adversarial network (CGAN). Each deep learning models sends its inputs and outputs to CGANs for asking vigilance service. Each CGAN calculates the corresponding vigilance decisions rapidly and distributes the vigilance decisions on demands. During the defence process, CGANs work at the control plane and deep learning models at the data plane. The failure risks of being deceived by adversarial examples can be detected with no requirement on complex robustness reinforcement to original deep learning model, which is critical for safety-prioritized, delay-sensitive industrial environments. Moreover, decoupling the control plane from the data plane reduces the vigilance latency significantly. DeSVig is of great advantage for industrial robots to correct the mistake in a few milliseconds. Contributions of our work are mainly summarized as the following three aspects.

- We formulate a mathematical description to identify the relationship CGANs and adversarial examples. And then, we investigate the applicability of existing defence models. Different from existing studies, we deduce a novel defence decision-making (DDM) function for IAISs to swiftly detect adversarial examples from a large number of data inputs, which fulfills the delay and privacy protection requirements of industrial environments.
- We propose the decentralized swift vigilance (DeSVig) framework, which decouples defence decision-making (control plane) from deep learning models (data plane). In the control plane, we exploit CGAN to generate an input copy and mobile edge computing agent to execute DDM function. We provide vigilance acceleration methods based on privacy-aware incentive mechanism and energy-efficient distributed consensus mechanism.
- We propose a flexible implementation prototype of DeSVig framework, which firstly extends the concept of OpenFlow [20] into "OpenExample". OpenExample is defined as a communication protocol between DeSVig controller and deep learning models. To demonstrate DeSVig's superiority on hitting rate and response latency, both open dataset (MNIST) and real industrial dataset are used in our experiments.

The rest of this article is organized as follows. Section II gives the related work and specifies the strength of work against existing studies. Section III introduces the system of DeSVig. Both main components and workflow of DeSVig are introduced in this section. Experimental evaluation is provided in Section IV. Section V concludes this paper.

II. Related Work

The susceptibility of deep learning to adversarial example is usually described as a quasi-chaos phenomenon, where a tiny but meticulously worst-case perturbation on inputs will lead to abnormal outputs with high confidence [30]. The adversarial example is an "obstinate disease" that can survive in the physical world, and can bypass the state of the art defence technologies with simple reinforcement. The study on circumventing adversarial examples in industrial scenarios is in an initial and rising stage.

As surveyed in [21], there have been 15 types of defence models to adversarial examples in total. Among these defence models, inputs transferring has low additional cost because it does not directly modify the original neurons or add new parameters on the neurons. The state-of-the-art inputs transferring-based defence models include data compression [22], wavelet-based noising [23], and Saak Transform [24]. However, inputs transferring may reduce the accuracy of normal classification because normal inputs also need to be processed by the above inputs transferring technologies. Considering a valid adversarial example is generated by iteratively optimizing the loss function with a proper gradient, many existing defence methods provide apparent robustness mainly relying on obfuscated gradients. However, most of these defence methods mainly rely on the obfuscated gradients

but this kind of defence methods has been validated to give a false sense of security [15].

The state of the art defences have two branches: 1) certified defence, and 2) GAN-based defence. The certified defence is often implemented via semi-definite relaxation [31] or differentiate privacy [32] Although this kind of defences achieves some promotions, it is not very scalable to high-dimensional datasets because it is a complex task to evaluate a reasonable noise. The GAN can learn data distribution of real inputs and has been applied in many fields like wireless communication [33]. The GAN-based defence often trains a generative model to learn the distribution of unperturbed images. By leveraging the expressive capability of generative models, a defence-GAN framework is proposed to defend deep neural networks against adversarial attacks [41]. The biggest advantage of defence-GAN is that it can be used without any modification on the classifier structure and its training procedure. However, this method require the used GAN to gain some prior knowledge about the attackers. To make the defence independent of the prior knowledge, study [42] proposes MimicGAN, which formulates the adversarial defence as a general inverse problem and address it with a unsupervised technique. However, since this kind of defence requires to train many generative models for different datasets, too many computational resources will be consumed. Up to now, there is no apposite solution that can completely eliminate this vulnerability of AI to adversarial examples with low costs.

In order to support the secure and fast control requirements of industrial artificial intelligent systems, integrating mobile edge computing (MEC) and 5G networks to achieve tactile platform has been perceived as a promising trend [25, 26]. Industrial operators can use MEC servers to train or run these complex learning tasks, while the 5G networks can be utilized to deliver the model parameters or aggregate the datasets. In particular, software-defined networking (SDN) is also adapted to enhance the manageability of industrial control networks [27, 28, 29]. The SDN uses a controller to decide the packet's fate. The decouple between routing algorithms and forwarding actions inspire us. The application of these emerging technologies in industrial environments brings defence up to par with offence, which will change everything in future industrial systems. Simultaneously, it provides soil for the breeding of new attacks and provides a new direction for defences.

Different from existing defence methods that blindly pursue high robustness, we propose a novel defence framework named as DeSVig to evade the unknown adversarial examples from a large number of inputs in real IAISs. The most important advantage of our work is that the proposed scheme can significantly reduce the failure risks of being deceived by adversarial examples with no requirement on complex robustness reinforcement, which is critical for safety-prioritized, delay-sensitive environments. The performance of DeSVig is independent of the trained model parameters. We identify the reason why adversarial examples can success is that the training process of deep learning has irreversibility but each label is of polysemy. The significant novelties of our work include: 1) we propose to use a CGAN, which learns the distribution law of given dataset, to generate a copy of input; 2) we train a discriminator to compare the distribution law of original input and generated copy; 3) we create OpenExample protocol to decouple defence decision-making from deep learning models. The proposed framework can fit any networks which enables artificial intelligence and edge computing.

III. System Model of DeSVig

We consider the system model of DeSVig with the control plane, data plane and an OpenExample protocol. In the data plane, there are N deep learning (DL) models with network brakes and each DL model serves for several users. In the control plane, there is a controller that consists of one mobile edge computing agent, one CGAN and one discriminator. The control plane is separated and hidden.

TABLE I
The Key Symbols and Explains

Symbols	Explains
x_i	The original input
δ_i	The small perturbation added into original input x_i
a_i	The adversarial example created based on x_i
$g(x_i, \delta_i)$	The attacking function to generate a_i
y_i	The real label of original input x_i
y'_i	The mistake label of adversarial a_i
$X^n_{t_i}$	The original input feed by DL n at time t_i
$X'^n_{t_i}$	The input copy generated by CGAN
$P(x_i\|y_i)$	The possibility of mapping x_i into y_i
$P(a_i\|y_i)$	The possibility of mapping a_i into y_i
$\beta^n_{t_m}$	The attacker's intent to DL n at t_m
G	The generator of CGAN
D	The discriminator of CGAN
$V(D,G)$	The value function of CGAN
θ	The parameter of a neural network
z	A random noise function in generator of CGAN
e	The loss function of a deep learning model
$F(.)$	The mapping function of a deep learning model
$H(.)$	The mapping function of a CGAN

A. Attacking Patterns

In this paper, we focus on the emerging attacking patterns caused by introducing deep learning into industrial scenarios. Different from traditional attacking patterns, industrial adversarial examples have many significant features such as irregular interference, shattered feeding, and ultra-short reaction.

We use δ_i to denote the perturbation on input x_i and $a_i = g(x_i, \delta_i)$ to denote adversarial example, where $g(\cdot, \cdot)$ is one kind of generation function of adversarial examples. An adversarial example can be described with a brief mathematical primitive [25]. There exists a a_i that satisfies the following two properties: 1) the euclidean metric of x_i and a_i is small enough, $lim||a_i - x_i|| = 0$; and 2) the possibility $P(x_i|y_i)$ of mapping image x_i as real category label y_i is far greater than the possibility $P(a_i|y_i)$ of mapping a_i into y_i, $P(x_i|y_i) \neq P(a_i|y_i)$. Adversarial example has been brought into the physical world, where the printed adversarial examples can deceive machine learning-based applications on cellphone camera [34, 35]. For a given input image x_i, the corresponding adversarial example is generated using iterative

optimization, which attempts to gain a minimum perturbation δ that make $||a_i^{\delta_i} - x_i|| \approx 0$ as well as $P(a_i|y_i) \neq P(x_i|y_i)$. Many methods have been developed to efficiently generate adversarial examples. In this article, we will use the IBM's Adversarial Robustness Toolbox (ART) [36], which provides an implementation for 9 attacking methods.

In industrial scenarios, several adversarial examples mixed in a large number of inputs may appear suddenly. For a smart plant with N workshop and each workshop has one DL model, the normal inputs of DeSVig observed during a period time can be formulated as the product of a $M \times N$ binary matrix and an input matrix:

$$X = \begin{pmatrix} \beta_1^{t_1} & \cdots & \beta_N^{t_1} \\ \vdots & \ddots & \vdots \\ \beta_1^{t_M} & \cdots & \beta_N^{t_M} \end{pmatrix} \begin{pmatrix} x_1 \\ \cdots \\ x_N \end{pmatrix} \tag{1}$$

where $\beta_{t_m}^n \in \{0, 1\}$. Therefore, the abnormal inputs with several adversarial examples are denoted as A:

$$A = X + \delta = \begin{pmatrix} \beta_1^{t_1} & \cdots & \beta_N^{t_1} \\ \vdots & \ddots & \vdots \\ \beta_1^{t_M} & \cdots & \beta_N^{t_M} \end{pmatrix} \begin{pmatrix} x_1 \\ \cdots \\ x_N \end{pmatrix} + \begin{pmatrix} \delta_1 \\ \cdots \\ \delta_N \end{pmatrix} \tag{2}$$

Usually, to pick out adversarial examples from a large number of inputs, we must know what/when/where is the attack, which is very hard in industrial scenarios.

B. Basic Components

To circumvent adversarial examples in industrial scenarios, we firstly propose to decouple defence control strategies from data processing of deep learning. DeSVig contains conditional generative adversarial networks (CGAN), mobile edge computing agent, untrusted deep learning and perceptual hashing discriminator. Fig. 2 shows the basic components and connection matrix of DeSViG.

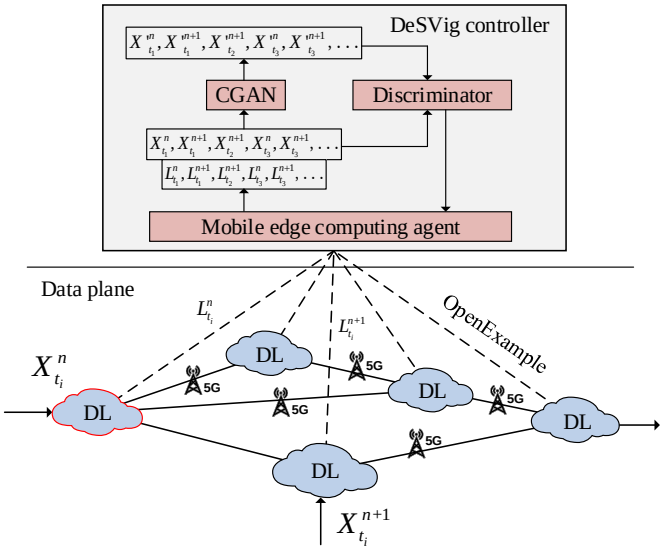


Fig. 2. Basic components in DeSVig.

1) Conditional Generative Adversarial Network: Conditional generative adversarial network (CGAN) is extended from the Generative adversarial network (GAN), which is a novel way of training generative model. It contains two "adversarial models": 1) Generative model G for learning data distribution, and 2) Discriminant model D for estimating the probability of an input coming from training dataset instead of G. Both G and D are multi-layer neural networks. In G, a nonlinear function $G(z; \theta_g)$ that maps noise distribution $p_z(z)$ into data space is defined for G to learn the distribution of training data x. The output of D is a percentage, which is denoted by $D(x; \theta_d)$ represents the probability of an input coming from training dataset instead of G. The optimization problem of a GAN can be formulated as a value function $V(G, D)$ base on a two-player min-max game.

$$\begin{aligned} \min_G \max_D V(D,G) =& E_{x\sim p_d(x)}[logD(x)]+ \\ & E_{z\sim p_z(z)}[log(1-D(G(z)))] \end{aligned} \tag{3}$$

Adding some specified conditions into G and D respectively can control their outputs. The value function of CGAN can be denoted as:

$$\begin{aligned} \min_G \max_{D|y} V(D,G) =& E_{x\sim p_d(x)}[logD(x|y)]+ \\ & E_{z\sim p_z(z)}[log(1-D(G(z|y)))] \end{aligned} \tag{4}$$

where y represents the specified condition. In this paper, we use CGAN to generate copy images of inputs in real-time. During the generation process, the category labels received from distributed DL models in data plane are specified as CGAN's conditions.

2) Mobile Edge Computing Agent: Mobile edge computing (MEC) agent is designed to improve the throughput capacity of DeSVig controller, which serves for multiple DL models. Usually, the MEC agent has two main functions: 1) queuing inspection requests, and 2) distributing defence decisions. As illustrated in Fig.1, the defence decisions must be generated with less than $0.35ms$. Otherwise, the DL model's output will be sent out. For a large scale industrial network, where there are several DeSVig controllers, MEC agent is also responsible to build communication among DeSVig controllers. This case will be introduced in Section V in detail.

3) Untrusted Deep Learning: The deep learning model trained for multi-label classification is often formulated as iterative optimization problem. For given inputs $x = (x_1, x_2, ..., x_k)$, the cross-entropy loss function is often defined as:

$$e = crossEntropy(s, y) = \sum_{i=1}^{k} y_i log(s_i) \tag{5}$$

where $s_i = \frac{e^{x_i}}{\sum_{t=1}^{k} e^{x_t}}$. Therein, k represents the number of inputs and y denotes the real category labels. A accurate deep learning model minimizes the above cross-entropy loss function e by calculating its gradient values. Under attack environment, the attackers may add some perturbations into the inputs to induce the direction of gradient descent.

The framework of DeSVig is hierarchical. All the deep learning models are working in the data plane of DeSVig, while CGAN, perceptual hashing discriminator, and mobile

edge computing agent are working in the control plane. Different deep learning models in the data plane remain independent and also have a certain association.

One one hand, since forwarding all data inputs to a data center is realistic obviously, the deep learning models are deployed in every corner of the future industrial systems in a distributed way. Considering a real scenario where a large industrial company has many factories around the world, each factory trains some different deep learning models based on different computational intelligence to classify the electronic components. Under such application scenario, the sensed images may be randomly feed into some of these heterogeneous deep learning models for classification. Therefore, different deep learning models are working independently. On the other hand, to schedule the resources among different factories, we need to monitor and synchronize the running statuses of different learning learning models. Therefore, the deep learning models are associated with each other through some efficient networks (i.e., 5G).

The adversarial examples can attack every node in the data plane of DeSVig. Obviously, it is inefficient and inelastic to deploy defences on every node especially when we want to configure new defences. The DeSVig creates a control plane for efficiently detecting adversarial examples and swiftly assigning vigilance among different deep learning models. This scheme allows the operators to add whatever they want into the DeSVig controller to update their defence strategies or configure new defences.

4) Perceptual Hashing Discriminator: In Fig. 2, the DeSVig controller contains a specially-designed discriminator, which is named as perceptual hashing discriminator. This special discriminator is different from the CGAN's discriminator. We summarize three key aspects to clarify the difference between the DeSVig's discriminator and the CGAN's discriminator as follows.

- Structural difference. The DeSVig's discriminator is designed based on the perceptual hashing [37] rather than the neuron network. The DeSVig's discriminator has two different inputs 1) the original images sent from the distributed DL models, and 2) the copy image generated by the CGAN. The DeSVig's discriminator is decoupled with the generative model and does not need any training procedure.
- Functional difference. The CGAN's discriminator is an assistant component, which works during the training procedure. The main function of CGAN's discriminator is to monitor if the generated sample is real. Different from the CGAN's discriminator, perceptual hashing enables fast and flexible similarity detection of multimedia data by calculating the Hamming Distance between the different inputs. The main function of DeSVig's discriminator is to monitor if the input is an adversarial example.

C. Defence Principles

In this section, we will introduce the defence principles of DeSVig in detail by clearly clarifying the relationship between basic components. It is common to see that there is an adversarial example a_i generated from x_i in a group of inputs $x = (x_1, x_2, ..., x_k)$. If a_i is feed into DL model n, the corresponding outputs $y = (y_1, y_2, ..., y_k)$ of DL model n will include a mistake label y_i'.

To explain it more intuitively, we follow the definition for data labels illustrated in Fig. 2. In real industrial scenarios, the inputs and outputs of DL models are denoted as $X_{t_i}^n$ and $L_{t_i}^n$ respectively, where t_i represents the arriving time of input X and n represents the serial number of DL model. Adversarial examples in this real scenario is denoted as $A_{t_i}^n\ L_{t_i}$, while the corresponding mistake label is denoted as $L_{t_i}^{'n}$.

Creatively, we propose to feed the identified labels into CGAN to generate a copy of each input. Given an accurate CGAN, the generated copy $X_{t_i}^{'n}$ is obviously similar to $X_{t_i}^n$ in Hamming space if the label is corrective. The mathematical formulation is listed as follows:

$$X_{t_i}^n \overset{hm}{=} H(F(X_{t_i}^n)) \tag{6}$$

where $F(.)$ is the function of a DL model and $H(.)$ is the function of CGAN. Additionally, hm denotes the Hamming distance between $X_{t_i}^n$ and $X_{t_i}^n$. Here, we focus on the workflow explanation of DeSVig. Table II and Table III respectively give the detail workflows in data plane and control plane of DeSVig.

1) Privacy-preserving in DeSVig Data Plane: Data plane of DeSVig consists of various DL models in a big industrial plant, which may directly receive requests from various end-users (EU). Data plane provides low-latency services for distributed EU and must protect the EU's privacy that is highly essential in industrial scenarios.

TABLE II

Principle 1: Privacy-Aware decentralized condition generation algorithm in DeSVig data plane.

Input: X, EU, R_Noise
Output: L, X_γ
1: Identify the recognition request and the ID of source EU
2: Iteratively optimize equation (5) and give the final value of $F(X)$
3: Mapping $F(X)$ into label L
4: **for** $label$ in RT
5: **if** $label$==L
6: **for** $label'$ in FT
7: **if** $label'$ == L
8: go to step 6
9: **else**
10: Add R_Noise into X: X_γ=X+R_Noise
11: Send X_γ and L to control plane
12: Update the FT with L
13: **end**
14: **else**
15: Add R_Noise into X: X_γ=X+R_Noise
16: Send X_γ and L to control plane
17: Update the FT with L
18: **end**
19: **end**

The privacy-aware decentralized condition generation algorithm illustrated by Table II exploits the method of local differential privacy [39] to hide the real values of the original inputs.

TABLE III

Principle 2: Decision-making process in DeSVig controller against multi-attacks.

Input: $X_\gamma = \{x_\gamma^1, x_\gamma^2, ..., x_\gamma^k\}$, $L_\gamma = \{l_\gamma^1, l_\gamma^2, ..., l_\gamma^k\}$
Output: S
1: Monitor the workload W of DeSVig controller
2: **if** $W < W_0$
3: Receive X_γ and L_γ
4: Identify the type of X_γ and L_γ
5: Select appropriate CGAN for each label L
6: Iteratively optimize equation (4) to generate input copy X'_γ
7: Send X_γ and X'_γ to perceptual hashing discriminator
8: **else**
9: Forward X_γ and L_γ to neighboring DeSVig controller
10: **end**
11: Resize X_γ and X'_γ as 8×8 matrix
12: Simplify the color into 64 levels
13: Calculate the mean of all pixels of each image
14: Binarization by comparing each pixel with the mean
15: Compose binary values as integer I_γ and I'_γ with 64 bits
16: Calculate hash value $H_1 = (I_\gamma)$ and $H_2 = (I'_\gamma)$
17: **if** H_1==H_2
18: Normal inputs, then set S=true
19: **else**
20: Adversarial example, and set S=false
21: Trace the sources of adversarial examples
22: **end**

Symbols used in Table II are denoted as follows. X denotes the raw inputs, while X_γ denotes the privacy-preserving inputs. L represents the real category label. R_Noise denotes the noises that will be added into X. At the beginning stage, DeSVig controller believes the outputs (L) of distributed DL models are corrective. The DeSVig controller can be maintained by the third-party organization because it cannot restore X with given X_γ.

This algorithm is of two great advantages: 1) Privacy-aware; 2) low latency. In step 10 and step 15, each device with a DL model needs to add some noises R_Noise into original data to make them fulfil a certain distribution. Step 5 and Step 7 remove some redundant condition generation, which will reduce DeSVig controller 's cost and accelerate its response. Additionally, RT and FT are two different tables maintained by individual DL model. Authenticated labels are logged in RT and pending labels are recorded in FT.

2) Decision-making of DeSVig Controller: In this section, we will introduce the decision-making process of DeSVig controller step by step. Decision-making process of DeSVig controller mainly consists of three modules: 1) load balance on MEC agent, 2) input copy generation on CGAN, and 3) Similarity calculation on discriminator. The detail decision-making process is shown step by step in Table III. Additionally, the symbols used in Table III are denoted as follows. X_γ and L_γ denote a pair of inputs and labels. W represents the workload of DeSVig controller. X'_γ denotes the generated input copy. H_1 and H_2 are the hashing values. S is an indicator. $S = false$ means that the input is an adversarial example.

Load balance on MEC agent: In a big industrial network, there are many DeSVig controllers. As mentioned above, DeSVig controller is responsible to swiftly detect adversarial examples from a large number of inputs. To generate trusted vigilances as soon as possible, the MEC agent is designated to provide load balance services among DeSVig controllers.

Input copy generation on CGAN: Initially, MEC agent receives data and their labels from distributed DL models. Subsequently, the CGAN generates an input copy X' according to the condition L and sends it to perceptual hashing discriminator together with X_γ. Notably, the D and G of CGAN must be trained with the same dataset to the corresponding DL model. If the DL models are heterogeneous, a DeSVig controller should smartly schedule the work of multiple CGANs.

Similarity calculation on discriminator: We propose to use a perceptual hashing algorithm to calculate the Hamming distance between X' and X_γ. By iteratively optimizing, DeSVig controller learns a flexible threshold of Hamming distance to identify the similarity between two images. Similarity calculation on discriminator is scalable to detect multiple kinds of attacks.

3) OpenExample Protocol: The OpenExample protocol starts from the reference of a typical industrial artificial intelligence system. Such a network has two major components: 1) deep learning nodes, and 2) industrial control networks. First, the deep learning node which receives inputs on input port, decide the input's fate by consulting some control strategies, and then enacts the executable actions. Second, the industrial control networks perform more high-level functions such as running dynamic routing protocols (BGP, OSPF, STP, etc), supporting a resource management interface (SNMP, CLI, SOAP, etc), and broadcasting the deep learning nodes' parameters as necessary. The industrial control networks can be established based on a typical ethernet and then generally upgrades as 5G networks, including software-defined networks. OpenExample protocol decouples the model control from model computing. Architectural components of OpenExample are shown as illustrated in Fig. 3.

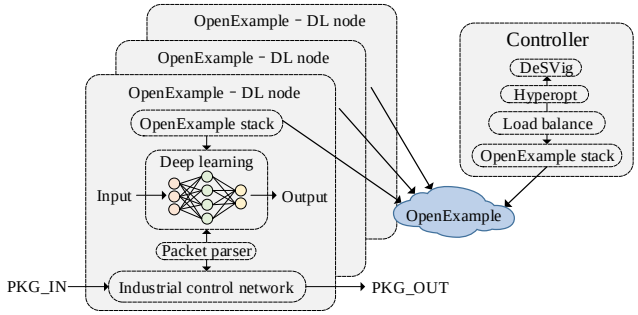


Fig. 3. Architectural components of OpenExample.

IV. Evaluation and Discussion

In this section, we introduce the performance evaluation of DeSVig on detecting adversarial examples. Usually, adversarial example generated by white-box attacking is more difficult to detect than that generated by black-box attacking because white-box attacking can observe the parameters of pre-trained DL models and ultimately provide more robust adversarial

TABLE IV
THE ACCURACY(%), COST AND DELAY OF DIFFERENT DEFENCES UNDER ATTACK SCENARIOS (BEST RESULT)

	Baseline	FGSM	Rand FGSM	DeepFool	Cost	Delay
Without defence	98.9	90.29	40.70	49.02	+0	0.06s
Feature squeezing	98.9	96.31	91.09	96.68	+2	0.12s
Adversarial training	98.9	91.86	49.77	85.91	+5	0.07s
Differential Privacy	**98.9**	**99.2**	**90.52**	**94.24**	**+1.5**	**0.08s**
GDA + RELU	98.9	98.47	80.25	97.84	+1	0.08s
DeSVig	**99.9 (+1)**	**98.29 (+8)**	**97.9 (+57.20)**	**96.02 (+47)**	**+0.1**	**0.062s**

examples. Additionally, detecting high dimensional adversarial examples (such as color images) will consume more resources.

Considering there are few open industrial datasets for adversarial examples, direct comparisons with existing defence methods are very hard. To fairly evaluate the performance of DeSVig, our experiments are implemented on two different datasets: 1) MNIST dataset; 2) self-made industrial dataset. The simulation results relying on MNIST datasets is to demonstrate the strengths of DeSVig against exiting defence methods, while simulation results relying on the self-made industrial dataset is to highlight the scalability, applicability, and rationality of DeSVig in industrial scenarios.

For easy maintenance, we develop the corresponding agent for each pre-trained model. The underlying communication stack for these agents is implemented with WIFI and the communication module is developed by $JSON$ tools. Each agent is assigned with one specific url. The CGAN models and the DL models are pre-trained on a computer with $Inteli7-4770CPU$ and then deployed on the Raspberry 3. The required hardware is affordable for many institutes. Some specific experimental setting are described according to the dataset.

A. Performance Validation based on MNIST

We deploy four DL models that are pre-trained based on MNIST dataset. DL_A and DL_B are disconnected with DeSVig; DL_C and DL_D are connected with DeSVig. DL_A and DL_C are feed with 100 normal images, respectively; DL_B and DL_D are feed with 100 adversarial examples, respectively. The selected attacking methods include FGSM [30], rand FGSM and DeepFool [38].

The used CGAN is also trained based on MNIST dataset. The function of CGAN is to generate an image according to the specified label. For example, given a label "3", CGAN will return an image written with "3". Communication between DeSVig controller and distributed DL models is formed by the local area network (LAN) and OpenExample protocol, which is defined to formulate the formats of exchanged data. Perception hashing discriminator exchanges data with CGAN relies on internal process iteration.

1) Certified Accuracy: To bring out the advantages of DeSVig, we measure the accuracy of different defences under several typical attacks. Simulation results show that adversarial examples both DeepFool and FGSM can be accurately detected by DeSVig. The detail comparisons are listed in Table IV.

Certified Accuracy is the most important inductor that decides the availability of a DL model. Due to the possibility theory base of deep learning, DL models with strong defences still can not achieve 100% accuracy. The DeSVig is more willing to discover the mistake of DL models but not improve the DL models' robustness. With the usage of DeSVig, almost all of the abnormal inputs will be evaded from the input queues. Therefore, the baseline accuracy is further improved with 1%.

For adversarial examples generated by FGSM, Rand FGSM, and DeepFool, DeSVig outperforms the same generalization to the state of art defence method differential privacy. Besides, defence effects of DeSVig also can be certified because we can adjust the threshold of hamming distance. Fig. 4 demonstrates the relationship between certified accuracy and threshold.

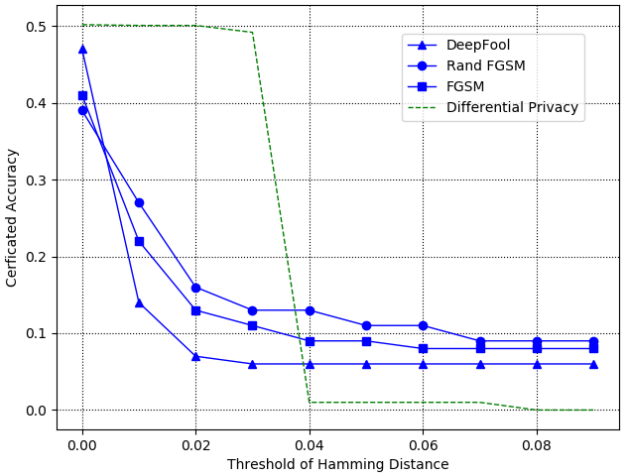


Fig. 4. Certificated accuracy on MNIST.

B. Application in Self-made Industrial Dataset

Except for the accuracy which has been perceived as an insufficient measure for evaluating the defence performance, average distortion is an important quantifiable inductor [40]. Although these parameters have provided well insights into the defence behaviors, some fine-grained features that can identify the source of adversarial example are still not presented.

In this article, a set of experiments are designed to validate the applicability of the DeSVig in real industrial scenarios. The used industrial datasets include 1000 images (8 classes) of common electronic components. Since most of the common electronic components are often encapsulated with black and

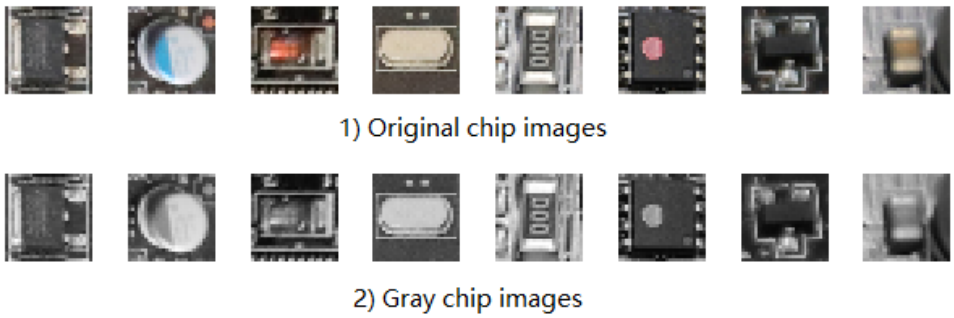


Fig. 5. Industrial datasets on common electronic components

white surfaces, all images can be transferred to gray images to save training time, as illustrated in Fig. 5. The size of used image is also reshaped as 28×28. In this experiment, the $keep\_prob$ is 0.25 and the activation function is $RELU$.

1) Controllable Cost: Controllable Cost is a key reference that decides the applicability of a new defence method. In industrial scenarios, available resources may be very limited especially when some emergent events appear. However, there are few studies that evaluate such critical inductor. By carefully collecting and deeply analyzing, in this paper we provide the cost comparison among different defences. Additional costs caused by defences are mainly divided into two parts: 1) training cost; and 2) running cost. Obviously, adversarial training consumes too many resources during the training process. Training cost of differential privacy is controllable by dynamically adjusting the size of added noises.

However, the cost of differential privacy is also proportional to the number of inputs in the whole network. The DeSVig reduces the additional cost caused by defences by decoupling the defence control logic from the data plane. Fig. 6 shows the cost comparison among different defences. Significant cost reduction makes DeSVig is more applicable especially in large scale industrial systems.

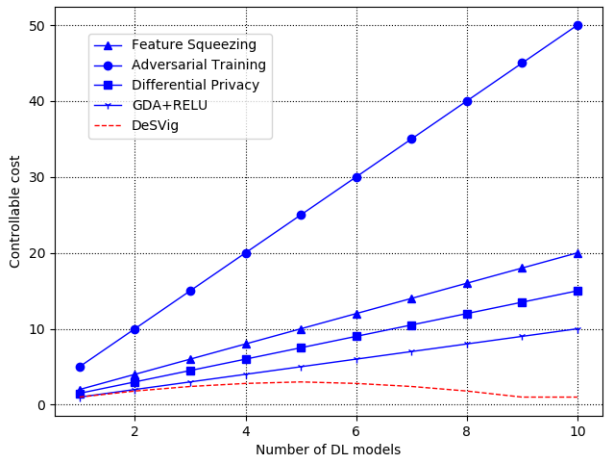


Fig. 6. Controllable cost.

2) Swift vigilance: Low delay is to fulfil the real requirements of industrial control. Previous experiments evaluate the effectiveness of defence models usually by imputing all adversarial examples into a single pre-trained DL model. However, for deep learning in a real physical world especially in industrial fields, it is common to see that the adversarial examples are often mixed with normal examples and these adversarial examples may be feed into multiple deep learning models synchronously.

The DeSVig aims to swiftly detect the unknown adversarial examples in distributed DL models. In a distributed scenario, it not only requires the incumbent DL model to identify which is the adversarial example from random inputs accurately but also requires the other involved DL models to swiftly circumvent the adversarial examples. Therefore, evaluating the latency is a must before deploying the defence methods in real IAISs. As shown in Fig. 7, we record the measuring delay of different defences. The additional delay is the measuring value minus the baseline. Therefore, the additional delay of DeSVig is about $2ms$, while the additional delay of the other defences are in range of $20ms \sim 60ms$.

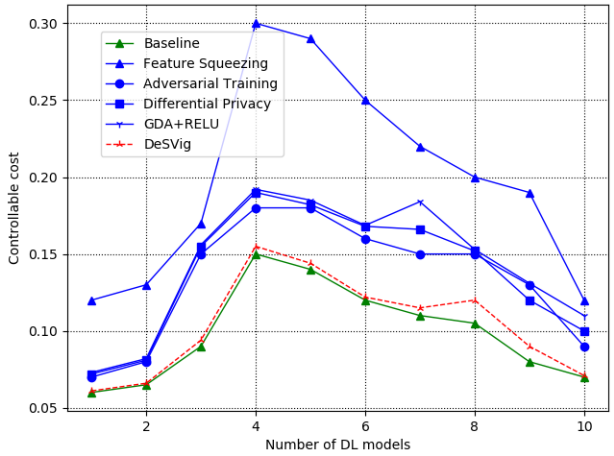


Fig. 7. Swift vigilance.

3) Elastic Robustness: As described in Section III, there may multiple DeSVig controllers in a large smart plant. To evaluate the robustness of the whole industrial network, we propose to compute the controller credibility. Controller credibility depends on the available resources and loads on the DeSVig controller.

In mobile edge computing, edge nodes' resources are often offloaded from the cloud by using some incentive mechanism. Thus, the robustness evaluation problem can be formulated as a constraint in the incentive mechanism. With the increase

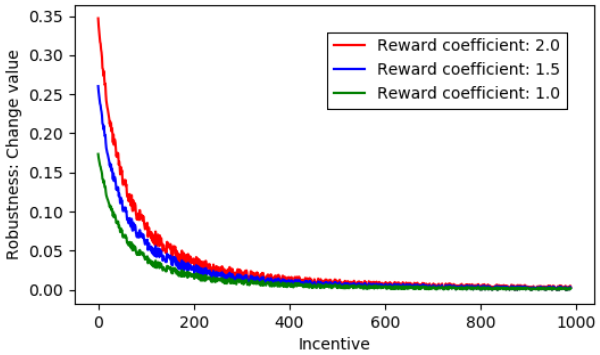


Fig. 8. Robustness estimation: a smaller dynamical change of IFGor's reward provides more robust defense accuracy at different rewards coefficients.

of incentive, edge nodes will gain more resources. And then, the defence effect of DeSVig to adversarial examples will be improved significantly. Finally, the robustness change of IAIS to adversarial examples will be reduced as illustrated in Fig. 8. And also, different reward coefficients can not resist the robustness convergence of IAIS. This interesting phenomenon will help us aggregate more edge nodes in the whole system to identify the unknown adversarial examples.

V. Conclusion

In this paper, we propose DeSVig, a decentralized swift vigilance framework to identify adversarial attacks in industrial artificial intelligence systems (IAISs). The most important advantage of our work is that the proposed scheme can significantly reduce the failure risks of being deceived by adversarial examples with no requirement on complex robustness reinforcement, which is critical for safety-prioritized, delay-sensitive environments. The DeSVig is highly distributed and enables fast response, which is essential to improve the effectiveness of recognizing adversarial examples. The challenges on privacy-preserving and ultra-low latency in industrial scenarios are resolved by integrating differential privacy, MEC agents, generative adversarial networks (GANs), and perceptional hashing. Recouping defence control from deep learning model significantly improves the vigilance delay. In our experiments, adversarial examples of industrial electronic components are generated by several classical attacking models in Adversarial-Robustness-Toolbox. Experimental results demonstrate the DeSVig is more robust, efficient, and scalable than some state-of-art defences.

References

[1] J. Wang, Y. Ma, L. Zhang, RX.Gao, D. Wu, "Deep learning for smart manufacturing: Methods and applications, " *Journal of Manufacturing Systems*, vol. 48, no. C, pp. 144-156, 2018.

[2] C. Zhang, P. Patras, H. Haddadi, "Deep learning in mobile and wireless networking: A survey, " *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2224-2287, 2019.

[3] A. Kamilaris, FX. Prenafeta-Bold, "Deep learning in agriculture: A survey, " *Computers and Electronics in Agriculture*, vol. 14, pp. 70-90, 2018.

[4] Z. Qin, H. Ye, GY. Li, and BHF. Juang, "Deep learning in physical layer communications," *IEEE Wireless Communications*, vol. 26, no. 2, pp. 93-99, April 2019.

[5] G. Li, J. Wu, J. Li, K. Wang, T. Ye, "Service Popularity-based Smart Resources Partitioning for Fog Computing-enabled Industrial Internet of Things, " *IEEE Transactions on Industrial Informatics (TII)*, vol. 14, no. 10, pp. 4702-4711, 2018.

[6] H. Li, K. Ota, M. Dong, "ECCN: Orchestration of Edge-Centric Computing and Content-Centric Networking in the 5G Radio Access Network, " *IEEE Wireless Communications*, vol. 25, no. 3, pp. 88-93, 2018.

[7] AH. Sodhro, S. Pirbhulal, VHC. Albuquerque, "Artificial Intelligence Driven Mechanism for Edge Computing based Industrial Applications," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4235-4243, 2019.

[8] S. Wang, T. Tuor, et al., "When Edge Meets Learning: Adaptive Control for Resource-Constrained Distributed Machine Learning, " *INFOCOM*, pp. 1-9, 2018.

[9] W. Zhang, et al., "MASM: A Multiple-algorithm Service Model for Energy-delay Optimization in Edge Artificial Intelligence, "*IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4216-4224, 2019.

[10] C. Liu, Y. Gao, et al., "A New Deep Learning-Based Food Recognition System for Dietary Assessment on An Edge Computing Service Infrastructure, " *IEEE Transactions on Services Computing*, vol. 11, no. 2, pp. 249-261, 2018.

[11] G. Li, G. Xu, K. Sangaiah, J. Wu, J. Li, "EdgeLaaS: Edge Learning as a Service for Knowledge-Centric Connected Healthcare, "*IEEE Network*, vol. PP, no. 99, pp. 1-1, 2018.

[12] K. Ota, M. Dong, J. Gui, A. Liu, "QUOIN: Incentive Mechanisms for Crowd Sensing Networks, " *IEEE Network*, vol. 32, no. 2, pp. 114-119, 2018.

[13] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, IJ. Goodfellow, and R. Fergus, "Intriguing properties of neural networks, " *ICLR*, pp. 1-10, 2013.

[14] X. Yuan, P. He, Q. Zhu, X. Li, "Adversarial Examples: Attacks and Defenses for Deep Learning,"*IEEE Transactions on Neural Networks and Learning Systems*, Vol. 30, No. 9, pp. 2805-2824, 2019.

[15] A. Athalye, N. Carlini, D. Wagner, "Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples, " *ICML*, pp. 274-283, 2018.

[16] SR. Chhetri, N. Rashid, S. Faezi, MAA. Faruque, "Security trends and advances in manufacturing systems in the era of industry 4.0, " *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1-8, 2017.

[17] BD. Rouani, M. Samragh. T. Javidi, F. Koushanfar, "Safe Machine Learning and Defeating Adversarial Attacks, " *IEEE Security & Privacy*, vol. 17, no. 2, pp. 31-38, 2019.

[18] J. Sachs, LA. A. Andersson, et al, "Adaptive 5G Low-Latency Communication for Tactile InternEt Services, " *Proceedings of the IEEE*, vol. 107, no. 2, pp. 325-349, 2019.

[19] Z. Xiang, F. Gabriel, et al., "Reducing Latency in Virtual Machines: Enabling Tactile Internet for Human-Machine Co-Working, " *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 5, pp. 1098-1116, 2019.

[20] N. McKeown, T. Anderson, et al, "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69-74, 2008.

[21] N. Akhtar, A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey, " *IEEE Access*, 6, pages: 14410-14430, 2018.

[22] A. E. Aydemir, A. Temizel, T. T. Temizel, "The Effects of JPEG and JPEG2000 Compression on Attacks using Adversarial Examples, "In Preprint *arXiv:1803.10418*, 2018.

[23] A. Prakash, N. Moran, et al., "Deflecting Adversarial Attacks With Pixel Deflection, "In *CVPR*, pp. 8571-8580, 2018.

[24] S. Song, Y. Chen, NM. Cheung, CCJ. Kuo, "Defense Against Adversarial Attacks with Saak Transform, " In preprint *arXiv:1808.01785*, 2018.

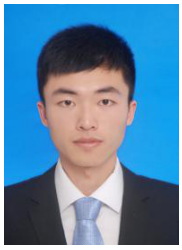
[25] J. Xu, K. Ota, M. Dong, "Energy Efficient Hybrid Edge Caching Scheme for Tactile Internet in 5G, " *IEEE Transactions on Green Communications and Networking (TGCN)*, vol. 3, no. 2, pp. 483-493, 2019.

[26] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, G. Fettweis, "5G-Enabled Tactile Internet, " *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 460-473, 2016.

[27] J. Wu, M. Dong, K. Ota, J. Li, Z. Guan, "FCSS: Fog computing based content-aware filtering for security services in information centric social networks, " *IEEE Transactions on Emerging Topics in computing*, DOI: 10.1109/TETC.2017.2747158, 2017.

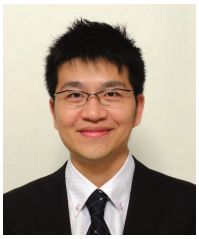
[28] AFM. Piedrahita, V. Gaur, J. Giraldo, A. Crdenas, SJ. Rueda, "Leveraging Software-Defined Networking for Incident Response in Industrial Control Systems, " *IEEE Software*, vol. 35, no. 1, pp. 44-50, 2018.

[29] H. Li, K. Ota, M. Dong, "Virtual Network Recognition and Optimization in SDN-enabled Cloud Environment, " *IEEE Transactions on Cloud Computing (TCC)*, DOI: 10.1109/TCC.2018.2871118, 2018.
[30] IJ. Goodfellow, J. Shlens, C. Szegedy, "Explaining and Harnessing Adversarial Examples, " In *ICLR*, pages 212-227, 2015.
[31] A. Raghunathan, J. Steinhardt, P. Liang, "Certified Defenses against Adversarial Examples, " *arXiv:1801.09344v1*, 2018.
[32] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, S. Jana, "Certified Robustness to Adversarial Examples with Differential Privacy, " *arXiv:1802.03471v3*, 2018.
[33] H. Ye, G.Y. Li, B.H.F. Juang, and K. Sivanesan, "Channel agnostic end-to-end learning based communication systems with conditional GAN, " *IEEE Globecom Workshops*, pp. 1-5, 2018.
[34] A. Kurakin, I. Goodfellow, S. Bengio, "Adversarial examples in the physical world, " In *ICLR*, 2017.
[35] STK. Jan, J. Messou, et al., "Connecting the Digital and Physical World: Improving the Robustness of Adversarial Attacks, " In *AAAI*, 2019.
[36] N. Maria-Irina, S. Mathieu, et al., "Adversarial Robustness Toolbox v0.3.0, " In *arXiv:1807.01069*, 2018.
[37] W. Fang, HM. Hu, Z. Hu, S. Liao, B. Li, "Perceptual hash-based feature description for person re-identification, " *Neurocomputing*, vol. 272, pp. 520-531, 2018.
[38] SM. Moosavi-Dezfooli, A. Fawzi, P. Frossard, "DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks, " *CVPR*, pp. 2574-2582, 2016.
[39] F. Liu, "Generalized Gaussian Mechanism for Differential Privacy, " *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 4, pp. 747-756, 2019.
[40] V. Zantedeschi, M.I. Nicolae, A. Rawat, "Efficient defenses against adversarial attacks, " In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security (AISec17)*, pp. 39-49, 2017.
[41] P. Samangouei, M. Kabkab, R. Chellappa, "Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models, " *ICLR*, pp. 1-17, 2018.
[42] R. Anirudh, JJ. Thiagarajan, B. Kailkhura, T. Bremer, "MimicGAN: Corruption-Mimicking for Blind Image Recovery & Adversarial Defense, " *arXiv:1811.08484*, 2018.

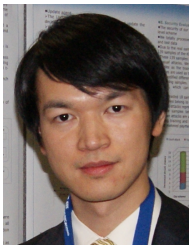


Gaolei Li (S'16) received the B.S. degree in electronic information engineering from Sichuan University, Chengdu, China, in 2015 and he is pursuing the Ph.D. degree in School of Cybe Security, Shanghai Jiao Tong University, Shanghai, China. From September 2018 to September 2019, he was supported by China Scholarship Council to visit the Muroran Institute of Technology, Japan. His research interests are focusing on deep learning, privacy security, etc.

Kaoru Ota (S'09-M'12) was born in Aizu-Wakamatsu, Japan. She received M.S. degree in Computer Science from Oklahoma State University, USA in 2008, B.S. and Ph.D. degrees in Computer Science and Engineering from The University of Aizu, Japan in 2006, 2012, respectively. She is currently an Assistant Professor with Department of Sciences and Informatics, Muroran Institute of Technology, Japan. From March 2010 to March 2011, she was a visiting scholar at University of Waterloo, Canada. Also she was a Japan Society of the Promotion of Science (JSPS) research fellow with Graduate School of Information Sciences at Tohoku University, Japan from April 2012 to April 2013. Her research interests include Wireless Networks, Cloud Computing, and Cyber-physical Systems. Dr. Ota has published about 240 papers and received many best conference paper awards. She is an editor of top journals (such as TVT, IOTJ, WCL, PPNA), as well as a guest editor of ACM Transactions on Multimedia Computing, Communications and Applications (leading), IEEE Internet of Things Journal, IEEE Communications Magazine, IEEE Network, IEEE Wireless Communications, IEEE Access, etc. She is the recipient of IEEE TCSC Early Career Award 2017 and The 13th IEEE ComSoc Asia-Pacific Young Researcher Award 2018.



Mianxiong Dong (S'07-M'13) received B.S., M.S. and Ph.D. in Computer Science and Engineering from The University of Aizu, Japan. He is currently a Professor in the Department of Sciences and Informatics, Advisor to Executive Director, and Vice Director of Office of Institutional Research at the Muroran Institute of Technology, Japan. He was a JSPS Research Fellow with School of Computer Science and Engineering, The University of Aizu, Japan and was a visiting scholar with BBCR group at University of Waterloo, Canada supported by JSPS Excellent Young Researcher Overseas Visit Program from April 2010 to August 2011. Dr. Dong was selected as a Foreigner Research Fellow (a total of 3 recipients all over Japan) by NEC C&C Foundation in 2011. He has received best paper awards from IEEE HPCC 2008, IEEE ICESS 2008, ICA3PP 2014, GPC 2015, IEEE DASC 2015, IEEE VTC 2016-Fall, FCST 2017, 2017 IET Communications Premium Award and IEEE ComSoc CSIM Best Conference Paper Award 2018. He has been serving as the Vice Chair of IEEE Communications Society Asia/Pacific Region Information Services Committee and Meetings and Conference Committee, Leading Symposium Chair of IEEE ICC 2019, Student Travel Grants Chair of IEEE GLOBECOM 2019, and Symposium Chair of IEEE GLOBECOM 2016, 2017. He is the recipient of IEEE TCSC Early Career Award 2016, IEEE SCSTC Outstanding Young Researcher Award 2017, The 12th IEEE ComSoc Asia-Pacific Young Researcher Award 2017, Funai Research Award 2018 and NISTEP Researcher 2018 (one of only 11 people in Japan) in recognition of significant contributions in science and technology from MEXT. He is currently the Member of Board of Governors and Chair of Student Fellowship Committee of IEEE Vehicular Technology Society, and Treasurer of IEEE ComSoc Japan Joint Sections Chapter.



Jun Wu (S'08-M'12) is an Associate Professor of School of Cyber Security, Shanghai Jiao Tong University, China and a Vice Director in National Engineering Laboratory for Information Content Analysis Technology. He obtained his Ph.D. Degree in Information and Telecommunication Studies at Waseda University, Japan. He was a postdoctoral researcher for the Research Institute for Secure Systems (RISEC), National Institute of Advanced Industrial Science and Technology (AIST), Japan, from 2011 to 2012. He worked as a researcher for the Global Information and Telecommunication Institute (GITI), Waseda University, Japan, from 2011 to 2013. He has hosted and participated in several research projects for the National Natural Science Foundation of China, National 863 Plan and 973 Plan, Japan Society of the Promotion of Science (JSPS) projects, etc. He is the associate editor of IEEE Access. He has been a Guest Editor for the IEEE Sensors Journal and TPC Member of more than ten international conferences including WINCON, ICC, GLOBECOM, etc. His research interests include the advanced computing and communications techniques of software-defined networks (SDN), smart grids, Internet of Things, etc.

Jianhua Li is a professor/Ph.D. supervisor and the dean of Academy of Cyber Security, Shanghai Jiao Tong University, Shanghai, China. He got his BS, MS and Ph.D. degrees from Shanghai Jiao Tong University, in 1986, 1991 and 1998, respectively. He was the chief expert in the information security committee experts of National High Technology Research and Development Program of China (863 Program) of China. He is also a committee expert of Information Technique Standardization Committee of Shanghai, China. He was the leader of more than 30 state/province projects of China, and published more than 200 papers. He got the Second Prize of National Technology Progress Award of China in 2005. He got the First Prize of National Technology Progress Award of Shanghai in 2003 and 2004, and he got two First Prize of National Technology Progress Awards of Shanghai in 2004. His research interests include cyberspace security, Data science, Next Generation Networks, etc.