Hierarchical Adversarial Attacks Against Graph-Neural-Network-Based IoT Network Intrusion Detection System

Xiaokang Zhou, *Member, IEEE*, Wei Liang, *Member, IEEE*, Weimin Li, *Member, IEEE*, Ke Yan, *Member, IEEE*, Shohei Shimizu, and Kevin I-Kai Wang, *Member, IEEE*

***Abstract*—The advancement of Internet of Things (IoT) technologies leads to a wide penetration and large-scale deployment of IoT systems across an entire city or even country. While IoT systems are capable of providing intelligent services, the large amount of data collected and processed in IoT systems also raises serious security concerns. Many research efforts have been devoted to design intelligent network intrusion detection system (NIDS) to prevent misuse of IoT data across smart applications. However, existing approaches may suffer from the issue of limited and imbalanced attack data when training the detection model, which make the system vulnerable especially for those unknown type attacks. In this study, a novel hierarchical adversarial attack (HAA) generation method is introduced to realize the level-aware black-box adversarial attack strategy, targeting the graph neural network (GNN)-based intrusion detection in IoT systems with a limited budget. By constructing a shadow GNN model, an intelligent mechanism based on a saliency map technique is designed to generate adversarial examples by effectively identifying and modifying the critical feature elements with minimal perturbations. A hierarchical node selection algorithm based on random walk with restart (RWR) is developed to select a set of more vulnerable nodes with high attack priority, considering their structural features, and overall loss changes within the targeted IoT network. The proposed HAA generation method is evaluated using the open-source data set UNSW-SOSR2019 with three baseline methods. Comparison results demonstrate its ability in degrading the classification precision by more than 30% in the two state-of-the-art GNN models, GCN and JK-Net, respectively, for NIDS in IoT environments.**

***Index Terms*—Adversarial attack, deep learning, graph neural network (GNN), Internet of Things (IoT), network intrusion detection.**

Manuscript received April 2, 2021; revised June 1, 2021, July 15, 2021, and September 20, 2021; accepted November 2, 2021. Date of publication November 24, 2021; date of current version June 7, 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFE0117500, Grant 2019YFE0190500; in part by the National Natural Science Foundation of China under Grant 62072171; in part by the Natural Science Foundation of Hunan Province of China under Grant 2020SK2089; and in part by the by Open Fund of Key Laboratory of Hunan Province under Grant 2017TP1026. *(Corresponding author: Wei Liang.)*

Xiaokang Zhou and Shohei Shimizu are with the Faculty of Data Science, Shiga University, Hikone 522-8522, Japan, and also with the RIKEN Center for Advanced Intelligence Project, RIKEN, Tokyo 103-0027, Japan (e-mail: zhou@biwako.shiga-u.ac.jp; shohei-shimizu@biwako.shiga-u.ac.jp).

Wei Liang is with the Base of International Science and Technology Innovation and Cooperation on Big Data Technology and Management, Hunan University of Technology and Business, Changsha 410205, China (e-mail: weiliang@csu.edu.cn).

Weimin Li is with the School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China (e-mail: wmli@shu.edu.cn).

Ke Yan is with the Department of the Built Environment, College of Design and Engineering, National University of Singapore, Singapore 117566 (e-mail: yanke@nus.edu.sg).

Kevin I-Kai Wang is with the Department of Electrical, Computer, and Software Engineering, The University of Auckland, Auckland 1010, New Zealand (e-mail: kevin.wang@auckland.ac.nz).

Digital Object Identifier 10.1109/JIOT.2021.3130434

I. Introduction

THE proliferation of Internet of Things (IoT) technologies and systems are growing at an unprecedented rate. The scale of modern IoT systems goes far beyond the individual level, with interconnected IoT devices that are widely spread across the entire cities or even countries. Supported by the increasing communication speed and bandwidth, IoT devices are capable of collecting, transmitting, and processing an enormous amount of data [1], [2]. These IoT systems, associated with the collected data, are offering great opportunities in designing and providing intelligent services in different applications, such as intelligent transportation, automated surveillance, and smart cyber–physical systems [3], [4]. However, the collected IoT data also contain sensitive information and therefore require more attention on privacy protection and reliable data security issues.

To deal with such increasing privacy and security concerns, modern IoT or distributed systems need to be able to detect and prevent network intrusions in a more intelligent way. Many research efforts have been devoted to develop machine learning or deep learning-based approaches for network intrusion detection system (NIDS), in order to prevent any deviation or misuse in IoT systems and infrastructures [5]–[7]. Although NIDS has been well exploited in detecting malicious network activities, one of the main vulnerabilities of existing NIDS is the lack of ability to detect unknown types of network intrusion, due to the limited or imbalanced intrusion data during the model training process [8], [9]. In addition, existing machine learning approaches are not able to handle multidomain intrusion detections, which calls for the further exploration on the hybrid deep learning architecture [6], [10], [11].

As a typical type of neural network in deep learning models, graph neural network (GNN) has demonstrated its promising performance in dealing with a graph or network data [12]. However, it still suffers when facing limited or imbalanced training data, and can also be vulnerable to adversarial attacks. In recent years, adversarial attacks or examples have been proved as one significant tool in analyzing deep neural networks in terms of their theoretical property and practical

performance [13]. It can affect deep graph learning algorithms with small and imperceptible perturbations and lead to inaccurate classifications or wrong decisions [14]. Therefore, further investigations are necessary in GNN-based NIDS. In general, adversarial attacks can be classified into three types, namely, white-box attacks, gray-box attacks, and black-box attacks, according to how much the attacker knows about the learning model. In white-box attacks, the entire model structure, parameters, input data, and labels are completely exposed to the attacker. While in gray-box attacks, the attacker has partial information of the training model. In black-box attacks, the attacker knows almost nothing about the network except the input data, which offers a more realistic representation for real threat scenarios.

In this study, a new hierarchical adversarial attack (HAA) generation method is proposed, which can be used to examine the robustness and generality of an NIDS designed for typical IoT applications. Considering the black-box attack scenario, a shadow GNN model is constructed with the intercepted network packets and extracted input data features to imitate the original model. The saliency map technique is used to find some critical elements in the feature vector, following which adversarial examples can then be generated to flip the classification labels with minimal modifications on those identified critical elements. In addition, a random walk with restart (RWR)-based algorithm is developed to select a set of nodes with high attack priority based on structural features and overall loss changes within the targeted IoT network. The main contributions of this article can be summarized as follows.

1) An integrated framework for the level-aware black-box adversarial attack strategy is designed and constructed to compromise the GNN-based NIDS in typical IoT environments with a limited budget.
2) An intelligent adversarial example generation mechanism is developed based on a constructed shadow GNN model, which can effectively modify the critical feature elements identified using saliency mapping with minimal perturbations.
3) An RWR-based hierarchical node selection algorithm, which considers both the link analysis and loss change in initializing and updating the transfer matrix, is designed to efficiently identify and select a set of more vulnerable nodes to attack the GNN model.

The remainder of this article is organized as follows. Section II presents the summary of related works on GNN-based modeling and adversarial attacks against GNN in modern IoT systems. Section III introduces the overall application scenario and problem formulations, followed by the proposed HAA generation mechanism explained in Section IV. Section V presents and discusses the evaluations using the open-source data set, and Section VI concludes this study and gives a promising perspective on future research.

II. Related Work

In this section, two emerging research directions related to this study, including the GNN-based network model and adversarial attacks against GNN, are addressed, respectively.

A. GNN-Based Network Modeling With IoT

With the rapid evolution of deep learning techniques in various smart applications for classification or prediction tasks, GNN has become an emerging learning paradigm when dealing with interdependent data with complex relationships in network modeling [12]. Several researches have explored the use of GNN in big data mining, machine learning, and IoT applications. Zhou *et al.* [15] introduced a so-called reinforced spatial-temporal attention GNN model for traffic prediction, which utilized the diffusion convolution neural network and a temporal attention mechanism to analyze spatial dependencies and temporal dynamics from traffic sensor networks. Zhang *et al.* [16] applied GNN in the modeling of IoT equipment. They reconstructed the input data using a variational autoencoder to analyze the temporal and inner logic relations of data. Rusek *et al.* [17] employed GNN to model the graph-structured information and designed a message-passing function to extract complex relationships from network topologies and routing configurations based on the generalized linear models, which could be applied for routing optimization and network planning. Guo and Wang [18] built a recommendation framework based on deep GNN for future IoT. They modeled feature spaces into two graph networks and used matrix factorization to improve the missing rating values in a user-item rating matrix. Cui *et al.* [19] presented a deep learning framework, in which the traffic network was modeled by a graph convolutional long short-term memory neural network. They designed a graph convolution operator to learn the spatial and temporal dependency and defined two regularization terms to optimize loss functions in model training. To identify graph patterns in directed role-based conceptual attributed graph, Krleža and Fertalj [20] proposed a fuzzy GNN for graph matching. They built this model based on the combination of graph element comparison using fuzzy logic and graph structure verification using a recursive neural network. Shen *et al.* [21] involved GNN into the large-scale radio resource management as a graph optimization problem. They designed a so-called message passing GNN, where agents were considered as nodes and communication channels were considered as edges, to achieve a low-complexity neural network operation. Gama *et al.* [22] introduced two improvements of GNN architectures. One called selection GNN replaced the linear time-invariant filter for convolutional feature generation, the other one called aggregation GNN used a temporal structure to capture the graph topology. Both of them were applied in synthetic networks for source localization. Zhu *et al.* [23] constructed a hierarchical unsupervised model based on cycle adversarial networks for graph alignment, in which an optimization module for group structure aggregation was developed to recognize similar IoT devices in different networks.

B. Adversarial Attacks Against GNN

Recently, adversarial attacks, especially in wireless communications, have drawn a lot of attention for vulnerability analysis, using deep learning techniques. Miller *et al.* [24]

conducted a survey on adversarial learning attacks in DNN-based classifications, and compared a series of defenses against test-time evasion, backdoor data poisoning, reverse engineering attacks, etc. Krithivasan *et al.* [25] focused on adversarial sparsity attacks, aiming to degrade the latency, and energy consumption in DNN. They employed adversarial perturbations to generate adversarial inputs for sparsity attacks, which could modify the model input to reduce the activation sparsity, but not affect the classification accuracy. Takahashi [26] investigated indirect adversarial attacks in graph convolutional neural networks and discussed a detection method to find one new attack which could poison node features and lead to the misclassification. Yuan and He [27] presented an adversarial dual network learning model for DNN-based defense. They formulated this problem using a generative adversarial network and developed a detector with a generative cleaning network to clean up the attack noise following a randomized nonlinear image transform. Based on the investigation of gradient-based attacks in GNN, Lin *et al.* [28] discussed an exploratory attack method to add adversarial noise in the graph topology, in order to avoid the misinformation and improve the semi-supervised classifications. Ioannidis and Giannakis [29] built a semi-supervised learning framework to deal with the perturbed networked data, in which they applied a link-dithering method to reconstruct the original neighborhood structure and used the graph convolutional network to extract features from unperturbed neighborhoods. Xu *et al.* [30] introduced an adversarial training scheme based on DNN. Considering both the targeted and untargeted attacks, they generated adversarial examples to improve the resistibility of the learning model, which could be applied in remote sensing scene classifications. Apruzzese *et al.* [31] developed a deep reinforcement learning scheme to generate realistic attack samples in an augmented training set, which could be applied to enable a more resilient detector for cyber security against evasion attacks. Li and Li [32] introduced a method called "mixture of attacks," and evaluated it against 26 evasion attacks for machine learning-based malware detection. They conducted the adversarial training based on a mixture of attacks, to enhance the ensemble of DNN. Sagduyu *et al.* [33] analyzed the wireless attack and designed a new type called over-the-air spectrum poisoning attack based on adversarial neural networks. They applied it in a wireless communication scenario and showed that the adversarial deep learning strategy could facilitate the learning of the transmitter's behavior, so as to boost the poisoning attacks.

III. Preliminary and Problem Definition

In this section, a brief introduction of a typical adversarial attack scenario within IoT networks is given first, followed by the problem definition and formulation for the adversarial attack generation in the specific GNN-based NIDS.

A. Application Scenario

In a typical IoT system, such as a surveillance system in a smart city, numerous smart nodes are interconnected across different IoT networks, presenting a hierarchical structure.

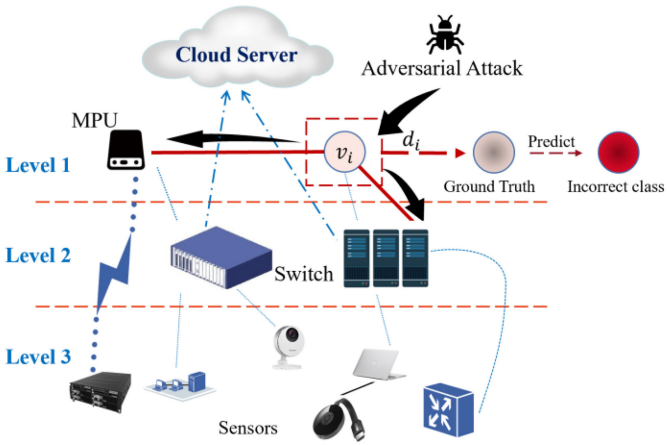


Fig. 1. Scenario on level-aware black-box adversarial attacks against GNN-based NIDS.

Each level of IoT network is composed of different kinds of smart nodes, depending on their functionalities or types of services. As shown in Fig. 1, the bottom level typically consists of various sensor nodes, e.g., digital cameras in a cloud-based surveillance system, or a set of sensors deployed in the flow line system for smart manufacturing applications, for the purpose of collecting raw data which will be used in the higher-level IoT services and applications. The second level contains several smart devices which may aggregate the information from the bottom level in an area. For example, programmable logic controllers (PLCs) for smart manufacturing or roadside units (RSUs) for intelligent transportation are deployed in the second level to handle specific control tasks in IoT-based applications. Finally, the topmost level is deployed with the master processing unit (MPU) to manage the aggregated information from the entire IoT system. Specifically, the level of a network node v_i can be determined by the weighted correlations from its neighbors similar to PageRank, which is initially measured according to the corresponding outlink connection d_i in this study. In addition, the black-box attack scenario is adopted to ensure the practicality of the proposed model, which means the attacker knows nothing about the model parameters but only the input data.

Referring to Fig. 1, considering a typical adversarial attack for IoT systems with a hierarchical network structure, an attacker who has a limited budget can only compromise a limited group of nodes. When an attack happens, the GNN-based NIDS may detect it by predicting each node in the network as a compromised node or normal node. Intuitively, it would be easier to compromise the entire IoT network by attacking nodes at higher levels of a network. In this study, with the IoT network structure presented, an HAA strategy targeting node selection tasks is designed to confuse the GNN model, which may lead to the misprediction or misjudgment of a compromised node as a normal one.

B. Problem Formulation

In this study, we consider a graph model $G = (V, E, D, X)$ to represent a typical IoT network, where $V = \{v_1, v_2, \ldots, v_P\}$

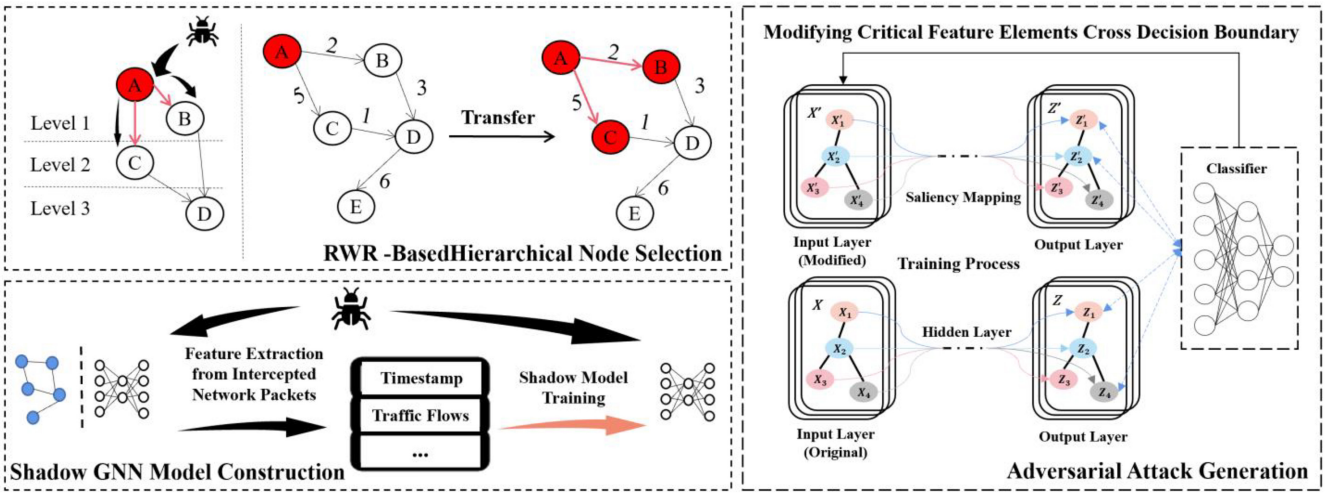


Fig. 2. Overview of HAA generation against GNN-based NIDS.

is a nonempty set of nodes in the network, $E = \{e_1, e_2, \ldots, e_Q\}$ is the edge set associated with V, and $D = \{d_1, d_2, \ldots, d_P\}$ denotes a set of measures to quantify the hierarchical relations based on link analysis among all the nodes in V. $P = |V|$ and $Q = |E|$ denote the total number of nodes and edges, respectively. $X = \{x_1, x_2, \ldots, x_P\}$ is the set of features that corresponds to each individual node in V, where each $x_i \in \mathbb{R}^L$ in X is the L-dimensional feature vector for each node $v_i \in V$.

Given an NIDS deployed with a GNN-based classification model, the GNN model may classify all the nodes into C classes. We define $y_i \in \{1, 2, \ldots, C\}$ as the ground truth class for node v_i, and $\hat{y}_i = f(X; \theta)$ as the predicted result for v_i, where θ indicates the set of parameters in the GNN model. As we discussed above, a practical black-box attack scenario is adopted, in which the NIDS's integrity is protected thus the attacker cannot directly obtain the detailed model structure and parameters. Accordingly, the attacker may attempt to degrade the classification performance of $f(*)$ by adding perturbations into the original G, and turn it into a perturbed graph G'. Specifically, we consider the perturbation only occurs in nodes of the graph model, which results in a perturbed $G' = (V', E, D', X')$.

The problem definition is given as follows. Assuming an attacker tries to degrade the classification performance of a GNN model by adding perturbation to the original G, it is needed to confuse the GNN model by perturbing the original feature vector set X to X'. Therefore, given a predicted classification result $\hat{y}$ comparing to the ground truth y, the goal is to obtain an optimal perturbed X' based on the loss optimization [34], [35], which can be described as follows:

$$\begin{aligned} &\underset{|X'|<h}{\operatorname{argmax}} \ \mathcal{L}_{\text{attack}}\big(f\big(X'; \theta^*\big), y\big) \\ &\quad \text{s.t.} \ \ \theta^* = \operatorname{argmin}_{|X'|<h} \mathcal{L}_{\text{predict}}(f(X; \theta), y) \end{aligned} \tag{1}$$

where $\mathcal{L}_{\text{attack}}$ and $\mathcal{L}_{\text{predict}}$ are the cross-entropy loss of attacker and original GNN, respectively. θ^* denotes the optimal parameters and $f(X'; \theta^*)$ denotes the prediction result based on θ^* using X'. In particular, we set $|X'| < h$, in which h is the maximum number of the nodes to be attacked initially due to the limited budget.

IV. HAA Generation Against GNN-Based NIDS

In this section, we first introduce the overview of HAA generation against the GNN-based NIDS. The modeling of shadow GNN for HAA generation is then addressed. Then, we discuss how to effectively generate the adversarial examples and how to efficiently compromise the GNN-based NIDS using a hierarchical node selection strategy.

A. Overview of HAA Generation

The overview of the proposed HAA generation method is shown in Fig. 2, which includes three essential parts. To generate black-box attacks, it is important to imitate the original GNN model. Thus, the first part is to construct a shadow GNN model, based on the intercepted network packets and extracted features from the input data of the original model. Then, the second part is to select an optimum node to attack. An RWR-based mechanism is designed to measure each node in the constructed shadow model, and the node with the higher weight is more likely to be selected as the attack node due to the limited budget. Finally, the third part is to generate adversarial examples based on the shadow GNN model, which aim to perturb critical features and alter the classification labels. In summary, the adversarial examples are generated based on the constructed shadow GNN model with the intercepted network packets, which can efficiently mislead the detection or prediction result, and ensure the attack damage to the target system via selecting some more vulnerable nodes.

B. Generation of the Shadow GNN Model

As mentioned previously, in a black-box attack scenario, attackers can only intercept the network traffic data, and assume a known GNN structure for training. They have no access to the actual parameters of the GNN model used in the NIDS. Therefore, it is necessary to construct a shadow model

to assist in generating adversarial examples that can confuse the GNN-based NIDS. In particular, assuming the attacker can monitor the traffic flow in or out of the target model and collect a certain number of network packets, these intercepted network packets, including the detailed information of source IP, destination IP, timestamp, traffic flows, etc., can be utilized to learn a shadow GNN model with an existing GNN network structure. The process of generating the shadow GNN model can be split into two parts as: 1) feature extraction and 2) shadow model training.

First, a feature extractor is constructed to extract the critical information from the intercepted network packets, transform it into a feature vector x_i, and form a feature set $X = \{x_1, x_2, \ldots, x_P\}$. A shadow model described as $f'(*)$ is then initialized based on the extracted X and original y, to reproduce the original $f(*)$ as much as possible.

Second, we go further to learn the temporary parameters θ' in the shadow model, thus it may output the same or similar prediction results as the original GNN model. Specifically, given the predicted result of the shadow model as $\hat{y}'_i = f'(x_i; \theta')$, the goal is to minimize the error between the new $\hat{y}'_i$ and $\hat{y}_i$ in the original GNN model. The root mean square error (RMSE) is used to measure the error between $\hat{y}'_i$ and $\hat{y}_i$, and a classical gradient descent method is employed to optimize the parameters to make sure that the shadow model is as close to the original model as possible.

C. Adversarial Example Generation

As the core issue in the HAA model, adversarial examples are generated by the shadow GNN to modify the feature set $X' = \{x'_1, x'_2, \ldots, x'_N\}$ from X, so as to disguise malicious packets as the normal or vice versa. Thus, the key idea is to learn the decision boundary from the GNN discriminator, then modify the features based on the original data packet and change it across the decision boundary with minimal modifications. Specifically, the modified feature x'_i for node v_i can be defined as follows:

$$x'_i = x_i + \varepsilon_i. \tag{2}$$

Obviously, the goal is to minimize $||\varepsilon_i||$ which satisfies $f(x'_i) \neq f(x_i)$, and how to distinguish the critical feature elements in the feature space becomes essential to generate adversarial examples with minimal perturbations to alter the labels.

In particular, the saliency map [37] is utilized to identify the critical elements from the feature space in a gradient-based back propagation process. Based on revising these identified key elements, adversarial examples are generated by adding some perturbations according to (2).

To identify the critical feature element with saliency map, the derivative weight ω for each element z in x is introduced and calculated based on the back propagation as follows:

$$\omega_z = \frac{\partial \hat{y}}{\partial x}|_z \tag{3}$$

where $\hat{y}$ is the corresponding predicted result. Each ω_z indicates the sensitivity of the corresponding z in x, in terms of its influence to the output $\hat{y}$.

Accordingly, we can investigate the sensitivity of each feature element z in x, and obtain the top-k critical elements according to the rank of ω_z. Then, the cross-entropy loss of attacker can be calculated via (1) based on the perturbation ε_i which is applied to those identified critical elements of the original x_i. Specifically, we set a trivial perturbation stride as 0.001 for ε_i in the maximum 20 episodes to estimate $\mathcal{L}_{\text{attack}}(f(X'; \theta^*), y)$ during the training process.

D. Hierarchical Node Selection Strategy

As discussed before, it is impossible for an attacker to compromise the whole network by modifying the whole feature set X for all nodes V. Considering the limited budget, the attacker usually chooses to compromise a subset of nodes with relatively smaller cost. Thus, an RWR-based algorithm on the GNN model is employed to capture the hierarchical structure feature in a weighted graph, which conducts the node selection task and generate a node set to be attacked with high priority.

RWR has been proved as an efficient way to calculate the weighting score in terms of the connections among networked nodes in a constructed graph model [36]. Basically, the RWR that measures the importance on the edge set E can be defined and expressed as follows:

$$HR^{(t+1)} = \lambda M \cdot HR^{(t)} + (1 - \lambda) HR^0 \tag{4}$$

where λ ranging from 0 to 1 is a damping coefficient for the random navigation during the iteration. HR^t denotes a score vector in terms of the feature importance at the step t. Particularly, $HR^0 = [0, \ldots, 1, \ldots, 0]$ is the initial vector when starting the RWR, in which the element of value "1" denotes that the corresponding node v_i is selected as a target of attack at the beginning. The transfer matrix $M \in \mathbb{R}^{P \times P}$ stores the probability of each node to transfer to the others.

Specifically, each $m_{ij} \in M$ can be initially measured according to the outlink d_i of v_i, as the probability of transfer from v_i to v_j, which can be calculated as follows:

$$m_{ij} = \begin{cases} 1/d_i, \text{ if } \exists e_{ij} \in E, \text{ or } i = j \\ 0, \quad \text{otherwise} \end{cases} \tag{5}$$

where d_i is quantified by the outlink of node v_i.

To measure the hierarchical feature of each node, A level-threshold τ is introduced to empirically evaluate d_i for each v_i in the network, which can measure and identify the level of each node as exemplified in Fig. 1. Given a randomly selected node v_i as an initial node, it will iteratively transmit to its neighborhood node based on M by calculating the corresponding λm_{ij}, while it may transmit back to itself with a probability of $(1 - \lambda) HR^0$.

Furthermore, the overall loss change is considered and investigated in terms of the update of transfer matrix M when we choose to attack different nodes with a limited budget. In particular, we evaluate the loss based on x'_i when perturbing a node v_i in the graph. Thus, the loss change for a selected node x'_i can be defined and described as follows:

$$\Delta_i(x) = \mathcal{L}_{\text{attack}}(f(x'_i, y) - \mathcal{L}_{\text{attack}}(f(x_i, y)). \tag{6}$$

Specifically, we can measure this loss change based on the first-order Taylor approximation

Algorithm 1 RWR-Based Hierarchical Adversarial Example Generation

Input: Graph $G = (V, E, D, X)$, ground truth class label y
Output: A set of adversarial examples generated according to the top-n selected nodes

Initialize HR^0, Max Iteration π, Threshold δ_{rwr}, δ_{loss}
for each e_{ij} in M **do**
  Initialize m_{ij} via Eq. (5)
end for
for $t = 1$ to π **do**
  Compute $HR^{(t+1)} = \lambda M \cdot HR^{(t)} + (1-\lambda)HR^0$
  Compute error $e^{(t)} = HR^{(t+1)} - HR^{(t)}$
  if $e^{(t)} < \delta_{rwr}$: **break**
  Compute loss change $\Delta_i(x)$ via Eq. (6) for the temporal top-n nodes in M and update M
end for
Select top-n nodes based on their ranking scores
for each v_i in the selected top-n nodes **do**
  Compute critical features for v_i via Eq. (3)
  Compute loss $\mathcal{L}_{attack}(f(X';\theta^*), y)$ via Eq. (1)
  While $\mathcal{L}_{attack}(f(X';\theta^*), y) < \delta_{loss}$ **do**
    Generate adversarial examples x_i' by adding perturbation to v_i via Eq. (2)
end for
return adversarial examples $\{x_i'|_{i\in\{1,2,\ldots,N\}}\}$

$\widetilde{\Delta}_i(x) \triangleq \Delta_i(x) + \nabla\delta_i{}^T(x - x_i)$ [13], which can be related to the column sum of the ith column in the transfer matrix M, and further be used to update M during the training process.

The RWR-based hierarchical node selection strategy for adversarial example generation is shown in Algorithm 1. Given a typical IoT network represented by a graph model $G = (V, E, D, X)$ with the ground truth label set y, a set of adversarial examples $\{x_i'|_{i\in\{1,2,\ldots,N\}}\}$ can be generated based on the cooperation of the hierarchical node selection and critical feature identification. In particular, we initialize the transfer matrix M based on the outlink of a specific node v_i, and iteratively update M considering the corresponding loss changes of the nodes, so as to select a set of nodes with high attack priority. Perturbations are added to the selected nodes to generate adversarial examples when the loss criterion described in (1) is satisfied. Finally, we can efficiently select a set of more vulnerable nodes and attack the GNN-based NIDS using the generated adversarial examples.

V. Evaluations and Discussion

In this section, we evaluate and compare the proposed HAA generation method with several baseline methods. Experiments are designed and conducted based on an open-source IoT data set, to demonstrate the effectiveness and usefulness of our method.

A. Data Set

An open data set UNSW-SOSR2019, which is collected by the security laboratory at the University of New South Wales using the tcpdump tool, is adopted for our evaluations. It collected packet traces of different kinds of attack and benign traffic from ten IoT devices in total [38]. Table I summarizes the overall data set, including the used IoT devices, and their corresponding training, testing, and attack sample sizes.

TABLE I
Data Set Description

Device	Device label	# Training Samples	# Test Samples	# Attack Samples
WeMo motion	WM	18750	55400	300
WeMo switch	WS	18750	55361	180
Samsung cam	SC	18750	55364	357
TP-Link plug	TP	18750	55372	178
Netatmo camera	NC	18750	55359	147
Chromecast Ultra	CU	18750	28730	252
Amazon Echo	AE	18750	28730	79
Phillips Hue bulb	PH	18750	28730	297
iHome plug	IH	18750	28730	30
LiFX bulb	LX	18750	28730	150

In addition, a set of preprocesses is conducted on the raw data set before training the model.

1) Construct the training set with 80% benign traffic and 20% attack traffic.
2) Remove the unreasonable traffics from the training set.
3) Build the graph model with the input data selected randomly from the training set.

B. Experiment Design

We evaluate the proposed HAA generation method on two typical GNN models, namely, the GCN [39] and JK-Net [40]. The layer of GCN is set to 3 and JK-Net is set to 7. The other hyper-parameters follow closely the setups in [33] and [37]. Specifically, to reflect the actual hierarchical network structure in IoT systems, three levels of the IoT nodes are configured as low level (outlinks: 0–5), medium level (outlinks: 5–10), and top level (outlinks: more than 10), with percentages 60%, 25%, and 15%, respectively, according to a typical real-world embedded Industrial IoT scenario. Experiments are conducted in a server with CentOS 8, GTX 1070, G39030 Duel Core, 16-G RAM, Python 3.6, and PyTorch 1.4, and all the results are generated with 40 repeated and independent trials.

The following three strategies are considered as the baseline methods when compromising the targeted GNN models.

1) *Improved Random Walk With Restart (iRWR) [36]:* This method took the time-varying features into consideration to find a navigation based on the importance score of nodes across network connections.
2) *Resistive Switching Memory (RSM) [41]:* This method used a cross-point array of RSM with a feedback configuration to solve the eigenvector calculation and webpage ranking tasks, which calculated the importance score similar to the PageRank strategy.
3) *Greedily Corrected Random Walk (GCRW) [13]:* This method generated black-box attacks based on analyzing the connection between the backward propagation of GNNs and random walks, which calculated the importance score in a greedy correction procedure.

C. Attack Effectiveness Evaluation

We evaluate the proposed method and compare its attack effectiveness with the mentioned three baseline methods, by

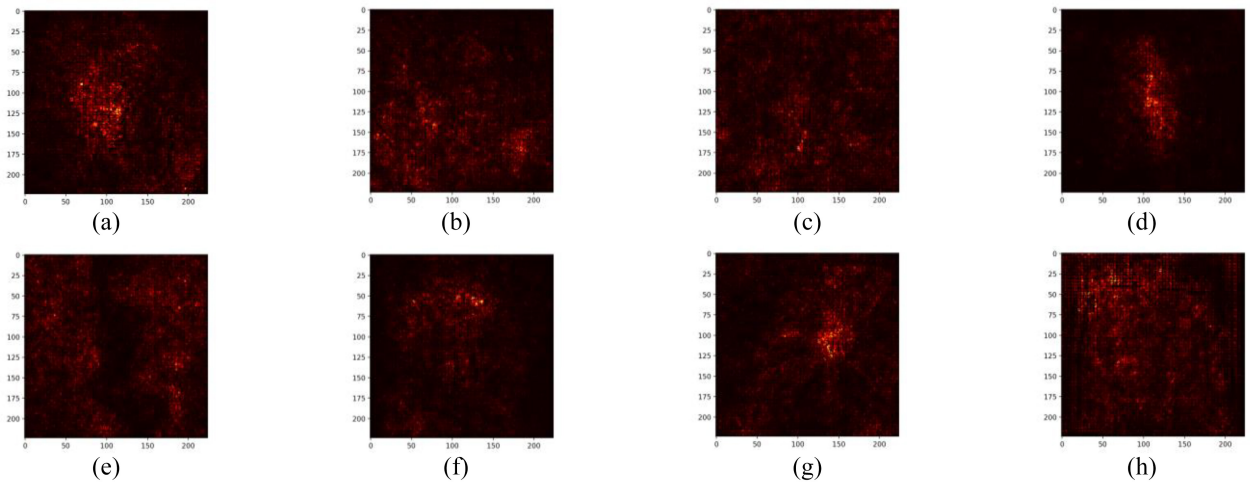


Fig. 3. Feature saliency map generated for different IoT devices. (a) WeMo motion. (b) Samsung cam. (c) TP-Link plug. (d) Netatmo camera. (e) Chromecast Ultra. (f) Amazon Echo. (g) Phillips Hue bulb. (h) iHome plug.

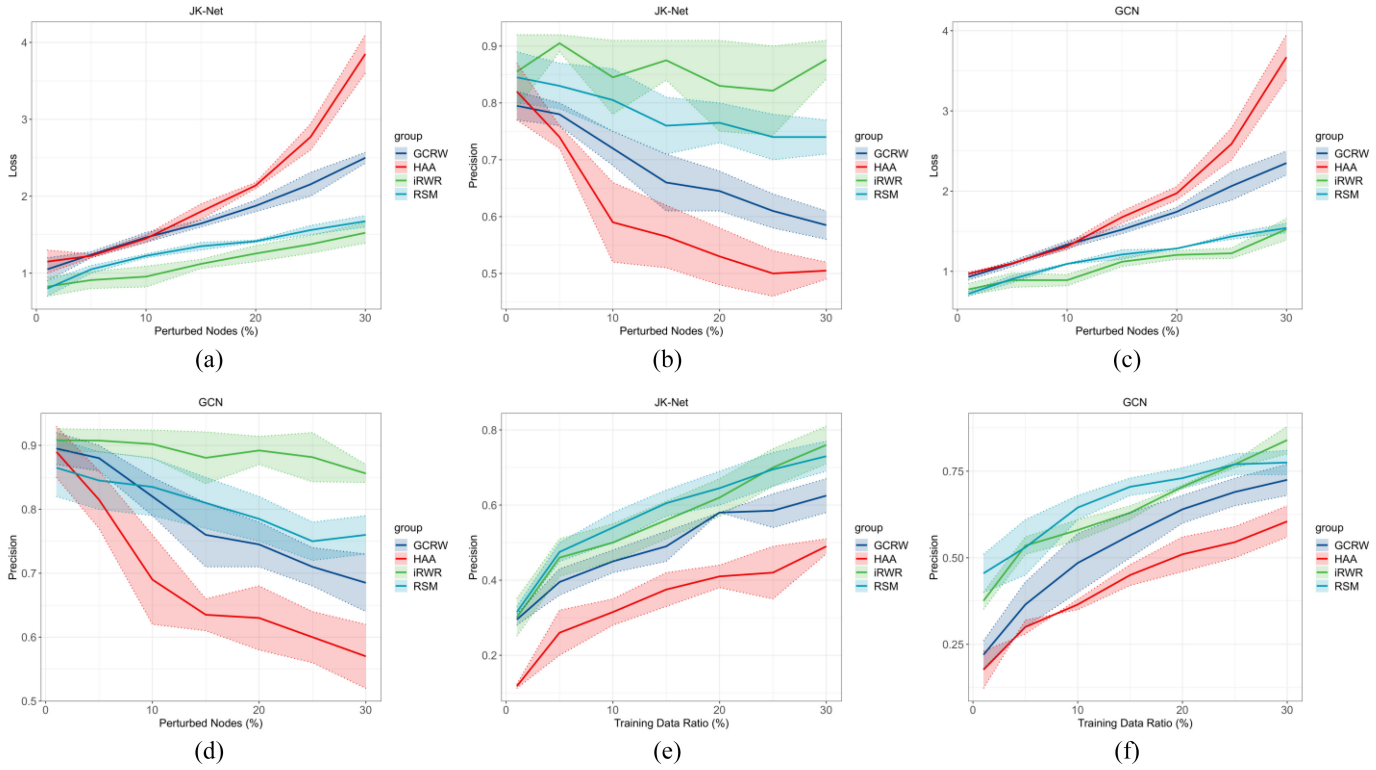


Fig. 4. Performance degradation in loss and classification precision in JK-Net and GCN with varying percentages of perturbed nodes and training data. (a) Loss with varying perturbation in JK-Net. (b) Precision with varying perturbation in JK-Net. (c) Loss with varying perturbation in GCN. (d) Precision with varying perturbation in GCN. (e) Precision with varying training data ratio in JK-Net. (f) Precision with varying training data ratio in GCN.

measuring the classification performance degradation based on two different GNN models in IoT environments.

First, we investigate the critical feature identification during the adversarial example generation process. The saliency map is utilized to illustrate the critical features of different IoT devices. Fig. 3 shows a set of generated feature saliency maps according to eight network traffic classes, after principal component analysis (PCA) dimension reduction.

Furthermore, we investigate attack effectiveness for all the methods. We first evaluate how these methods perform with different levels of perturbation. The performance of the GNN models is evaluated based on its classification loss and precision. A method that has a larger impact (i.e., large performance degradation) to the GNN models is expected to result in a higher loss and a lower precision. Fig. 4 presents the loss and precision results for JK-Net (Fig. 4(a) and (b)) and GCN (Fig. 4(c) and (d)), respectively, under varying % of the perturbed nodes from 2% to 30%.

In Fig. 4(a) and (b), the expected trends can be observed, where higher loss and greater precision degradation are

resulted when more nodes are perturbed (i.e., higher level of perturbation). Referring to Fig. 4(a), it can be found that the proposed HAA results in the highest loss in the JK-Net model in comparison with the other three methods. In addition, instead of having a linear increment in loss with respect to the increasing level of perturbation, the HAA generation method demonstrates a more exponential increase in loss. This indicates that the method is more effective when the number of perturbed nodes increases. Similar results are also reflected in Fig. 4(b), where both iRWR and RSM do not demonstrate very effective attack strength, and hence the precision of the GNN model remains reasonably well. In comparison, HAA is able to reduce the classification precision to close to 0.5 at 30% perturbation, which is a significant reduction and impact on the model performance. Overall, the proposed HAA method outperforms the other three methods and is able to achieve a 42.5% reduction in classification precision between 2% and 30% perturbed nodes.

Fig. 4(c) and (d) present the loss and classification precision for the GCN model, respectively, under the four attack models. The results follow the general trend, where higher loss and greater precision degradation will be achieved when more nodes are perturbed, although it can be observed that the GCN model offers a slightly better performance under our targeted scenarios. Similar to the results for JK-Net, the performance degradation in GCN also exhibit similar behavior. The HAA can achieve the best loss increment and classification precision reduction. Overall, the proposed method has achieved more than 30% reduction for classification precision in the GCN model.

We go further to analyze the effectiveness of the attack methods under varying sizes of training data from 2% to 30% of the original training data set. Referring to Fig. 4(e) and (f), which show the model performance with varying sizes of training data set for JK-Net and GCN, respectively. The general trend can be observed, where more training data will result in better GNN model performance. It can also be observed that, with 30% training data, the model performance has already reached to a comparable level to the complete training data set in the cases of iRWR and RSM. This clearly indicates that the attack strength of iRWR and RSM is not very strong. With GCRW, the model performance is also reaching close to 0.76, whereas in the case of HAA, the classification precision remains to be close to or lower than 0.6 when up to 30% of the training data is used. This gives a strong indication of the attack strength achieved by the proposed HAA method, which performs more consistently with varying sizes of training data.

VI. Conclusion

Advanced IoT networks and systems are growing at an unforeseen rate, reaching every corner of our cities and countries, to collect useful data, and to offer intelligent services. Considering the amount of data collected and processed by modern IoT systems, it is of critical importance to make sure that those systems are secure and not to be misused for any malicious purposes. To address this issue, tremendous amount of research effort is devoted to design robust NIDS to ensure the security of IoT systems. However, existing NIDS approaches all suffer from the fact that there is only a limited amount of very imbalanced training data, which leads to the vulnerability against unknown types of malicious attack.

In this article, we introduced an HAA generation method, targeting the state-of-the-art GNN-based NIDS in black-box attack scenarios. Specifically, we presented an integrated framework for the level-aware black-box adversarial attack strategy, which could generate adversarial examples based on the constructed shadow GNN model with a limited budget. The saliency map technique was utilized to facilitate the generation mechanism, based on which we could effectively identify those critical feature elements, so as to modify them with minimal perturbations. The RWR algorithm was employed to realize the hierarchical node selection, in which both structural features and overall loss changes within the targeted IoT network were considered to improve the transfer matrix, so as to efficiently select a set of more vulnerable nodes to attack the GNN-based NIDS using the generated adversarial examples. Evaluations were conducted using the open-source data set UNSW-SOSR2019. The results compared with three baseline methods demonstrate the ability of the proposed method in reducing the classification precision by more than 30% in two state-of-the-art GNN models, GCN and JK-Net, respectively.

In the future, we will go further to study more efficient and effective adversarial attack strategy. More evaluations in different IoT network application scenarios will be investigated to improve the adaptability of our method.

References

[1] Z. Cai and Z. He, "Trading private range counting over big IoT data," in *Proc. 39th IEEE Int. Conf. Distrib. Comput. Syst.*, 2019, pp. 144–153.

[2] X. Zhou *et al.*, "Intelligent small object detection for digital twin in smart manufacturing with industrial cyber-physical systems," *IEEE Trans. Ind. Informat.*, vol. 18, no. 2, pp. 1377–1386, Feb. 2022.

[3] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 2, pp. 766–775, Apr./Jun. 2020.

[4] X. Zhou, X. Xu, W. Liang, Z. Zeng, and Z. Yan, "Deep-learning-enhanced multitarget detection for end–edge–cloud surveillance in smart IoT," *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12588–12596, Aug. 2021.

[5] K. Sha, T. A. Yang, W. Wei, and S. Davari, "A survey of edge computing-based designs for IoT security," *Digit. Commun. Netw.*, vol. 6, no. 2, pp. 195–202, 2020.

[6] A. Aldweesh, A. Derhab, and A. Z. Emam, "Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues," *Knowl. Based Syst.*, vol. 189, pp. 105–124, Feb. 2020.

[7] X. Zhou, X. Yang, J. Ma, and K. I.-K. Wang, "Energy efficient smart routing based on link correlation mining for wireless edge computing in IoT," *IEEE Internet Things J.*, early access, May 6, 2021, doi: 10.1109/JIOT.2021.3077937.

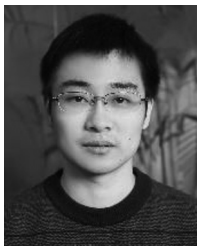
[8] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *J. Netw. Comput. Appl.*, vol. 84, pp. 25–37, Apr. 2017.

[9] X. Zhou, W. Liang, S. Shimizu, J. Ma, and Q. Jin, "Siamese neural network based few-shot learning for anomaly detection in industrial cyber-physical systems," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5790–5798, Aug. 2021.

[10] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 5, pp. 968–979, May 2020.

[11] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y. Pan, "Generative adversarial networks: A survey toward private and secure applications," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1–38, 2021.

[12] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.
[13] J. Ma, S. Ding, and Q. Mei, "Towards more practical adversarial attacks on graph neural networks," in *Proc. 34th Conf. Neural Info. Process. Syst. (NeurIPS 2020),* Vancouver, BC, Canada, Dec. 2020. [Online]. Available: https://nips.cc/Conferences/2020, NeurIPS 2020 is a Virtual-only Conference
[14] H. Qiu, T. Dong, T. Zhang, J. Lu, G. Memmi, and M. Qiu, "Adversarial attacks against network intrusion detection in IoT systems," *IEEE Internet Things J.*, vol. 8, no. 13, pp. 10327–10335, Jul. 2021.
[15] F. Zhou, Q. Yang, K. Zhang, G. Trajcevski, T. Zhong, and A. Khokhar, "Reinforced spatiotemporal attentive graph neural networks for traffic forecasting," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6414–6428, Jul. 2020.
[16] W. Zhang *et al.*, "Modeling IoT equipment with graph neural networks," *IEEE Access*, vol. 7, pp. 32754–32764, 2019.
[17] K. Rusek, J. Suárez-Varela, P. Almasan, P. Barlet-Ros, and A. Cabellos-Aparicio, "RouteNet: Leveraging graph neural networks for network modeling and optimization in SDN," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 10, pp. 2260–2270, Oct. 2020.
[18] Z. Guo and H. Wang, "A deep graph neural network-based mechanism for social recommendations," *IEEE Trans. Ind. Informat.*, vol. 17, no. 4, pp. 2776–2783, Apr. 2021.
[19] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4883–4894, Nov. 2020.
[20] D. Krleža and K. Fertalj, "Graph matching using hierarchical fuzzy graph neural networks," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 4, pp. 892–904, Aug. 2017.
[21] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "Graph neural networks for scalable radio resource management: Architecture design and theoretical analysis," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 101–115, Jan. 2021.
[22] F. Gama, A. G. Marques, G. Leus, and A. Ribeiro, "Convolutional neural network architectures for signals supported on graphs," *IEEE Trans. Signal Process.*, vol. 67, no. 4, pp. 1034–1049, Feb. 2019.
[23] D. Zhu, Y. Sun, H. Du, N. Cao, T. Baker, and G. Srivastava, "HUNA: A method of hierarchical unsupervised network alignment for IoT," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3201–3210, Mar. 2021.
[24] D. J. Miller, Z. Xiang, and G. Kesidis, "Adversarial learning targeting deep neural network classification: A comprehensive review of defenses against attacks," *Proc. IEEE*, vol. 108, no. 3, pp. 402–433, Mar. 2020.
[25] S. Krithivasan, S. Sen, and A. Raghunathan, "Sparsity turns adversarial: Energy and latency attacks on deep neural networks," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 11, pp. 4129–4141, Nov. 2020.
[26] T. Takahashi, "Indirect adversarial attacks via poisoning neighbors for graph convolutional networks," in *Proc. IEEE Int. Conf. Big Data*, Los Angeles, CA, USA, 2019, pp. 1395–1400.
[27] J. Yuan and Z. He, "Adversarial dual network learning with randomized image transform for restoring attacked images," *IEEE Access*, vol. 8, pp. 22617–22624, 2020.
[28] X. Lin *et al.*, "Exploratory adversarial attacks on graph neural networks," in *Proc. IEEE Int. Conf. Data Min. (ICDM)*, Sorrento, Italy, 2020, pp. 1136–1141.
[29] V. N. Ioannidis and G. B. Giannakis, "Defending graph convolutional networks against adversarial attacks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Barcelona, Spain, 2020, pp. 8469–8473.
[30] Y. Xu, B. Du, and L. Zhang, "Assessing the threat of adversarial examples on deep neural networks for remote sensing scene classification: Attacks and defenses," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 2, pp. 1604–1617, Feb. 2021.
[31] G. Apruzzese, M. Andreolini, M. Marchetti, A. Venturi, and M. Colajanni, "Deep reinforcement adversarial learning against botnet evasion attacks," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 4, pp. 1975–1987, Dec. 2020.
[32] D. Li and Q. Li, "Adversarial deep ensemble: Evasion attacks and defenses for malware detection," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3886–3900, 2020.
[33] Y. E. Sagduyu, Y. Shi, and T. Erpek, "Adversarial deep learning for over-the-air spectrum poisoning attacks," *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 306–319, Feb. 2021.
[34] X. Zhang and M. Zitnik, "GNNGuard: Defending graph neural networks against adversarial attacks," 2020, *arXiv:2006.08149*.
[35] D. Zügner, A. Akbarnejad, and S. Günnemann, "Adversarial attacks on neural networks for graph data," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Min. (KDD)*, London, U.K., Aug. 2018, pp. 2847–2856.
[36] X. Zhou, W. Liang, K. I.-K. Wang, R. Huang, and Q. Jin, "Academic influence aware and multidimensional network analysis for research collaboration navigation based on scholarly big data," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 1, pp. 246–257, Jan.–Mar. 2021.
[37] M. Ahmadi, M. Hajabdollahi, N. Karimi, and S. Samavi, "Context-aware saliency map generation using semantic segmentation," in *Proc. Iranian Conf. Elect. Eng. (ICEE)*, Mashhad, Iran, 2018, pp. 616–620.
[38] A. Hamza, H. H. Gharakheili, T. Benson, and V. Sivaraman, "Detecting volumetric attacks on loT devices via SDN-based monitoring of MUD activity," in *Proc. ACM SOSR*, San Jose, CA, USA, Apr 2019, pp. 36–48.
[39] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.
[40] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-I. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2018, pp. 5453–5462.
[41] Z. Sun, E. Ambrosi, G. Pedretti, A. Bricalli, and D. Ielmini, "In-memory PageRank accelerator with a cross-point array of resistive memories," *IEEE Trans. Electron Devices*, vol. 67, no. 4, pp. 1466–1470, Apr. 2020.



Xiaokang Zhou (Member, IEEE) received the Ph.D. degree in human sciences from Waseda University, Tokyo, Japan, in 2014.

He is currently an Associate Professor with the Faculty of Data Science, Shiga University, Hikone, Japan. From 2012 to 2015, he was a Research Associate with the Faculty of Human Sciences, Waseda University. He has been working as a Visiting Researcher with the RIKEN Center for Advanced Intelligence Project, RIKEN, Tokyo, since 2017. He has been engaged in interdisciplinary research works in the fields of computer science and engineering, information systems, and social and human informatics. His recent research interests include ubiquitous computing, big data, machine learning, behavior and cognitive informatics, cyber-physical-social systems, cyber intelligence, and security.

Dr. Zhou is a member of IEEE CS, and ACM, USA, IPSJ, and JSAI, Japan, and CCF, China.

Wei Liang (Member, IEEE) received the M.S. and Ph.D. degrees in computer science from Central South University, Changsha, China, in 2005 and 2016, respectively.

From 2014 to 2015, he was a Researcher with the Department of Human Informatics and Cognitive Sciences, Waseda University, Tokyo, Japan. He is currently working with the Base of International Science and Technology Innovation and Cooperation on Big Data Technology and Management, Hunan University of Technology and Business, Changsha. He has published more than 20 papers at various conferences and journals. His research interests include information retrieval, data mining, and artificial intelligence.

Dr. Liang is a member of IEEE CS and CCF, China.

Weimin Li (Member, IEEE) received the Ph.D. degree in control theory and control engineering from Donghua University, Shanghai, China, in 2008.

He is a Professor with the School of Computer Engineering and Science, Shanghai University, Shanghai, China. He was a JSPS Research Fellow with the Department of Human Informatics and Cognitive Sciences, Waseda University, Tokyo, Japan, from 2012 to 2013. He was a Visiting Scholar with the Department of Computer Science, University of California at Santa Barbara, Santa Barbara, CA, USA, from 2015 to 2016. He has been involved in the extensively research works in the fields of computer science, service computing, group behavior, and database technology. His current research interests include social computing, data mining and analytics, group behavior modeling and simulating, and service recommendations.

Ke Yan (Member, IEEE) received the bachelor's and Ph.D. degrees in computer science from the School of Computing, National University of Singapore (NUS), Singapore, in 2006 and 2012, respectively.

He is currently an Assistant Professor with NUS. He has published more than 70 full length papers with highly ranked conferences and journals, including Association for the Advancement of Artificial Intelligence, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON SUSTAINABLE ENERGY, IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS: SYSTEMS, and *Applied Energy*. He is actively engaged in cross-discipline research fields, including machine learning, artificial intelligence, cyber intelligence, applied mathematics, sustainability, and applied energy.

Shohei Shimizu received the Ph.D. degree in engineering (statistical science) from Osaka University, Suita, Japan, in 2006.

He is a Professor with the Faculty of Data Science, Shiga University, Hikone, Japan, and leads the Causal Inference Team, RIKEN Center for Advanced Intelligence Project, RIKEN, Tokyo, Japan. His research interests include statistical methodologies for learning data generating processes such as structural equation modeling and independent component analysis and their application to causal inference.

Prof. Shimizu received the Hayashi Chikio Award (the Excellence Award) from the Behaviormetric Society in 2016. He has been a Coordinating Editor of *Behaviormetrika* (Springer) since 2016.

Kevin I-Kai Wang (Member, IEEE) received the Bachelor of Engineering (Hons.) degree in computer systems engineering and the Ph.D. degree in electrical and electronics engineering from the Department of Electrical and Computer Engineering, University of Auckland, Auckland, New Zealand, in 2004 and 2009, respectively.

He is currently a Senior Lecturer with the Department of Electrical and Computer Engineering, University of Auckland. He was also a Research Engineer designing commercial home automation systems and traffic sensing systems from 2009 to 2011. His current research interests include wireless sensor network-based ambient intelligence, pervasive healthcare systems, human activity recognition, behavior data analytics, and bio-cybernetic systems.