Variational LSTM Enhanced Anomaly Detection for Industrial Big Data

Xiaokang Zhou, *Member, IEEE*, Yiyong Hu, *Member, IEEE*, Wei Liang, *Member, IEEE*, Jianhua Ma, *Member, IEEE*, and Qun Jin, *Senior Member, IEEE*

***Abstract*—With the increasing population of Industry 4.0, industrial big data (IBD) has become a hotly discussed topic in digital and intelligent industry field. The security problem existing in the signal processing on large scale of data stream is still a challenge issue in industrial internet of things, especially when dealing with the high-dimensional anomaly detection for intelligent industrial application. In this article, to mitigate the inconsistency between dimensionality reduction and feature retention in imbalanced IBD, we propose a variational long short-term memory (VLSTM) learning model for intelligent anomaly detection based on reconstructed feature representation. An encoder–decoder neural network associated with a variational reparameterization scheme is designed to learn the low-dimensional feature representation from high-dimensional raw data. Three loss functions are defined and quantified to constrain the reconstructed hidden variable into a more explicit and meaningful form. A lightweight estimation network is then fed with the refined feature representation to identify anomalies in IBD. Experiments using a public IBD dataset named UNSW-NB15 demonstrate that the proposed VLSTM model can efficiently cope with imbalance and high-dimensional issues, and significantly improve the accuracy and reduce the false rate in anomaly detection for IBD according to F1, area under curve (AUC), and false alarm rate (FAR).**

***Index Terms*—Anomaly detection, feature representation, industrial big data (IBD), long short-term memory (LSTM), variational Bayes.**

Manuscript received March 13, 2020; revised June 5, 2020 and August 10, 2020; accepted August 24, 2020. Date of publication September 11, 2020; date of current version February 22, 2021. This work was supported in part by the National Key R&D Program of China under Grant 2017YFE0117500, in part by the Natural Science Foundation of Hunan Province of China under Grant 2019JJ40150, and National Natural Science Foundation of China under Grant 61702183, in part by the Key R&D Project in the industrial field funded by Hunan Provincial Science & Technology Department under Grant 2019GK2131, and in part by and the Key Project of Hunan Provincial Education Department under Grant 17A113. Paper no. TII-20-1295. *(Corresponding author: Wei Liang.)*

Xiaokang Zhou is 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).

Yiyong Hu and Wei Liang are with the Key Laboratory of Hunan Province for New Retail Virtual Reality Technology, Hunan University of Technology and Business, Changsha 410205, China (e-mail: aminkira2019@gmail.com; weiliang@csu.edu.cn).

Jianhua Ma is with the Faculty of Computer and Information Sciences, Hosei University, Chiyoda-ku 102-8160, Japan (e-mail: jianhua@hosei.ac.jp).

Qun Jin is with the Faculty of Human Sciences, Waseda University, Tokorozawa 359-1192, Japan (e-mail: jin@waseda.jp).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2020.3022432

I. Introduction

With the rapid development of Industry 4.0, more and more industrial applications, empowered by intelligent and real-time signal processing, are connected interactively, due to the increasingly wide use of wireless network technology with diversified smart devices in industrial Internet of Things (IIoT). The increase of devices and applications in IIoT leads to a large scare of real-time data with more complexity generated across industrial cyber–physical systems, which can be called as industrial big data (IBD). It becomes a critical issue to protect the infrastructure and network security for core tasks in IIoT. As an indispensable technology in IIoT security, network intrusion detection system is usually deployed as a software mechanism to monitor and detect intrusion events or anomalies across the whole industrial network, which can be categorized into the signature-based and anomaly based intrusion detection. Recently, anomaly detection has drawn increasing attentions, due to its ability in detecting novel attacks from high-dimensional IBD across a variety of IIoT sensors [1].

To enhance the accuracy of anomaly detection when dealing with IBD, machine learning and deep learning techniques have been employed for both host and network based systems. However, it is still a troublesome task to carry out the reliable anomaly detection result from large amounts of high-dimensional data in IIoT. It would be even worse for conventional classification methods to extract meaningful features from the imbalanced input data, especially when positive samples become extremely sparse in IBD environments. To mitigate the computation complexity caused by high-dimensional issues, a two-stage implementation, including the dimension reduction and feature extraction [2], is usually used in the potential low-dimensional space. Different kinds of auto encoding techniques have been explored in intrusion detection and achieved great success in reencoding high-dimensional features to lower dimension features [3]. Although these methods could improve the accuracy of anomaly detection to a certain extent, the false alarm rate (FAR) is still an unsolved issue especially when facing the imbalanced dataset.

One limitation of conventional deep learning techniques in handling low FAR issue is, it is difficult to deduce whether the critical information related to the network intrusions can

be preserved after dimension reduction, due to their restrained unidirectional encoding structure. Auto encoder (AE) technology [4] is investigated broadly to reconstruct the original input data. Although it can achieve a relatively good tradeoff between dimension reduction and feature retention from the original raw data, AE scheme cannot ensure all the critical features can be retained during the dimension reduction process. This is because AE may be too aggressive when handling the imbalanced input data. Challenges in handling anomaly detection for IBD can be mainly summarized as: first, it is difficult to achieve a well-balanced result that cannot only keep as many key features as possible for anomaly detections, but also effectively compress the raw data; second, it is hard to obtain prior knowledge regarding the subsequent feature extraction task when performing the previous dimension reduction task, situations will become even worse in IBD scenarios. Therefore, it is essential to design an adaptive strategy to overcome the aforementioned limitations in the state-of-art approaches, and optimize the balance between dimension reduction and feature extraction properly for anomaly detection in IBD environments.

In this article, a novel anomaly detection model based on variational long short-term memory (VLSTM) is designed to deal with the imbalanced and high-dimensional issues in IBD. Specifically, an encoder–decoder neural network associated with a variational reparameterization scheme is designed to learn the low-dimensional feature representation while avoiding the loss of key information. The hidden variable is constructed using variational Bayes and refined based on three loss functions. A lightweight estimation network is then built to provide classifications based on the refined feature representation for anomaly detection. Major contributions of this article are concluded as follows.

1) A framework of VLSTM is newly designed, in which a compression network with a variational reparameterization, and an estimation network are constructed for low-dimensional feature representation.
2) Three loss functions are defined and seamlessly integrated together to constrain the reconstructed hidden variable, which can efficiently retain the critical features during the dimension reduction process.
3) An intelligent learning algorithm is developed for anomaly detection based on the VLSTM model, which can be applied to handle the imbalanced and high-dimensional issues for IBD.

The rest of this article is organized as follows. Section II presents an overview of related works. In Section III, the basic framework of the proposed VLSTM is addressed. In Section IV, we discuss the detailed implementation of intelligent anomaly detection via the VLSTM model. Section V demonstrates the experiment and evaluation results based on an open dataset. Finally Section VI concludes this article.

II. Related Work

In this section, several related issues, including studies on intrusion detection system, and machine learning methods used in anomaly detection, are reviewed respectively.

A. Issues on Intrusion Detection System

Network intrusion detection has been investigated extensively for cyber security in wireless network [5], 6], IoT [7], [8], cloud [9], and blockchain systems [10]. Traditional firewall system and intrusion detection are incompatible with the new Industry 4.0 environment, which may be reflected in the volume, accuracy, diversity, dynamics, low-frequency attacks, and adaptability issues [11].

Anthi *et al.* [12] concluded the rule/event/signature-based intrusion detection systems in IoT networks. Midi *et al.* [13] proposed a knowledge-driven intrusion detection system called Kalis, which leveraged the collected knowledge to dynamically configure the monitored network in a rule-based intrusion detection system. Pongle and Chavan [14] proposed a hybrid network intrusion detection system, which aimed at detecting routing attacks in event-based intrusion detection systems. Contrastively, the signature-based intrusion detection system has drawn more attentions during the current years. Stephen and Arockiam [15] designed a routing protocol with low power and lossy network, based on which a centralized hybrid approach was explored to detect Sybil attacks in IoT. However, they only tested the sinkhole, spoofed information, and selective forwarding attacks.

B. Machine Learning in Anomaly Detection

Generally, the signature-based and anomaly based detections are viewed as the two major problems investigated in intrusion detection systems [16]. Although anomaly detection has attracted increasing attentions in both research and industrial areas due to its capability in detecting unknown attacks, high dimensions of the raw data, and high value of FAR are still the major problems [17]. Different kinds of machine learning algorithms and their combinations were applied to resolve these issues. For example, Doshi *et al.* [18] employed several machine learning methods to detect distributed denial of service (DoS) attacks in IoT networks. They utilized network behaviors for the feature selection, and achieved a high accuracy of distributed DoS detection. Zuo *et al.* [19] built a learning-based anomaly detection framework, which combined a logging-tracing model and a temporal-spatial model to detect the network traffic intrusion.

Recently, deep learning techniques have achieved remarkable results and become one kind of successful approaches in intrusion detection system. Zhao *et al.* [20] surveyed a series of deep learning approaches to security monitoring. They compared a couple of conventional machine learning techniques with four typical deep learning schemes. The review result demonstrated the usability of deep learning methods for cyber security protection. Particularly, Ma *et al.* [21] implemented deep neural networks into the KDD99 dataset to detect intrusion behaviors. Brun *et al.* [22] developed a dense random neural network to detect cyberattacks. Potluri and Diedrich [23] proposed a deep neural network with three hidden layers, based on which they found that the classification results obtained by fewer intrusive classes were better than those with more classes. Tian *et al.* [24] implemented a web attack detection system based on distributed edge devices, in which multiple concurrent

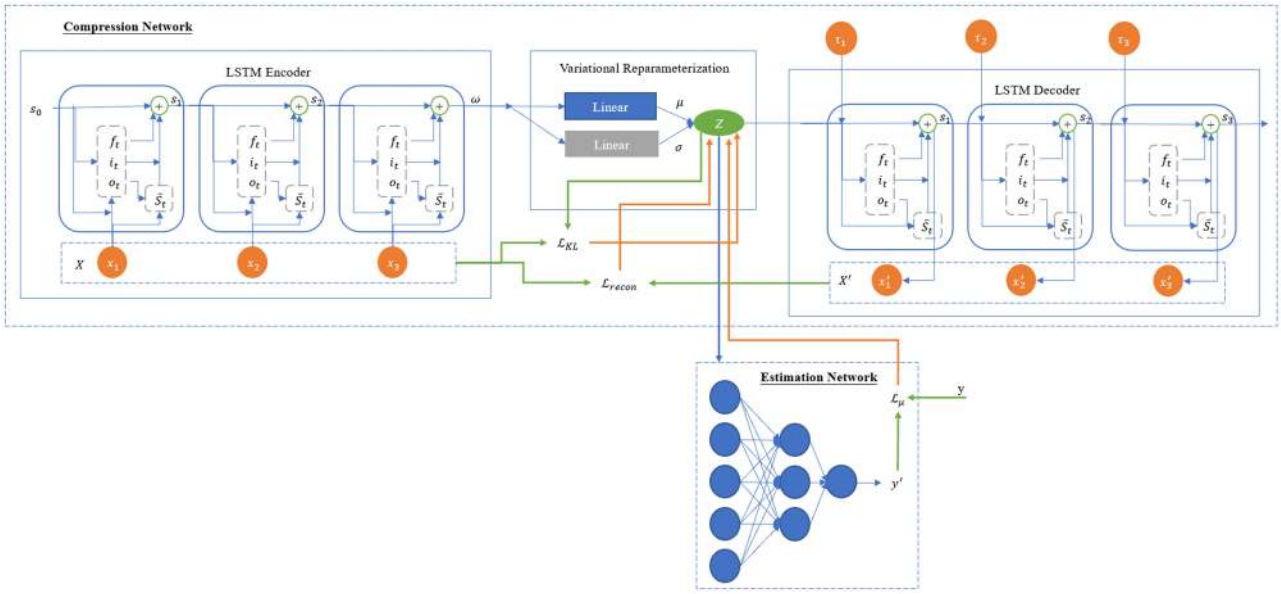


Fig. 1. Generic framework of VLSTM.

learning models were used to improve the system stability and performance.

III. Modeling of VLSTM

In this section, following the formal definition of anomaly detection problem in IBD, we introduce the basic structure of the proposed VLSTM model with detailed modules.

A. Problem Definition

Given an input set $D = \{X^{(1)}, X^{(2)}, \ldots, X^{(k)}\}$ representing k labeled network traffic data in IBD, which consists of p normal samples and q anomaly samples, $p+q = k$. In particular, we assume $p \gg q$, to describe the anomaly detection scenario with an imbalanced dataset. $Y = \{y^{(1)}, y^{(2)}, \ldots, y^{(k)}\}$ is a set of corresponding labels for D. The problem investigated in this article is to identify the category (i.e., one specific type of network attack, or just the normal traffic data) described as Y'. In addition, each sample $X^{(i)}$ is a n-dimensional feature vector, and can be described as $X^{(i)} = \langle x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)} \rangle$. It is noted that $X^{(i)}$ is usually a high-dimensional feature vector from IBD environments, thus one challenge is to represent the raw data into a lower feature space but avoid the loss of key features.

To tackle the high-dimensional anomaly detection problem with an imbalanced dataset, a VLSTM model, which basically consists of a compression network and an estimation network, is designed to pursue a well-balanced tradeoff between raw data compression and critical features retention in IBD. A generic framework of the proposed VLSTM model is shown in Fig. 1.

As shown in Fig. 1, the compression network is composed of the core modules of the VLSTM model, including the LSTM encoder module, variational reparameterization module, and LSTM decoder module. On the other hand, the estimation network is designed to obtain the classification results based on the input network traffic data with the refined feature representation.

B. VLSTM Framework

Basically, the compression network is employed to mitigate the complexity of high-dimensional input data, while retain adequate information to ensure the detection accuracy. As shown in Fig. 1, motivated by the conventional AE, the LSTM encoder is designed to compress the dimensionality of input data which is represented as a feature vector $\langle x_1, x_2, x_3 \rangle$. Given an input data X, the general expression of output ω of the LSTM encoder can be described as follows:

$$\omega = h(X; \theta_e) \tag{1}$$

where θ_e indicates the set of parameters used in LSTM encoder, $h(*)$ represents the LSTM encoding function. Specifically, f_t, i_t, o_t, and s_t shown in Fig. 1 stand for the forget gate, input gate, output gate, and memory cell for the state persistence in LSTM, respectively.

Similar to AE, the unobservability of the obtained hidden variables usually leads to the uncertainty of classification results when directly using the output of LSTM encoder. Therefore, the variational reparameterization module is designed to work with the LSTM encoder to reconstruct an explicit hidden variable via variational Bayes, so as to ensure the obtained hidden variable can retain critical features of anomalies as much as possible from the imbalanced dataset. In this module, the LSTM encoder result ω is used as the input to generate a low-dimensional hidden variable represented as Z. The general formulation can be expressed as follows:

$$Z = v(\omega; \theta_v) \tag{2}$$

where θ_v indicates the set of parameters used in variational Bayes function $v(*)$.

Typically, the compression process may loss some critical information or bring in some noise data. To evaluate the compression loss, The LSTM decoder is responsible for transforming

the hidden variable Z back to the reconstructed feature vector $\langle x'_1, x'_2, x'_3\rangle$. Both Z and $\langle \tau_1, \tau_2, \tau_3\rangle$ are the input to LSTM decoder, which stand for the initial state and initial input for the LSTM network, respectively. $\langle \tau_1, \tau_2, \tau_3\rangle$ is generated by a random initialization and then continuously updated via training process. Additionally, X' is defined to describe the reconstructed feature vector of Z. It is noted that the dimension of X' should be the same as that of X. The general expression of X' can be formulated as follows:

$$X' = g(Z; \theta_d) \tag{3}$$

where θ_d indicates the set of parameters used in LSTM decoder, $g(*)$ represents the LSTM decoding function.

The estimation network, which is constructed based on a fully connected deep neural network, is designed to identify whether the input data can be classified as the normal traffic data or one specific type of network attack. The input of estimation network is from the low-dimensional hidden variable Z calculated by the compression network. y' represents the output of estimated network, which can be viewed as the final classification result based on our proposed VLSTM model.

In general, the main functions of the VLSTM model can be concluded as: first, obtain a low-dimensional output ω from the high-dimensional raw input data via the LSTM encoder. Second, construct a hidden variable Z using variational Bayes and refine it from the LSTM decoder, variational reparameterization, and classification results, to cope with the imbalanced data. Finally, provide the network traffic classification via the estimation network based on the more explicit and meaningful hidden variable Z for anomaly detection.

IV. Anomaly Detection Based on VLSTM

In this section, we discuss the reconstruction of hidden variable with three loss functions, and develop a VLSTM-based algorithm for intelligent anomaly detection in IBD environments.

A. Hidden Variable Reconstruction via Variational Bayes

The variational Bayes method in variational auto encoder [4] is used to process the hidden variable Z since it can facilitate the reconstruction process based on unobservable variable. Usually sampling methods are utilized to obtain approximation solution for the marginal likelihood function $\int p(\omega)p(X|\omega)d\omega$ of AE. However, the computation cost is very expensive even if a modern sampling method (e.g., Markov Monte Carlo) is applied on a very small dataset. Therefore, a stochastic gradient variational Bayes scheme is developed to solve this problem by approximating the true posterior distribution $p(\omega|X)$ with the approximation $q(Z|X)$ and optimizing the lower bound of the log-likelihood, in which we have $Z \sim q(Z|X)$ and $\omega \sim p(\omega|X)$.

Specifically, the log-likelihood of an input $X^{(i)}$ can be calculated as the sum of the Kullback–Leibler (KL) divergence term D_{KL} based on $p(\omega|X^{(i)})$ and $q(Z|X^{(i)})$, and the lower bound of probability density of $X^{(i)}$, which can be expressed as follows:

$$\log p\left(X^{(i)}\right) = D_{\mathrm{KL}}(q(Z|X^{(i)})||p(\omega|X^{(i)})) + L\left(\theta; X^{(i)}\right) \tag{4}$$

Since KL divergence is nonnegative, $L(\theta; X^{(i)})$ can be deduced as the lower bound of log-likelihood by Eq. (5).

$$L\left(\theta; X^{(i)}\right) = E_{Z\sim q(Z|X^{(i)})}\left[\log p_\theta\left(X^{(i)}|Z\right)\right] - D_{\mathrm{KL}}[q(Z|X^{(i)}||p(\omega|X^{(i)}))] \tag{5}$$

where $E_{Z\sim q(Z|X^{(i)})}[\log p_\theta(X^{(i)}|Z)]$ is defined as the reconstruction term, which represents the approximation between the distribution of Z and the distribution of $X^{(i)}$. $D_{\mathrm{KL}}[q(Z|X^{(i)}||p(\omega|X^{(i)}))]$ represents the approximation between $q(Z|X^{(i)})$ and $p(\omega|X^{(i)})$.

The gradient ascent is utilized to maximized the lower bound for $L(\theta; X^{(i)})$ according to the maximum likelihood function. As for gradient calculations for all the parameters in Eq. (5), the gradient of $q(Z|X^{(i)})$ can be obtained directly, but the gradient of $p(\omega|X^{(i)})$ cannot be directly computed. Therefore, the variational reparameterization is utilized to reconstruct the hidden variable Z. Specifically, a parameter $\varepsilon \sim N(0, 1)$, is introduced to obtain the gradient of $q(Z|X^{(i)})$, and Z can be reparameterized by $Z = \mu + \varepsilon \times \sigma$ because Z is a univariate Gaussian variable and $Z \sim N(\mu, \sigma^2)$. μ and σ are calculated by two different nonlinear neurons, which represent the mean vector and covariance vector of ω, respectively. Through this reparameterization process, a more reasonable and explicit hidden variable Z can be learned comparing with the conventional AE scheme.

B. Robust Constraint for Hidden Variable

During the optimization process, the learning model may be affected by several factors, and even bring in unnecessary variables in the adversarial competition. Therefore, three loss functions are introduced to constrain the hidden variable Z during the learning process, in order to ensure the distribution of the reconstructed hidden variable is consistent with that of the raw input data.

First, we design a reconstruction loss L_{recon} between $X^{(i)}$ and $X^{(i)'}$, to measure how much the hidden variable Z can retain the original input information, which is defined as follows:

$$\mathcal{L}^{(i)}_{recon} = -\sum_{j=1}^{n} p\left(x_j^{(i)}\right) \log q\left(x_j^{(i)'}\right) \tag{6}$$

where $x_j^{(i)}$ and $x_j^{(i)'}$ are the jth feature ($j \in [1, n]$) of $X^{(i)}$ and $X^{(i)'}$, respectively.

Then, to measure the classification loss between y and y' from the estimation network, the cross entropy of $y^{(i)}$ and $y^{(i)'}$ is defined as $\mathcal{L}^{(i)}_{\mu}$, which can be expressed as follows:

$$\mathcal{L}^{(i)}_{\mu} = -p\left(y^{(i)}\right) \log q\left(y^{(i)'}\right) \tag{7}$$

In addition, we investigate the divergence loss between Z and $X^{(i)}$. The $I(X^{(i)}, Z)$ is designed to described the mutual information of $X^{(i)}$ and Z. The larger value of $I(X^{(i)}, Z)$ will

indicate more complete feature information retained in Z. The detailed formulation is described as follows:

$$\begin{aligned} I\left(X^{(i)}, Z\right) &= E_{p\left(X^{(i)}, Z\right)}\left[\log p\left(X^{(i)}, Z\right)\right. \\ &\quad \left. - \log p\left(X^{(i)}\right) p\left(Z\right)\right] \\ &= E_{p\left(X^{(i)}, Z\right)}\left[D_{\mathrm{KL}}[p(Z|X^{(i)})||p\left(Z\right)]\right] \end{aligned} \tag{8}$$

It is observed that $I(X^{(i)}, Z)$ is estimated by D_{KL}. However, as discussed in Eq. (5), D_{KL} needs to be minimized when pursuing the maximum of $L(\theta; X^{(i)})$, which will decrease $I(X^{(i)}, Z)$ accordingly. Thus, the core issue is to find an equilibrium between D_{KL} and $I(X^{(i)}, Z)$, because D_{KL} affects $I(X^{(i)}, Z)$ in an adversarial manner.

Actually, the reconstructed term and KL divergence term in Eq. (5) are not independent to each other. Following [25], the lower bound of the maximum $I(X^{(i)}, Z)$ can be achieved by minimizing the reconstruction term. Based on these, the reconstruction term is used to adjust the equilibrium between D_{KL} and $I(X^{(i)}, Z)$, which can finally facilitate the learning of more explicit and meaningful hidden variable in the VLSTM model.

Accordingly, we define $\mathcal{L}_{kl}^{(i)}$ to denote the divergence loss between Z and $X^{(i)}$, which can be represented in an adversarial competition way as follows:

$$\mathcal{L}_{kl}^{(i)} = -L\left(\theta; X^{(i)}\right) \tag{9}$$

C. VLSTM Enhanced Anomaly Detection Algorithm

To handle the imbalanced and high-dimensional data, the reconstruction loss $\mathcal{L}_{recon}^{(i)}$, classification loss $\mathcal{L}_{\mu}^{(i)}$, and divergence loss $\mathcal{L}_{kl}^{(i)}$, are utilized together to constrain Z, and mitigate the influence of unnecessary variables during the optimization. The overall loss function $\mathcal{L}_{vlstm}^{(i)}$ for an input data $X^{(i)}$ can be elaborated as follows:

$$\mathcal{L}_{vlstm}^{(i)} = \mathcal{L}_{recon}^{(i)} + \mathcal{L}_{kl}^{(i)} + \mathcal{L}_{\mu}^{(i)} \tag{10}$$

More precisely, $\mathcal{L}_{\mu}^{(i)}$ is used to ensure Z to keep the essential features during the optimization process. $\mathcal{L}_{recon}^{(i)}$ is employed to provide Z with more meaningful features from $X^{(i)}$. $\mathcal{L}_{kl}^{(i)}$ is designed to retain more complete feature information for Z in an adversarial competition way.

The workflow of the VLSTM enhanced anomaly detection can be addressed as follows: first, normalize the feature vectors from samples, and construct the input sets D and Y. Second, reduce the dimension of input feature vectors using the LSTM encoder, and generate the hidden variable Z which will be continuously updated during the learning process. Third, reconstruct X' using the LSTM decoder, which should have the same dimension with input X. Finally, input the refined hidden variable Z into the estimation network to obtain the classification result for anomaly detection. The concrete algorithm is illustrated in Algorithm 1.

As described in the algorithm, the VLSTM model M is trained through T iterations. In each iteration, a batch of network traffic data with N samples are fed to the model to learn the parameters for the model. For each input sample $X^{(i)}$, the pseudocode between Line 4 and 9 illustrates the whole workflow mentioned above. The training process will terminate when the maximum iteration is reached or the loss $\mathcal{L}_{vlstm}^{(i)}$ is less than a given threshold δ. After that, the trained model M is ready for anomaly detection tasks.

TABLE I
UNSW-NB15 FEATURE DESCRIPTION

Category	Feature Name
Basic Feature	dur, proto, service, state, spkts, dpkts, sbytes, dbytes, rate, sttl, dttl, sload, dload, sloss, dloss
Content Feature	swin, dwin, stcpb, dtcpb, smean, dmean, trans_depth, res_bdy_len
Time Feature	sintpkt, dintpkt, sjit, djit, tcprtt, synack, ackdat
Extra Generated Feature	ct_srv_src, ct_state_ttl, ct_dst_ltm, ct_src_dport_ltm, ct_dst_sport_ltm, ct_dst_src_ltm, is_ftp_login, ct_ftp_cmd, ct_flw_http_mthd, ct_src_ltm, ct_srv_dst, is_sm_ips_ports

Algorithm 1: Anomaly Detection Algorithm Based on VLSTM.

Input: A set of input data $D = \{X^{(1)}, X^{(2)}, \ldots, X^{(k)}\}$ and the corresponding label $Y = \{y^{(1)}, y^{(2)}, \ldots, y^{(k)}\}$

Output: A trained anomaly detection model M

1: Initialize the model M
2: Initialize the iteration count T, batch size N, threshold δ
3: **for** $q = 1$ to T **do**
4: **for** each batch $\{X^{(i)}\}_{i=1}^{N}$ **do**
5: Transfer $X^{(i)}$ into ω via the LSTM Encoder by Eq. (1)
6: Obtain hidden variable Z by Eq. (2)
7: Input Z into the LSTM decoder to get the reconstructed $X^{(i)'}$
8: Predict $y^{(i)'}$ based on Z via the estimation network
9: Update M to minimize $\mathcal{L}_{vlstm}^{(i)}$ by Eq. (10)
10: **end for**
11: **if** $\mathcal{L}_{vlstm} < \delta$: **break**
12: **end for**
13: **return** M

V. EXPERIMENT AND ANALYSIS

In this section, experiments are designed and conducted to evaluate the VLSTM model with other similar works, to demonstrate the usefulness of our proposed method.

A. Data Set

To investigate the effectiveness of the proposed VLSTM model in IBD environments, an open dataset named UNSW-NB15 is used for comparison evaluations. UNSW-NB15 is generated by an Australian security laboratory using IXIA PerfectStrom tool, which combines the real normal network traffic

TABLE II
ATTACK TYPE AND THE CORRESPONDING DATASET DESCRIPTION FOR UNSW-NB15

Attack Name	Description	Training Set	Testing Set
Normal	Normal traffic	56000	37000
Backdoor	An attack that stealthily accesses to a program or system via bypassing normal authentication	1746	583
Analysis	An attack that penetrates web applications through ports, emails and web scripts	2000	677
Fuzzers	An attack that attempts to discover a security vulnerability in applications, operating systems, or networks, thus feed them with a large amount of random data to crash them	18184	6062
Shellcode	An attack that sends the intrusion code to exploit a specific vulnerability and control a target machine	1133	378
Reconnaissance	An attack that collects the network system information to evade its security control	10491	3496
Exploit	An attack that sends the intrusion code to control a target system by triggering a vulnerability from unintentional behaviors on a host or network	33393	11132
DoS	An attack that directly or indirectly exhausts the resources of a target computer or network, to disable the normal service or resource access	12264	4089
Worm	An attack that makes security failures by replicating itself and transmitting through networks to other computers	130	44
Generic	An attack that uses the hash function to make collisions between each block cipher regardless configurations	40000	18871

and man-made attack traffic in the modern network [26]. As listed in Table I, totally 42 features are included in the dataset, and are divided into four categories as: basic features, content features, time features, and extra generated features. In particular, the one-hot encoding is utilized for numeralization and normalization of some features (e.g., content feature). Finally, the total feature dimensions reach to 196.

There are nine types of anomalies within the dataset, i.e., fuzzers, analysis, backdoor, DOS, exploit, generic, shellcode, reconnaissance, and worm. This dataset with multiple attack types enables us to evaluate the FAR for the proposed method effectively. The detailed description and number of attacks are shown in Table II.

Besides, data preprocessing is necessary to refine the raw data before conducting anomaly detection evaluations. The concrete data preprocessing in our experiment consists of the following three steps.

1) Transform the symbol features into numerical representation. Taking the feature "state" for example, this feature holds 11 alternatives for its value, and is represented by digits 0 to 10, respectively.
2) Convert the type label to a numerical representation, e.g., one represents the normal type, two represent the backdoor type, three represent the analysis type, and etc.
3) Normalize the data processed based on the first two steps.

B. Experiment Design and Evaluation Metrics

To evaluate the effectiveness of the proposed method, classical methods including Naïve Bayes (NB), random forest (RF), AdaBoost, machine learning methods including LSTM, CNN-LSTM, and a deep learning method named stacked sparse auto encoder (SSAE) [27], are chosen as baseline methods for comparison. All the experiments were conducted in a server with Ubuntu, GTX 1070, G39030 Duel Core, 16G RAM, and Python 3.6.

Metrics such as precision, recall, F1, FAR, and area under curve (AUC), are used to demonstrate performances of methods mentioned above for evaluations. In particular, Precision reflects the model's ability in distinguishing anomalies from the normal ones. Recall reflects the ratio of the model in finding the existing anomalies. F1 depicts the overall prediction performance based on the balance between precision and recall. FAR indicates the false alarm rate when detecting the network traffic anomalies. Practically, the higher the FAR, the worse detection performance will be. AUC represents a classifier's ability in classifying positive and negative examples. Especially, it makes reasonable evaluations of classifiers in cases of imbalanced samples. In this experiment, AUC and FAR are viewed as two important metrics to evaluate the anomaly detection performance when dealing with the imbalanced dataset in IBD environments.

C. Evaluation on Reparameterization Effectiveness

We first investigate the representation of hidden variable Z, to evaluate the reparameterization effectiveness of the compression network. Three variables: the generated hidden variable Z, input raw feature vector X, and reconstructed vector X', are compressed into a three-dimensional (3-D) vector, and further visualized using principal components analysis (PCA). We compare and observe the potential representations of the data in a 3-D view, which are illustrated in Fig. 2.

Obviously, as shown in Fig. 2(a) and (b), the distributions of the input vector and reconstruction vector are almost consistent. This result demonstrates that the reconstruction vector X' basically holds the identical information compared with the original input vector. It also indicates that the hidden variable, which is used to generate the reconstruction vector, retains the adequate information for recovery.

To verify whether the hidden variable retains necessary features to distinguish anomalies from the normal data, we visualize the hidden variable based on PCA as shown in Fig. 2(c). It is observed that results represented by the hidden variable, which are illustrated as dots with different shapes and colors, are clearly clustered into two parts, namely, the attacks and normal ones. This clustering phenomenon obviously indicates that the anomalies are successfully identified and distinguished from the normal ones, which means those critical features in the original input data are efficiently retained in the hidden variable. In addition, the distinct distance between the two clusters, denoted as the

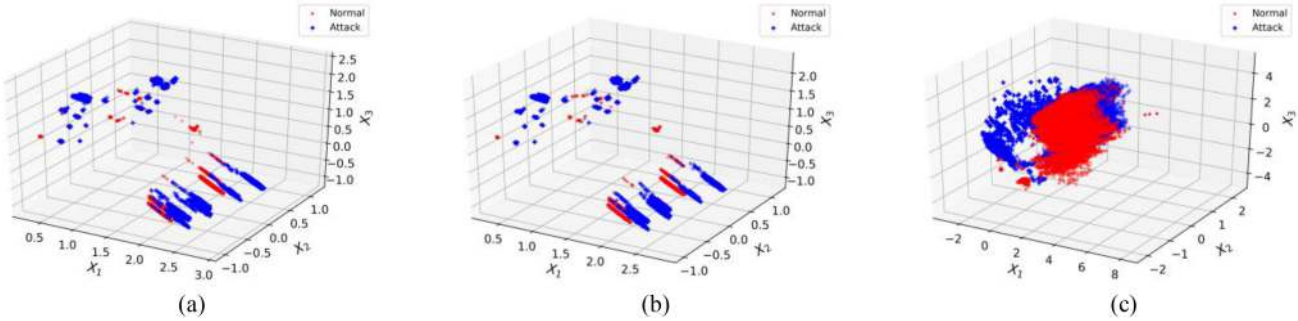


Fig. 2. Reparameterization evaluation based on PCA. (a) Input vector. (b) Reconstructed vector. (c) Hidden variable.

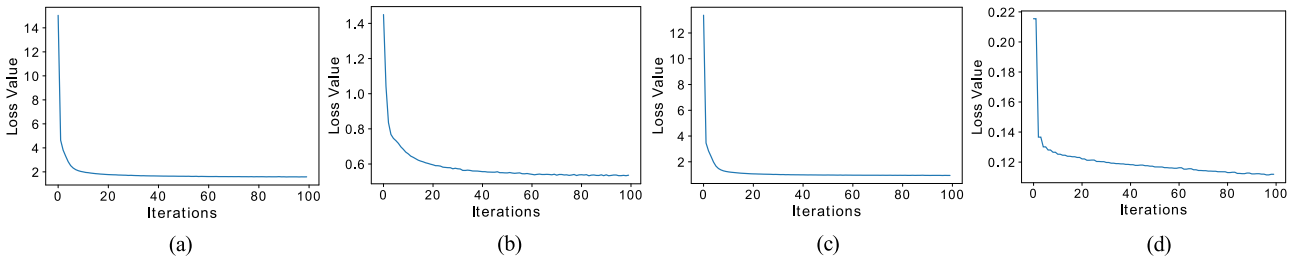


Fig. 3. Loss curves for VLSTM training process. (a) $\mathcal{L}_{vlstm}(x,\theta)$ loss curve. (b) $\mathcal{L}_{kl}$ loss curve. (c) $\mathcal{L}_{recon}$ loss curve. (d) $\mathcal{L}_{\mu}$ loss curve.

blue dots and red dots, illustrates this hidden variable may lead to a relatively low FAR score in the latter estimation network. In summary, these PCA results can verify the effectiveness of our VLSTM model in optimizing the hidden variable via the proposed compression network, especially when dealing with the imbalanced and high-dimensional IBD.

D. Analysis on Anomaly Detection Performance

We choose Adam, which is an upgraded version of stochastic gradient descent, as our optimizer. The learning rate is set to 0.005, and maximum iterations are set to 200 in this experiment. To evaluate the training process of our proposed learning model, the losses of $\mathcal{L}_{recon}$, $\mathcal{L}_{kl}$, $\mathcal{L}_{\mu}$, and $\mathcal{L}_{vlstm}$ in each iteration are compared and shown in Fig. 3, respectively.

As shown in Fig. 3, the overall loss in the proposed method declines fast within the first ten iterations then becomes relatively stable. This result indicates the adaptability of our learning model in IBD environments.

We further compare our method with the six baseline methods in terms of their capabilities in detecting sparse attacks from IBD. In this scenario, FAR is a significant indicator to demonstrate the performance of anomaly detection in real world. The evaluation results are listed and compared in Table III.

As shown in Table III, we demonstrate results based on the validation data and testing data, respectively. Basically, it is found that the six baseline methods perform well on the validation data but become relatively worse on the testing data, which can be explained as an overfitting issue. Contrastively, the VLSTM method outperforms these six methods with F1 at 0.907, FAR at 0.117, and AUC at 0.895, on the testing data. This indicates that the proposed VLSTM model can effectively tackle the overfitting issue for IBD. The overall results illustrate that our method can efficiently distinguish the true anomalies from normal network traffic data, and significantly mitigate the false anomaly detection rate comparing with the baseline methods.

TABLE III
COMPARISONS ON ANOMALY DETECTION PERFORMANCE

Method	Validation Dataset					Testing Dataset				
	Precision	Recall	F1 score	FAR	AUC	Precision	Recall	F1 score	FAR	AUC
SSAE	0.877	0.969	0.921	0.281	0.844	0.731	0.963	0.832	0.43	0.76
CNN-LSTM	0.993	**0.997**	**0.996**	**0.01**	0.994	0.801	0.956	0.872	0.29	0.833
LSTM	0.95	0.969	0.959	0.108	0.93	0.808	**0.99**	0.89	0.289	0.851
NB	0.92	0.989	0.953	0.184	0.903	0.749	0.975	0.847	0.401	0.806
RF	**0.995**	0.918	0.957	0.09	**0.995**	0.776	0.986	0.872	0.352	0.821
AdaBoost	0.949	0.972	0.96	0.111	0.93	0.811	0.973	0.885	0.344	0.848
VLSTM	0.967	0.949	0.958	0.039	0.941	**0.86**	0.978	**0.907**	**0.117**	**0.895**

E. Discussion

We summarize our observations and discuss reasons that why the proposed model can achieve better results and outperform other baseline methods as follows.

The proposed VLSTM model is mainly composed of a compression network and an estimation network. The compression network encodes the high-dimensional raw data into a low-dimensional hidden variable. Benefited by this newly designed neural network structure, it can successfully achieve a good tradeoff between the mitigation of computation complexity and retention of critical features for anomaly detection, as shown in Fig. 2. The novel design of the loss functions, including a

reconstruction loss function, a classification loss function, and a divergence loss function, can effectively constrain the hidden variable, and result in a reasonable gradient descent speed with few iterations during the training process, as shown in Fig. 3. Although the six conventional learning methods may perform well on the validation data, the highest results of F1, FAR, and AUC shown in Table III demonstrate the efficiency of our VLSTM method in improving the accuracy of classification tasks, and reducing the false rate of anomaly detections for the imbalanced and high-dimensional data in IBD environments.

VI. Conclusion

In this article, a VLSTM learning model was designed to cope with the imbalance and high-dimensional issues, which could be applied for intelligent anomaly detection based on reconstructed feature representation in IBD environments.

We introduced a generic framework to realize the VLSTM model, which was mainly composed of a compression network and an estimation network. The core structure of compression network included the LSTM encoder module, variational reparameterization module, and LSTM decoder module, which was designed to mitigate the complexity of high-dimensional raw data but without losing critical features. A reparameterization scheme based on variational Bayes was proposed to reconstruct a hidden variable for low-dimensional feature representation. In particular, three loss functions, namely, the reconstruction loss $\mathcal{L}_{recon}^{(i)}$, classification loss $\mathcal{L}_{\mu}^{(i)}$, and divergence loss $\mathcal{L}_{kl}^{(i)}$, were defined and seamlessly integrated together to constrain the hidden variable into a more explicit and meaningful form. The lightweight estimation network fed with the refined feature representation was then constructed to provide the network traffic classification. A learning algorithm was developed for the intelligent anomaly detection. Experiments were conducted using an open dataset named UNSW-NB15. Evaluation results demonstrated that the proposed VLSTM model could significantly enhance the feature extraction, reduce the false rate, and improve the detection accuracy based on an efficient training process, thus indicated the usefulness of our method in intelligent anomaly detection for IBD.

In future studies, we realize that the imbalanced data are still a challenge in anomaly detection tasks. We will investigate more deep learning techniques to adjust our models. More evaluations and experiments will be conducted to improve our algorithm to deal with more complex IBD.

References

[1] J. Qi, P. Yang, L. Newcombe, X. Peng, Y. Yang, and Z. Zhao. "An overview of data fusion techniques for Internet of Things enabled physical activity recognition and measure," *Inf. Fusion*, vol. 55, pp. 269–280, 2020.

[2] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surv. Tut.*, vol. 18, no. 2, pp. 1153–1176, Apr./Jun. 2016

[3] S. M. Kasongo and Y. J. Sun, "A deep long short-term memory based classifier for wireless intrusion detection system," *ICT Express*, vol. 6, no. 2, pp. 98–103 2020.

[4] D. P. Kingma and M. Welling, "Stochastic gradient vb and the variational auto-encoder," in *Proc. 2nd Int. Conf. Learn. Representations*, 2014, vol. 19.

[5] J. Qi, P. Yang, M. Hanneghan, S. Tang, and B. Zhou, "A hybrid hierarchical framework for gym physical activity recognition and measurement using wearable sensors," *IEEE Int. Things J.*, vol. 6, no. 2, pp. 1384–1393, Apr. 2019.

[6] Y. Zuo, Y. Wu, G. Min, and L. Cui, "Learning-based network path planning for traffic engineering," *Future Gener. Comput. Syst.*, vol. 92, pp. 59–67, 2019.

[7] J. Li, Z. Zhao, R. Li, and H. Zhang, "AI-based two-stage intrusion detection for software defined iot networks," *IEEE Int. Things J.*, vol. 6, no. 2, pp. 2093–2102, Apr. 2019.

[8] J. Qi, P. Yang, A. Waraich, Z. Deng, Y. Zhao, and Y. Yang, "Examining sensor-based physical activity recognition and monitoring for healthcare using Internet of Things: A systematic review," *J. Biomed. Informat.*, vol. 87, pp. 138–153, 2018.

[9] C. Huang, G. Min, Y. Wu, Y. Ying, K. Pei, and Z. Xiang, "Time series anomaly detection for trustworthy services in cloud computing systems," *IEEE Trans. Big Data*, to be published, doi: 10.1109/TBDATA.2017.2711039.

[10] W. Meng, E. W. Tischhauser, Q. Wang, Y. Wang, and J. Han, "When intrusion detection meets blockchain technology: A review," *IEEE Access*, vol. 6, pp. 10179–10188, 2018.

[11] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Trans. Emerg. Topic Comput. Intell.*, vol. 2, no. 1, pp. 41–50, Feb. 2018.

[12] E. Anthi, L. Williams, M. Słowińska, G. Theodorakopoulos, and P. Burnap, "A supervised intrusion detection system for smart home IOT devices," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 9042–9053, Oct. 2019.

[13] D. Midi, A. Rullo, A. Mudgerikar, and E. Bertino, "KALIS—A system for knowledge-driven adaptable intrusion detection for the Internet of Things," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst.*, 2017, pp. 656–666.

[14] P. Pongle and G. Chavan, "Real time intrusion and wormhole attack detection in Internet of Things," *Int. J. Comput. Appl.*, vol. 121, no. 9, pp. 1–9, 2015.

[15] R. Stephen and L. Arockiam, "Intrusion detection system to detect sinkhole attack on RPL protocol in Internet of Things," *Int. J. Elect. Electron. Comput. Sci. Eng.*, vol. 4, no. 4, pp. 16–20, 2017.

[16] W. Wang *et al.*, "HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection," *IEEE Access*, vol. 6, pp. 1792–1806, 2018.

[17] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[18] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning DDoS detection for consumer Internet of Things devices," in *Proc. IEEE Secur. Privacy Workshops*, 2018, pp. 29–35.

[19] Y. Zuo, Y. Wu, G. Min, C. Huang, and K. Pei, "An intelligent anomaly detection scheme for micro-services architectures with temporal and spatial data analysis," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 2, pp. 548–561, Jun. 2020.

[20] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring: A survey," 2016, *arXiv: 1612.07640*.

[21] T. Ma, F. Wang, J. Cheng, Y. Yu, and X. Chen, "A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks," *Sensors*, vol. 16, no. 10, 2016, Art. no. 1701.

[22] O. Brun, Y. Yin, and E. Gelenbe, "Deep learning with dense ran- dom neural network for detecting attacks against IoT-connected home environments," *Procedia Comput. Sci.*, vol. 134, pp. 458–463, Jul. 2018.

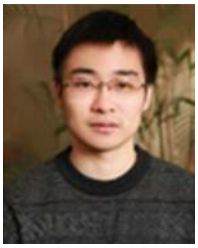
[23] S. Potluri and C. Diedrich, "Accelerated deep neural networks for enhanced intrusion detection system," in *Proc. IEEE 21st Int. Conf. Emerg. Technol. Factory Autom.*, Sep. 2016, pp. 1–8.

[24] Z. Tian, C. Luo, J. Qiu, X. Du, and M. Guizani, "A distributed deep learning system for web attack detection on edge devices," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 1963–1971, Mar. 2020.

[25] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. Dec, pp. 3371–3408, 2010.

[26] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Inf. Secur. J. Global Perspective*, vol. 25, no. 1/3, pp. 18–31, 2016.

[27] Y. Lin, J. Wang, Y. Tu, L. Chen, and Z. Dou, "Time-related network intrusion detection model: A deep learning method," in *Proc. IEEE Global Commun. Conf.*, 2019, pp. 1–6.



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, Shiga, Japan. From 2012 to 2015, he was a Research Associate with the Faculty of Human Sciences, Waseda University, Japan. He also works as a Visiting Researcher at the RIKEN Center for Advanced Intelligence Project, RIKEN, Japan, since 2017. He has 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-system, cyber intelligence and security.

Dr. Zhou is a member of the IEEE Computer Society, Association of Computing Machinery, USA, Information Processing Society of Japan, and Japanese Society for Artificial Intelligence, Japan, and China Computer Federation, China.

Yiyong Hu (Member, IEEE) received the bachelor's degree in measurement and control technology and instrumentation from the Harbin University of Science and Technology, Harbin, China, in 2017. He is currently working toward the M.S. degree in computer science with the Hunan University of Technology and Business, Changsha, China.

He worked as a Software Development Engineer from 2017 to 2018. His main research interests include cybersecurity, artificial intelligence, and natural language processing.

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, Japan. He is currently working with the Key Laboratory of Hunan Province for Mobile Business Intelligence, Hunan University of Technology and Business, Changsha. He has authored more than 20 papers at various conferences and journals. His current research interests include information retrieval, data mining, and artificial intelligence.

Dr. Liang is a member of the IEEE Computer Society and China Computer Federation, China.

Jianhua Ma (Member, IEEE) received the Ph.D. degree in information engineering from Xidian University, Xi'an, China, in 1990.

He is a Professor with the Department of Digital Media, Faculty of Computer and Information Sciences, Hosei University, Tokyo, Japan. He has authored more than 300 papers, coauthored five books and edited more than 30 journal special issues. His research interests include multimedia, networking, pervasive computing, social computing, wearable technology, IoT, smart things, and cyber intelligence.

Prof. Ma is one of pioneers in research on Hyper World and Cyber World (CW) since 1996. He first proposed Ubiquitous Intelligence toward Smart World, which he envisioned in 2004, and was featured in the European ID People Magazine in 2005. He has conducted several unique CW-related projects including the Cyber Individual (Cyber-I), which was featured by and highlighted on the front page of IEEE Computing Now in 2011. He has founded three IEEE Congresses on "Smart World," "Cybermatics," and "Cyber Science and Technology", respectively, as well as IEEE Conferences on Ubiquitous Intelligence and Computing, Pervasive Intelligence and Computing, Advanced and Trusted Computing, Dependable, Autonomic and Secure Computing, Cyber Physical and Social Computing, Internet of Things, and Internet of People. He is a member of IEEE and Association for Computing Machinery, Chair of IEEE SMC Technical Committee on Cybermatics, founding Chair of IEEE CIS Technical Committee on Smart World, and in advisory board of IEEE CS Technical Committee on Scalable Computing.

Qun Jin (Senior Member, IEEE) received the Ph.D. degree in electrical engineering and computer science from Nihon University, Tokyo, Japan, in 1992.

He is a Professor with the Networked Information Systems Laboratory, Department of Human Informatics and Cognitive Sciences, Faculty of Human Sciences, Waseda University, Tokyo, Japan. He has been extensively engaged in research works in the fields of computer science, information systems, and social and human informatics. He seeks to exploit the rich interdependence between theory and practice in his work with interdisciplinary and integrated approaches. His current research interests cover human-centric ubiquitous computing, behavior and cognitive informatics, big data, data quality assurance and sustainable use, personal analytics and individual modeling, intelligence computing, blockchain, cyber security, cyber-enabled applications in healthcare, and computing for well-being.

Prof. Jin is a Senior Member of the Association of Computing Machinery and Information Processing Society of Japan.