HADIoT: A Hierarchical Anomaly Detection Framework for IoT

HAOTIAN CHANG ©, JING FENG, (Member, IEEE), AND CHAOFAN DUAN
College of Meteorology and Oceanography, National University of Defense Technology, Changsha 410073, China
Corresponding author: Jing Feng (jfeng@seu.edu.cn)

This work was supported by the China Nature Science Foundation Committee under Grant 61371119.

ABSTRACT The Internet of Things establishes the intimacy between the Internet and the physical world. Due to portable size, most IoT devices have limited computing and storage capabilities and are vulnerable to various malicious intrusions. Therefore, it is vital to have efficient approaches to distinguish the true IoT data from fake one, we term such methods as anomaly detection (AD). To detect anomalies accurately and efficiently, in this article a 3-hierarchy joint local and global anomaly detection framework, HADIoT, is proposed, in which IoT devices generate and transmit sensory data to their local edge servers for local AD after data refinement which includes re-framing, normalization, complexity reduction via Principal Component Analysis, and symbol mapping. High detection accuracy is achieved by jointly local and global ADs. The local AD focuses on the data pattern consistency of individual devices via the Gated Recurrent Unit, and the processed data is then forwarded from edge servers to the cloud server for global AD. The global AD focuses on the analysis of the data pattern correlations between different IoT devices, using the Conditional Random Fields. For the maintenance of cyber-security, the proposed anomaly detection framework HADIoT enables to provide an accurate and faster anomaly detection for IoT applications, compared with existing anomaly detection methods. The performance of the proposed method is also empirically evaluated through simulations, using a real dataset - the Information Security Center of Excellence (ISCX) 2012 dataset. Simulation results demonstrate the effectiveness of the proposed framework in terms of True Positive Rate, False Positive Rate, Precision, Accuracy and F_score, compared with three benchmark schemes.

INDEX TERMS Anomaly detection, Internet of Things, hierarchical framework, local data pattern, global data correlation.

I. INTRODUCTION

The recent advancement of Internet of Things (IoT) has further flourished new applications of IoT devices [1]. Manufactures are devouring the IoT device market, releasing new IoT products at exponential growth rate. Consequently, there are intrinsic vulnerabilities in many IoT devices where potential risks lurk. Since IoT devices bridge the physical world and cyberspace, threats to the vulnerable devices can be aggregated and amplified in the core network. Hence, it is vital to develop efficient detection methods to detect and mitigate the vulnerability of IoT devices to ensure the cyber-security of IoT environments.

In order to protect the IoT world from anomalous intruders, intrusion detection systems (IDSs) have been developed. There are two main types of IDSs, signature-based detection systems and anomaly-based detection systems [2]. Signature-based IDSs are supposed to recognize familiar patterns, thus not suitable to detect the anomalous data with evolving patterns. In contrast, anomaly-based detection systems are the one more suitable for IoT environments. Specifically, an anomaly is a data point in data space which is significantly different from the rest of the data [3]. Since anomaly-based detection systems are able to reveal unusual behaviors, they are capable of detecting intrusions with unknown patterns. Therefore, Anomaly Detection (AD) for continuously changing intrusion patterns in IoT environments poses great challenges.

One challenge is the heterogeneity of IoT devices. Because there are heterogeneous IoT devices on the market, the benign data patterns of different devices with dissimilar surveillance objectives varies. Therefore, it is unrealistic to train only one precise AD model for all types of IoT devices. Meanwhile, IoT devices are endowed with different quantities
of resources, so not all of them are capable of performing on-device AD.

Another challenge is the detection accuracy. Anomaly detection methods can be categorized into parametric and non-parametric categories [4]. The parametric methods assume that the distributions of data are a priori. However, network data is generated by multiple processes that reflect the functionalities of corresponding systems. It is unrealistic to suppose that the data is distributed as a single normal distribution. In contrast, non-parametric methods are well performed in the cases where data patterns are vague. The basic idea is to find the regions with low density of data. Nevertheless, this method usually has high computational complexity because of the large numbers of data features, which is commonly referred to as the curse of dimensionality. However, some data features may be irrelevant with anomalous behaviors and they slow down the data training. Therefore, reducing computational complexity of datasets is also a fundamental issue.

To tackle the aforementioned challenges, an efficient AD framework for IoT needs to be designed. It should take considerations of detection accuracy and computational complexity seamlessly. Hence, a hierarchical AD framework for IoT applications, the HADIoT, is proposed in this article. Three important issues are explored in the framework, i.e., the architecture design, accuracy improvement and reduction of computational complexity.

In the proposed framework, a 3-hierarchy architecture is designed. Specifically, IoT devices generate and transmit sensory data to local edge servers for local processing which includes dimensionality reduction via Principal Component Analysis (PCA) and symbol mapping, and local AD via the Gated Recurrent Unit (GRU). Afterwards, large-scale data is delivered from edge servers to cloud servers for global AD, using the Conditional Random Fields (CRF). High detection accuracy is jointly achieved by local and global AD via edge servers and cloud servers, respectively. Local AD focuses on the data pattern consistency of individual devices while global AD emphasizes the pattern correlations between different devices. We perform simulations of the proposed framework and prove its effectiveness, using a real dataset from the Information Security Center of Excellence (ISCX).

The main contributions of this article are as follows.

- We design a novel hierarchical AD architecture for IoT. IoT devices transmit their data to local edge servers via which the processed data is then forwarded to the cloud server for further analysis.
- To the best of our knowledge, we are the first to propose a joint local and global AD framework to enhance the detection accuracy in IoT environments.
  - In local edge servers, we adopt PCA to reduce the data dimensionality. We map data vectors to symbols, which reduces the computational complexity and overfitting simultaneously. We use the Gated Recurrent Unit to locally detect the anomalies of each device in reference of its individual symbol patterns.
  - In the global processing, IoT data from each local server are forwarded to the cloud server for large-scale AD. We adopt CRF to detect anomalies via data correlations in the macro data space, i.e., anomalies of each device can be identified in reference of the data patterns of the other devices.
- We demonstrate the effectiveness of the proposed framework through simulations, using a real dataset from the ISCX.

The remainder of this article is organized as follows. Section II will review the related work of AD. Section III will introduce preliminaries, including the system model, AD framework and problem definitions. Section IV will propose the refinement of data. Section V will detail the local AD and global AD methods. Section VI will conduct the simulations and analysis of our proposed framework, and Section VII will conclude the the paper and potential study in future.

II. RELATED WORK

Non-parametric methods for anomaly detection are more sophisticated in multi-pattern systems. Xie et al. adopted the K-Nearest Neighbors algorithm to define a hyperplane around a data point, and a data point is identified as anomalous if there are less than \( k \) data points inside its hyperplane [5]. But the computational complexity of this method is too high to be applied in high dimensional cases. Chernogorov et al. applied diffusion maps algorithm for dimensionality reduction of the data and then employ the K-means algorithm to detect the anomalies in [6]. Lyu et al. proposed an efficient hyperellipsoidal clustering algorithm for anomaly detection [7]. In addition, Bars et al. presented a probabilistic framework to model and assess the abnormal communication volume at any single node by knowing the set of nodes involved [8].

With the development in deep learning techniques, Ezeme et al. introduced a hierarchical attention-based anomaly detection (HAbAD) model based on stacked Long Short-Term Memory (LSTM) Networks with Attention [9]. Yahyaoui et al. proposed a detection protocol that dynamically executes the on-demand Support Vector Machine (SVM) classifier in a hierarchical way whenever an intrusion is suspected [10]. Nguyen et al. proposed an autonomous self-learning distributed system, DIoT, to detect compromised IoT devices, which uses a novel self-learning approach to detect compromised devices [11]. However, they did not consider the correlations between different types of devices.

In this article, we construct a novel hierarchical AD framework for IoT. We reduce the data dimensionality with both PCA and symbol mapping. To the best of our knowledge, we are the first to jointly consider the local and global data patterns, i.e., the intra-node and inter-node behaviors, which is the main difference between our work and the existing ones.
H. Chang et al.: HADIoT: A Hierarchical Anomaly Detection Framework for IoT

III. PRELIMINARIES

A. THE SYSTEM MODEL

Consider a 3-hierarchy IoT architecture as shown in Fig. 1. The HADIoT consists of multiple local edge servers and one global cloud server. The local edge servers perform data pre-processing and local AD in reference of each device’s own data pattern. The local edge servers then forward the processed data to the cloud server, and the latter provides global AD services which is required of higher computational capacity. Specifically, the three hierarchy of the architecture is detailed as follows.

1) $H1$: The cloud server is equipped with higher computation capacity than edge servers. Local edge servers offload the processed data to the cloud server for global AD which takes into consideration the correlation of data patterns in the macro data space, i.e., between all the IoT devices. Global AD adopts a machine learning method, CRF, which is usually applied to natural language processing.

2) $H2$: The local edge servers perform data pre-processing and local AD for IoT devices. They re-frame and normalize the data from heterogeneous devices. Also, local servers use PCA to reduce the data dimensionality and then adopt symbol mapping to further decrease the computational complexity of AD. Lastly, they use a sequence prediction method to detect anomalies of each device according to its historical data pattern.

3) $H3$: Heterogeneous IoT devices generated and deliver data to local edge servers.

B. ANOMALY DETECTION FRAMEWORK

This subsection provides an overview of the proposed framework for AD in the context of IoT data. The flow chart is depicted in Fig. 2. There are three stages of data processing: 1) data pre-processing; 2) local AD; and 3) global AD. The detailed descriptions of these three stages are presented as follows.

Stage 1. Data pre-processing is the first stage of the proposed framework. We here use the ISCX 2012 dataset as the input of the framework. The first stage is performed on the edge server and consists of three phases, i.e., re-framing and normalizing, dimensionality reduction and symbol mapping.

i) Since IoT devices are heterogeneous, e.g., smart plugs report in frames while electronic thermometers deliver packets to routers, the processing of IoT data needs to re-frame the data in the same format. Moreover, the payloads of devices vary from each other. Specifically, some payloads are quantitative while some are qualitative. Different payloads must be normalized in order to perform the universal AD on them.

ii) Some indicators of IoT data within the same local region always remain the same, so they are redundant for the AD processing. Hence, we can discard these redundant indicators so as to reduce the computational complexity. We adopt PCA to realize dimensionality reduction of the IoT data.

iii) IoT devices are typically appliances with particular functions, their behavior patterns are relatively static and limited. Hence the framework is able to capture all possible benign behaviors of IoT devices, where benign patterns of each device can be easily calculated. And the reduced data vectors from the second phase can be classified into different classes in correspondence of the behaviors. Each class is mapped to a specific data symbol $S_i$, where $i$ refers to the device index and $t$ denotes the time slot. Symbol mapping further reduces the complexity of multi-dimensional vector calculations, which enhances the feasibility of local and global AD in both Stage 2 and Stage 3.

Stage 2. The second stage of data processing is the local AD via Gated Recurrent Unit (GRU), which is also performed on the edge server. The classifications of the data vectors of device $i$, denoted by the symbol sequence $S^i = [S_{i1}^i, S_{i2}^i, ..., S_{it}^i]$, are recorded in the local edge server, and the symbol sequence reveals the pattern of each device. We identify anomalies using the likelihood of the occurrence of symbols in the sequence of its device, because the data of IoT devices usually follows particular historical patterns. The GRU method is adopted here because of its promising capability of time serial prediction along with less parameters to tune, compared with the well-known Long Short Term Memory (LSTM) method.

Stage 3. The third stage is the global AD using Conditional Random Fields (CRF). Having identified by its own
C. DATA MODEL

Given the raw dataset of IoT devices \( \mathcal{D} = \{ D_i^t \in \mathbb{R}^k | t = 1, 2, \ldots, T, i = 1, 2, \ldots, N \} \), where \( N \) is the number of IoT devices and \( l_i \) denotes the length of the data vector of device \( i \). Then, \( \mathcal{D} \) is re-framed, normalized and mapped to the symbol set \( \mathcal{S} = \{ S_i^t | t = 1, 2, \ldots, T, i = 1, 2, \ldots, N \} \), where \( S_i^t \in \{ 1, 2, \ldots, J_t \} \) and \( J_t \) indicates the number of data classes, i.e., the number of data symbols, of device \( i \). Therefore, the IoT data can be indexed by time and device. The re-framing, normalization and symbol mapping of the IoT data will be introduced in Section 4.

D. PROBLEM DEFINITION

We aim to compute the anomalous score for symbols of IoT devices and to determine its anomaly. It is defined locally and globally, by local edge servers in Stage 2 and by the cloud server in Stage 3, respectively.

Definition 1 (Local Anomaly): Symbol \( S_i^t \) corresponding to device \( i \) at time \( t \) is determined as a local anomaly, if its local probability of occurrence is below the local threshold \( \theta_l \), as shown in (1).

\[
A_l^i = \begin{cases} 
1, & P_l^i = P(S_i^t | S_{i-1}^t, S_{i-2}^t, \ldots, S_{i-k}^t) < \theta_l \\
0, & \text{otherwise}
\end{cases}
\]

where \( P_l^i \) represents the likelihood of \( S_i^t \) in reference of the historical pattern of device \( i \), and \( A_l^i = 1 \) indicates \( S_i^t \) is locally anomalous. The calculation of \( P_l^i \) will be introduced in detail in Section IV.

Definition 2 (Global Anomaly): Symbol \( S_i^t \) of device \( i \) at time \( t \) is determined as a global anomaly when its global likelihood of occurrence \( P_g^i \) is below the global threshold \( \theta_g \), as presented in (2). The calculation of \( P_g^i \) will be introduced in Section V.

\[
A_g^i = \begin{cases} 
1, & P_g^i = P(S_i^t | S_1^t, S_2^t, \ldots, S_N^t) < \theta_g \\
0, & \text{otherwise}
\end{cases}
\]

where \( A_g^i = 1 \) indicates that \( S_i^t \) is globally anomalous.

Definition 3 (Overall Anomaly): Whether or not \( S_i^t \) is anomalous can be determined by the Boolean result of \( A_l^i \) and \( A_g^i \), as shown in (3).

\[
A^i = \begin{cases} 
1, & A_l^i \land A_g^i = 1 \\
0, & \text{otherwise}
\end{cases}
\]

where \( A^i = 1 \) indicates that \( S_i^t \) is anomalous.

IV. DATA REFINEMENT

In this section, we introduce the refinement of IoT data. Raw data packets \( D_i^t \) in the packet sequence \( D_1^t, D_2^t, \ldots, D_N^t \) of device \( i \) are re-framed and normalized to \( C_i^t \) according to 8 features of the packet characteristics, \( (c_1, c_2, \ldots, c_8) \). Then, the data is further reduced to \( X_i^t \) via PCA, and finally mapped to data symbols \( S_i^t \).

A. RE-FRAMING AND NORMALIZATION

The normalization of re-framed data packets according to the selected 8 features are shown in Table 1. The features are selected by the PCA method which will be demonstrated in the following subsection, and the final selection of the optimal features will be presented in detail in Section VI. The data value of \( c_1, c_2, c_3, c_6, c_7, \) and \( c_8 \) are normalized linearly from 0 to 1, in reference of the minimum and maximum values of each feature. Meanwhile, the direction and protocol name are both quantified as binary values. Hence, all the values of the 8 features are normalized between 0 and 1, which makes it achievable to perform generic processing over the packets from heterogeneous devices.

The re-framed and normalized data can be depicted as:

\[
[C_1, C_2, \ldots, C_N]^T = \begin{bmatrix} C_1^1 & C_1^2 & \cdots & C_1^{t-k} \\ C_2^1 & C_2^2 & \cdots & C_2^{t-k} \\ \vdots & \vdots & \ddots & \vdots \\ C_N^1 & C_N^2 & \cdots & C_N^{t-k} \end{bmatrix} \tag{4}
\]

where \( C_i = [C_i^1, C_i^2, \ldots, C_i^{t-k}]^T \) indicates the data sequence of device \( i \) with a window of \( k \) time slots, and \( A^T \) denotes the transposition matrix of matrix \( A \).

B. DIMENSIONALITY REDUCTION

Having re-framed and normalized from \( D_i^t \) to \( C_i^t \), we now refine the data by filtering its redundant features via PCA. In this phase, \( C_i^t \) is linearly transformed to the value with the mean of zero at first, as shown in (5),

\[
\tilde{C_i^t} = C_i^t - \frac{1}{8} \sum_{j=1}^{8} C_i^{ij}
\]

where, \( m = 1, 2, \ldots, 8 \) denotes the index of the entry of \( C_i^t \). Hence, the square sum of each entry denotes the variance of \( \tilde{C_i^t} \). The covariance matrix of device \( i \) is
formulated as (6),

$$\text{Cov}^i = \bar{C}^i (\bar{C}^i)^T$$

$$= \begin{bmatrix}
\bar{C}^i_1 (\bar{C}^i_1)^T & \bar{C}^i_1 (\bar{C}^i_{-1})^T & \cdots & \bar{C}^i_k (\bar{C}^i_{-k})^T \\
\bar{C}^i_{-1} (\bar{C}^i_1)^T & \bar{C}^i_{-1} (\bar{C}^i_{-1})^T & \cdots & \bar{C}^i_{-1} (\bar{C}^i_{-2})^T \\
\vdots & \vdots & \ddots & \vdots \\
\bar{C}^i_{-k} (\bar{C}^i_1)^T & \bar{C}^i_{-k} (\bar{C}^i_{-1})^T & \cdots & \bar{C}^i_{-k} (\bar{C}^i_{-k})^T 
\end{bmatrix}$$

(6)

where \(\bar{C}^i = [\bar{C}^i_1, \bar{C}^i_{-1}, \ldots, \bar{C}^i_{-k}]^{'}\). Since \(\text{Cov}^i\) is a real diagonalizable matrix, it can be diagonalized as \(\Lambda^i\), as shown in (7),

$$\Lambda_i = \begin{bmatrix}
\lambda_{i,1} & & \\
& \ddots & \\
& & \lambda_{i,k+1}
\end{bmatrix}$$

(7)

where \(\lambda_{i,j}\) is the \(j\)-th largest eigenvalue of \(\text{Cov}^i\) and \(\Lambda^i = (B^i)^T \text{Cov}^i B^i\). \(B^i\) denotes the matrix consists of all the eigenvectors of \(\text{Cov}^i\). Lastly, the larger \(\lambda_{i,j}\) is, the more significant its corresponding component is. And the redundant features corresponding to the smaller eigenvalues are to be discarded, thus reducing \(\bar{C}^i\) to \(X^i\) via (8).

$$X^i = (B^i)^T \bar{C}^i$$

(8)

where \(B^i\) is the reduced matrix constituted by eigenvectors with less dimensionality, \(i.e.,\) redundant eigenvectors corresponding to small eigenvalues are discarded from \(B^i\). Therefore, we realize the projection from \(\bar{C}^i\) to \(X^i\) with dimensionality reduced, and \(\bar{C}^i\) is transformed to \(X^i\).

C. SYMBOL MAPPING

Each row vector in \(X_i\), denoted by \(X^i_j\), corresponds to the refined data of device \(i\) at time \(t\). We aim to simplify the data from vectors to symbols, \(i.e.,\) symbol mapping. We adopt the K-means algorithm with the Silhouette method to cluster the data of device \(i\) into \(J_i\) classes. Therefore, the symbol set of device \(i\) is denoted by \(\mathcal{S} = \{S^i_j | 1, 2, \ldots, J_i\}\), where \(\mathcal{S} \subset \mathcal{S}\).

\(J_i\) is determined by the Silhouette method which uses the average Silhouette coefficient to evaluate how well the data is clustered under different parameters. Having mapped the data to the symbol set, the volume of IoT data is further reduced. Meanwhile, the noise interference on the data can be smoothed, because the data in the same cluster is represented by the cluster center and the impact of data variations in the same cluster on the AD computation can be ignored.

V. ALGORITHM OF JOINT LOCAL AND GLOBAL ADs

In this section we propose an algorithm for anomaly detection of IoT application by jointly considering both local and global ADs as follows.

A. LOCAL AD

We here detail an application of the standard Gated Recurrent Unit (GRU) to perform the local AD in local edge servers. The GRU method, by Cho et al., is a variant of The Long Short-Term Memory (LSTM) [12], which combines the forget gate and the input gate to a simple update gate. Each recurrent unit adaptively captures dependencies of different time scales, thus enabling the GRU method to reveal correlations of symbols of any device within a given time span.

In the gated recurrent unit of device \(i\) at time \(t\), there are two gates, the update gate \(r^i_t\) and the reset gate \(r^i_t\). The update gate decides how much the unit updates its content. The update gate is computed as:

$$z^i_t = \sigma(W^i_s[h^i_{t-1}, x^i_t])$$

(9)

where \(W^i_s\) indicates the weight coefficient matrix of the update gate of device \(i\), \(h^i_{t-1}\) denotes the memory content propagated from the previous time slot, and \(x^i_t\) represents the input symbol of device \(i\) at time \(t\).

The current memory content \(h^i_t\) is calculated as follows.

$$h^i_t = \tanh(W^i_r[h^i_{t-1} \odot r^i_t, x^i_t])$$

(10)

where \(W^i_r\) is the weight coefficient matrix of the reset gate and \(\tanh\) and \(\odot\) indicate the sigmoid function, hyperbolic tangent function and element-wise multiplication, respectively. And \(r^i_t\) indicates the reset gate factor, which can be acquired as:

$$r^i_t = \sigma(W^i_r[h^i_{t-1}, x^i_t])$$

(11)

The final memory content of the GRU at time \(t\), \(i.e.,\) \(h^i_t\), is a linear interpolation between the previous memory \(h^i_{t-1}\) and the current memory \(h^i_t\), which is defined as follows.

$$h^i_t = (1 - z^i_t) \odot h^i_{t-1} + z^i_t \odot h^i_t$$

(12)

The probability distribution of the candidate symbols of device \(i\) at time \(t\), denoted by \(\tilde{y}^i_t\) can be calculated as:

$$\tilde{y}^i_t = \text{softmax}(V^i h^i_t)$$

(13)

where \(V^i\) denotes the projection matrix from \(h^i_t\) to \(\tilde{y}^i_t\).

GRU preserves the ability of LSTM, \(i.e.,\) learning the context in its context, and has a simpler structure which reduces the computational cost. When applied in our framework, \(x^i_t\) is the current input \(S^i_t\). And \(\tilde{y}^i_t\) denotes the likelihood of \(S^i_t\). The weight coefficient matrices of the reset and update gate, \(W^i_r\) and \(W^i_s\), are to be trained by the sample data.

In the HADIoT, we treat the sensing data of each device as a sentence. The \(t\)-th word of sentence \(i\) is denoted by \(D^i_t\). After symbol mapping, \(S^i_t\) represents the \(t\)-th word in sentence \(i\) with only one letter. We make use of a dataset of \(N\) sentences with each containing \(T\) words to train the GRU language model.

Each sentence is individually processed in its local edge server. For time slot \(t\), the GRU unit computes the output \(\hat{y}\) using the input \(S^i_t\) and the previous memory content \(h^i_{t-1}\), via Equation (9) to Equation (13). The dimensions of the input,
output and parameters are as follows:
\[
\begin{align*}
    x^t &= [S^t_1, S^t_2, \ldots, S^t_J] \in \mathbb{R}^{1 \times T} \\
    y^t_i &= [p(1), p(2), \ldots, p(J_i)] \in \mathbb{R}^{1 \times J_i} \\
    W^i_Z, W^i_f, W^i_l &\in \mathbb{R}^{k \times (k+1)} \\
    h^t_i, r^t, z^t &\in \mathbb{R}^{k \times 1} \\
    V &\in \mathbb{R}^{J \times k} \\
    y^*_i &\in \mathbb{R}^{1 \times 1}
\end{align*}
\]  

where \( k \) is the internal memory size of GRU. \( \hat{y}^t_i \) denotes the probability vector of the \( J_i \) potential symbol values of sentence \( i \), while the vector \( y^*_i \) indicates the ground truth of sentence \( i \) at time \( t \).

We then calculate the cross entropy loss \( L^i_t \) as follows.
\[
    L^i_t = -\hat{y}^*_i (\log \hat{y}^i)
\]  

where \( \log \) indicates the element-wise logarithm function.

To train the GRU, we need to acquire the values of all the parameters that minimize the total cross entropy loss of sentence \( i \), i.e.,
\[
    L^i = \sum_{t=1}^{T} L^i_t;
\]

\[
    \Theta^i = \arg \min_{\Theta^i} L^i
\]  

where \( \Theta^i = \{W^i_Z, W^i_f, W^i_l, V\} \).

We adopt the Stochastic Gradient Descent method, to tune the parameters [13]. With the trained GRU model, we calculate the likelihood vector \( \hat{y}^t_i \) to detect the local anomalies of each device \( i \). When the likelihood of the input data is calculated, the probability threshold \( \theta_t \) is automatically adjusted from 1 to 0 by the edge server. Corresponding to certain \( \theta_t \), the detection results, i.e., the \( TPR, FPR, Accuracy, Precision \) and \( F\_score \) are acquired. And the threshold \( \theta_t \) is ascertained when the corresponding \( Precision \) is maximized.

When the parameters and threshold are trained, the GRU model will be applied to the local AD. However, if the nature of the network traffic changes, the GRU parameters and threshold \( \theta_t \) will be re-trained.

**B. GLOBAL AD**

We then adopt the CRF in the global AD stage. The CRF is a machine learning model, through which the conditional probability distribution of hidden states can be calculated via the given observations. In the HADIoT, the CRF model constructs the conditional probability distribution of each potential state and determines the most probable labeling of the observed data. It was first introduced by Lafferty et al. for text sequence prediction [14], and has been applied to many problem in NLP, bioinformatics, and computer vision.

As mentioned in Definition 2, the global anomalous states of all devices at time \( t \) can be denoted as \( [A^1_{g_t}, A^2_{g_t}, \ldots, A^N_{g_t}] \), which is regarded as a Markov-chain. This assumption sounds because the input at time \( t \), \( x_t = [S^t_1, S^t_2, \ldots, S^t_N], \) has a sequential form, such as sequence prediction for Natural Language Processing (NLP) problems and gene sequence analysis for bioinformatics. Markov conditional random fields are also suitable for modeling and classifying IoT data because the data can be intrinsically regarded as sentences generated at each time slot. Moreover, the data generated by device \( i \) at time \( t \) can be described as the \( i \)th word in the \( r \)th sentence.

The probability distribution over random variables \( (x_t, A_{g_t}) \) is modeled via a Markov Random Field, which presents that the joint probability can be factorized into a product of potential functions over intrinsic features. The joint probability of a potential state variable anomaly \( A_g = [A^1_g, A^2_g, \ldots, A^N_g] \) given \( x_t \) can be calculated as:
\[
    P(A_{g_t} \mid x_t) = \frac{1}{Z(x_t)} \exp \left\{ \sum_{i=1}^{N} \sum_{w=1}^{W} \mu_{w} f_w(i, A^i_{g_t}, A^{i-1}_{g_t}, x_t) \right\}
\]  

where \( f_w \) denotes the \( w \)th feature function, along with the corresponding weight factor \( \mu_w \). \( Z(x_t) \) is the normalization divisor:
\[
    Z(x_t) = \sum_{A_{g_t}} \exp \left\{ \sum_{i=1}^{N} \sum_{w=1}^{W} \mu_{w} f_w(i, A^i_{g_t}, A^{i-1}_{g_t}, x_t) \right\}
\]

where \( A'_{g_t} \) denotes all the possible values of the state vector \( [A^1_{g_t}, A^2_{g_t}, \ldots, A^N_{g_t}] \) at time \( t \).

A feature function is an indicator which describes the contributions of each pair of adjacent states. For example, \( f_w(i, A^i_{g_t}) = 1, A^{i-1}_{g_t} = 1, x_t = 1 \). The CRF model is fully characterized by the feature function set and the weight factors, i.e., \( \{f, \mu\} \).

Since the CRF is also a supervised learning method, it needs to go through a training stage before applied in practice. The training stage computes the model parameters \( \mu \) according to the labeled sample data \( (A_{g_t}, x_t) \). The objective function can be described as
\[
    \sum_{i=1}^{T} \log P(A_{g_t} \mid x_t) = \sum_{i=1}^{T} \sum_{w=1}^{N} \mu_{w} f_w(i, A^i_{g_t}, A^{i-1}_{g_t}, x_t) - \sum_{t=1}^{T} \log Z(x_t).
\]

To gain the model parameters \( \mu \), we have to maximize the objective function. Because the objective function is convex, the gradient based optimization can guarantee to acquire the global optimal solution. We adopt the well-known gradient ascent method, limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm, to tune the CRF parameters in HADIoT.

When the CRF model is trained, it can be applied to predict the states of the newly-come input data \( x_t \) at time \( t \) via the Viterbi algorithm. The prediction sequence can be acquired as:
\[
    A^*_g = \arg \max_{A_{g_t}} P(A_{g_t} \mid x_t)
\]  

where \( A^*_g \) denotes the anomaly prediction vector of \( x_t \) with the maximum likelihood.
Similar to the training procedures of the local AD, when the likelihood of the input data is acquired, the probability threshold $\theta_{g}$ is automatically adjusted from 1 to 0 by the cloud server. And the threshold $\theta_{g}$ is ascertained when the corresponding Precision is maximized. When the nature of the network data changes, the CRF model and $\theta_{g}$ need to be re-trained.

Thus, the algorithm of the HADIoT can be presented as Algorithm 1.

**Algorithm 1 HADIoT**

**Require:** $\mathcal{D} = \{D_{t}^{i} \in \mathbb{R}^{s} | t = 1, 2, \ldots, T, i = 1, 2, \ldots, N\}$

**Ensure:** $\{A_{t}^{i} \in \{0, 1\} | t = T + 1, T + 2, \ldots, T', i = 1, 2, \ldots, N\}$

1. $\mathcal{D} \Rightarrow \mathcal{C}$; // Data re-framing and normalization.
2. $\mathcal{C} \Rightarrow \mathcal{X}$; // Dimensionality reduction via PCA.
3. $\mathcal{X} \Rightarrow \mathcal{S}$; // Symbol mapping via K-means.
4. $\Theta = \{W_{z}, W_{r}, W, V\}$; // Parameter initialization.

**for** $i = 1$ to $N$ **do**

1. While $\frac{\partial L_{i}}{\partial \mu_{i}}$ does not converge do
2. Calculate $L_{i}$; // Forward propagation.
3. Calculate $\frac{\partial L_{i}}{\partial \mu_{i}}$; // Back propagation.
4. $\Theta^{i} \leftarrow \Theta^{i} - \frac{\partial L_{i}}{\partial \mu_{i}}$; // Parameter update.

**end while**

**end for**

**for** $i = 1$ to $N$ **do**

1. For $t = T$ to $T'$ **do**
2. Calculate $V_{t}^{i}$;
3. Calculate $A_{t}^{i}$;

**end for**

$\mu = [\mu_{1}, \ldots, \mu_{w}, \ldots, \mu_{W}]$; // Parameter initialization.

**While** $\frac{\partial U}{\partial \mu_{j}}$ does not converge do

1. Calculate $U = \sum_{t=1}^{T} \log P(A_{g_{t}} | x_{t})$;
2. Calculate $\frac{\partial U}{\partial \mu_{j}}$; // Gradient Calculation.
3. $\mu \leftarrow \mu + \frac{\partial U}{\partial \mu_{j}}$; // Parameter Update.

**end while**

**for** $t = T$ to $T'$ **do**

1. Calculate $A_{t}^{i}$;
2. $A_{t}^{i} = A_{t}^{i} \parallel A_{t}^{j}$; // Final anomaly detection result.

**end for**

**A. EVALUATION METRICS**

We evaluate the performance of the proposed framework in terms of the following parameters: True Positive Rate (TPR), False Positive Rate (FPR), Precision, Accuracy and F-score. The mathematical definition of these parameters are illustrated as follows.

$$\text{TPR} = \frac{TP}{TP + FN}$$

$$\text{FPR} = \frac{FP}{FP + TN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Accuracy} = \frac{TN}{TP + TN + FP + FN}$$

$$F - score = 2 \cdot \frac{\text{Precision} \times \text{TPR}}{\text{Precision} + \text{TPR}}$$

$TP$, $TN$, $FP$ and $FN$ refer to True Positive, True Negative, False Positive and False Negative, respectively. More specifically, $TP$ indicates that the normal data is identified as normal and $TN$ refers to the case when the abnormal data is classified as normal. On the contrary, when the data is incorrectly detected as normal or abnormal, it is described as $FP$ and $FN$, respectively.

**B. DATASET**

We evaluate the efficiency of the proposed framework HADIoT, using the Information Security Center of Excellence (ISCX) 2012 dataset that consists of almost 1.5 million network packets of seven days, with 20 features. The ISCX 2012 dataset is labelled and the total size of it is 84.45GB. Since we suppose the desktop as the edge server in the simulation, the communication overhead from the IoT devices to the edge server can be regarded as the data volume of the original ISCX 2012 dataset. We randomly select the training dataset and testing dataset by a size ratio of 3 : 7. Specifically, there are 1046500 and 468500 data records in the training and testing set, respectively. Additionally, there are mainly three kinds of anomalies in the ISCX 2012 dataset, infiltration from inside, Denial of Service (DOS) and brute force attacks, which were generated on 13-15 and 17 June 2010. Meanwhile, network traffic data generated on 11, 12 and 16 June 2010 is normal.

**C. RESULTS**

For the sake of clarity, the simulation results are illustrated in four parts corresponding to the three stages and the overall performance of the HADIoT framework. In the first part, we demonstrate the results of feature selection and symbol mapping of the IBRL dataset. Then, the performance of local AD via GRU is illustrated in comparison with three benchmark schemes, Long Short-Term Memory (LSTM), K-Nearest Neighbors (KNN) and Convolutional Neural Networks (CNN). Thirdly, the global AD via CRF simulated on a DELL server is presented. Lastly, the performance of
the combination of local and global AD is demonstrated and compared with the benchmark schemes.

1) **Stage 1** PCA is adopted for the feature selection from the dataset. To ascertain the optimal feature selection, the False Positive Rate (FPR) of the HADIoT corresponding to different numbers of feature selected is shown in Fig. 3. It is evident that the FPR of the framework can be minimized by the selection of the first 8 features out of the 20 features ranking in the decreasing order of eigenvalues. The 8 features with relatively larger eigenvalues are considered as the optimal feature selection which includes the data type, payload, source port, destination port, protocol type, source byte, destination byte, and direction. Since each data packet is reduced to 8 features from 20 features, the communication overhead from the desktop to the DELL server decreases by 60%. Therefore, the communication overhead from the edge to the cloud in the framework is reduced to 33.81GB from 84.45GB.

2) **Stage 2** In the training section, the parameters $\Theta = \{W^l, W^u, W^v, V^l\}$ are learnt. By tuning the threshold probability $\theta$, we can determine the anomalies when the data likelihood is below the threshold. Specifically, when adjusting the the probability threshold $\theta_l$ from 1 to 0, various matches of $TPR$ and $FPR$ are acquired, depicted as the Receiver Operating Characteristic (ROC) curve in Fig. 4, and the optimal $\theta_l \in [0, 1]$ is ascertained when the local Precision is maximized.

Then, the local AD is applied to the testing dataset. The data of each time slot is considered as the input of the GRU model, by which the probability of the input data is calculated. Further, the local anomaly is determined by the probability threshold. As can be seen from Fig. 5, the GRU method achieves the highest $TPR$ and precision, compared with the benchmark schemes. Specifically, the GRU method yields an $FPR$ as low as 3.86% while it achieves a $TPR$ at 96.10%.

3) **Stage 3** The global AD via CRF is simulated on a DELL server with Xeon Silver 4116 @2.1GHz and 128GB RAM. Similar to the training procedures of local AD, the CRF model is trained by the same training set to learn the model parameters. We compute the anomaly score of each data packet by setting the threshold $\theta_g$. Specifically, by adjusting the probability threshold $\theta_g$ from 1 to 0, the ROC curve which depicts the $TPR$ and $FPR$ corresponding to different $\theta_g$ is acquired, as shown in Fig. 6. The threshold $\theta_g$ is ascertained when the Precision attains at the peak value.

4) **Overall Performance** Combining the local and global AD result as mentioned in Equation (3), the overall performance of the HADIoT can be achieved. By jointly consider the time serial characteristics and the inter-correlations of various types of IoT data, the overall framework is largely
enhanced, compared with GRU alone. This is mainly because the CRF can detect some anomalies that GRU cannot. For example, some data falls into the normal pattern of its device, but its correlation with other device are obviously absurd. For instance, if the thermometer reads 0 ◦C under the standard atmosphere pressure and the humidity is 90%, it is likely that there is an anomaly. As can be seen from Fig. 7, the comparison of the HADIoT, LSTM, CNN and KNN demonstrates the effectiveness of our proposed method. The HADIoT achieves the highest TPR along with the lowest FPR among the four methods. Specifically, the HADIoT achieves a TPR at 98.12% and a FPR at 4.53%, much better than the other three methods. Because of the combination of GRU and CRF, the HADIoT outperforms the benchmark schemes to a large extent.

Due to the extra computational task introduced by the global AD, the total running time of the HADIoT ascends in a reasonable amount, as depicted in Fig. 8, in comparison with LSTM, CNN and KNN. It can be seen that the processing time of the HADIoT is slightly shorter than LSTM, but longer than CNN and KNN.

VII. CONCLUSION

In this article, we presented HADIoT: a hierarchical anomaly detection framework for IoT devices, which provides a solution for anomaly detection using both local and global anomaly characteristics. We demonstrated the efficiency of HADIoT by testing on the Information Security Center of Excellence (ISCX) 2012 dataset, and HADIoT achieved the True Positive Rates of 98.12% with only 4.53 False Positive Rate. Simulation results show that the proposed framework achieves high effectiveness and outperforms the benchmark schemes. In the future, we aim to develop an autonomous training approach of the AD framework, which trains the framework adaptively according to the change of data patterns.

REFERENCES

[1] J. Li, W. Liang, M. Huang, and X. Jia, “Providing reliability-aware virtualized network function services for mobile edge computing,” in Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS), Jul. 2019, pp. 732–741.
[2] M. Injadat, F. Salo, A. B. Nassif, A. Essex, and A. Shami, “Bayesian optimization with machine learning algorithms towards anomaly detection,” in Proc. IEEE Global Commun. Conf. (GLOBECOM), Dec. 2018, pp. 1–6.
[3] N. Laptev, S. Amizadeh, and I. Flint, “Generic and scalable framework for automated time-series anomaly detection,” in Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD), 2015, pp. 1939–1947.
[4] T. Luo and S. G. Nagarajan, “Distributed anomaly detection using autoencoder neural networks in WSN for IoT,” in Proc. IEEE Int. Conf. Commun. (ICC), May 2018, pp. 1–6.
[5] M. Xie, J. Hu, S. Han, and H.-H. Chen, “Scalable hypergrid k-NN-based online anomaly detection in wireless sensor networks,” IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 8, pp. 1661–1670, Aug. 2013.
[6] F. Chernogorov, J. Turkca, T. Ristaniemi, and A. Averbuch, “Detection of sleeping cells in LTE networks using diffusion maps,” in Proc. IEEE 73rd Veh. Technol. Conf. (VTC Spring), May 2011, pp. 1–5.
[7] L. Lyu, J. Jin, S. Rajasegarar, X. He, and M. Palaniswami, “Fog-empowered anomaly detection in IoT using hyperellipsoidal clustering,” IEEE Internet Things J., vol. 4, no. 5, pp. 1174–1184, Oct. 2017.
[8] B. L. Bars and A. Kalogeratos, “A probabilistic framework to node-level anomaly detection in communication networks,” in Proc. INFOCOM Conf. Comput. Commun., Apr. 2019, pp. 2188–2196.
[9] M. O. Ezeme, Q. H. Mahmoud, and A. Azim, “Hierarchical attention-based anomaly detection model for embedded operating systems,” in Proc. IEEE 24th Int. Conf. Embedded Real-Time Comput. Syst. Appl. (RTCSA), Aug. 2018, pp. 225–231.
[10] A. Yahyaoui, T. Abdellatif, and R. Attia, “Hierarchical anomaly based intrusion detection and localization in IoT,” in Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC), Jun. 2019, pp. 108–113.
[11] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, “Diot: A self-learning system for detecting compromised IoT devices,” CoRR, vol. abs/1804.07474, pp. 1–12, Apr. 2018.
[12] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” 2014, arXiv:1412.3555. [Online]. Available: http://arxiv.org/abs/1412.3555
[13] L. Bottou, “Large-scale machine learning with stochastic gradient descent,” in Proc. COMPSTAT. Cham, Switzerland: Springer, 2010, pp. 177–186.
[14] J. Lafferty, A. McCallum, and F. Pereira, “CRF: Probabilistic models for segmenting and labeling sequence data,” in Proc. 18th Int. Conf. Mach. Learn. (ICML), 2001, pp. 282–289.

HAOTIAN CHANG lives in Nanjing, Jiangsu, China. He received the B.S. degree from Nanjing University, Nanjing, in 2013, and the M.S. degree from PLAUST, in 2016. He is currently pursuing the Ph.D. degree with the National University of Defense Technology, China. His research interests include information and communication technology and wireless sensor networks.

FIGURE 7. Performance of the HADIoT.

FIGURE 8. Runtime comparison.
JING FENG (Member, IEEE) was born in Nanjing, Jiangsu, China, in 1962. She received the Ph.D. degree from Southeast University, Nanjing, in 2000. She is currently a Professor with the National University of Defense Technology, China. Her research interest includes information system integration.

CHAOFAN DUAN lives in Nanjing, Jiangsu, China. He received the bachelor’s degree from the Nanjing University of Science and Technology, Nanjing, in 2014, and the master’s degree from PLAUST, in 2017. He is currently pursuing the Ph.D. degree with the National University of Defense Technology, China. His research interest includes modeling and analysis of the wireless sensor networks.

* * *