Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 3678493, 10 pages
https://doi.org/10.1155/2022/3678493

Research Article

A Novel Intrusion Detection Method Based on Supplement Gate Recurrent Unit for IoT

Zi-yi Liu[1,2], **Chang-song Yang**[1,2], **Jun Xiao**[1,2], **Bo-wen Song**[1,2] and **Ke-xing Shi**[3]

[1]*Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Guilin 541004, China*
[2]*Guangxi Cooperative Innovation Centre of Cloud Computing and Big Data, Guilin University of Electronic Technology, Guilin 541004, China*
[3]*School of Electronic Engineering and Automation, Guilin University of Electronic Technology, Guilin 541004, China*

Correspondence should be addressed to Chang-song Yang; csyang@guet.edu.cn

Received 7 April 2022; Revised 14 July 2022; Accepted 2 August 2022; Published 22 August 2022

Academic Editor: Ruinian Li

With the rapid development of information technology, the internet of things (IoT) technology has been integrated into most people's daily life and work. However, the IoT must confront many new security challenges. Specifically, the increase in the variety of IoT-connected devices has diversified the network. Meanwhile, the high data rates and spectral efficiency offered by 5G cellular networks facilitates the increasing capacity of IoT network traffic. Therefore, network traffic data are characterized by an expanded large scale, wide diversity, and high dimensions, which greatly affects the functionality and efficiency of intrusion detection methods. Although the existing neural network-based intrusion detection methods partially resolve the above problems, they need to execute a lot of nonlinear transformations when learning and characterizing data, resulting in a large loss of feature information. To address this problem, in this paper, we first design a new neural network model based on the gate recurrent unit (GRU), namely, the supplement gate recurrent unit (SGRU). Compared with a traditional GRU, through loss compensation, a SGRU can reduce the loss of feature information caused by nonlinear transformations when learning and characterizing network traffic data. Then, we adopt the SGRU to propose a novel intrusion detection method to monitor the security of the network. Finally, we developed the corresponding prototype system and verified its performance. The experimental results demonstrate that our proposed intrusion detection method is more accurate than previous intrusion detection methods.

1. Introduction

With the rapid development of information technology, especially the development of the internet of things (IoT) [1, 2] technology, IoT has gradually integrated with people's daily lives, causing significant changes to the traditional ordinary network environment. Different from the traditional ordinary network environment, the IoT is based on a wireless network, seamlessly connecting various objects into the network. This technology helps connect the network world with the physical world. However, the existing wireless network infrastructure is relatively simple, so no fixed self-protection mechanism exists. This leads to weak security of the IoT devices, which makes IoT security protection particularly important.

To protect network security, many network security protection methods are available, such as firewalls [3, 4], vulnerability scanning [5, 6], data encryption [7], and user authentication [8]. Although these methods can achieve security protection in traditional network environments, they are not perfectly suited for IoT network environments. The main reason is that in the IoT network environment, attackers may actively launch various types of attacks through system vulnerabilities, thus entering the computer system and stealing private information. In an effort to solve the aforementioned problems, intrusion detection

has gradually attracted extensive attention both in academia and industry [9, 10].

Intrusion detection is a network security software mechanism that can be used for monitoring of network traffic and provides alerts to network administrators when network traffic data are abnormal. Different from traditional network security protection methods with only passive defenses, intrusion detection is a proactive security protection technology. By deploying intrusion detection, network administrators can control the security threats faced by the IoT network systems in real time. Therefore, intrusion detection is one of the most important parts of network security protection.

Generally, the existing intrusion detection methods can be summarized as statistical-analysis-based intrusion detection methods, time-series-based intrusion detection methods, and machine-learning-based intrusion detection methods. However, in the IoT network environment, network traffic data usually present large-scale and high-dimensional characteristics. Additionally, network traffic data are often time-sequential, which brings unique challenges to existing intrusion detection methods. Specifically, on the one hand, because of high-dimensional and large-scale network traffic data, the existing statistical-based intrusion detection methods need to execute a large amount of calculations, resulting in low time-series-based intrusion detection method detection efficiency. On the other hand, the time-series-based intrusion detection methods only use time as an analysis factor and fail to consider other relevant factors. However, the network traffic data are characterized by randomness and diversity. Hence, the time-series-based intrusion detection methods cannot achieve accurate intrusion detection. Although traditional machine-learning-based intrusion detection methods can improve the efficiency of time-series-based intrusion detection method detection, they also suffer from low accuracy because of the randomness and diversity of the network traffic data.

Deep neural network (DNN) can learn the characteristics of complex and high-dimensional data effectively, which offers a new approach for the implementation of intrusion detection. As far as we know, the existing intrusion detection methods are mainly based on backpropagation neural networks (BPNN). However, they cannot work well for high-dimensional time series. Meanwhile, the existing recurrent neural network (RNN), such as the gated recurrent unit (GRU), often cause a loss of feature information due to the occurrence of many nonlinear transformations when learning and characterizing network traffic data.

1.1. Contributions. In this paper, we examine a practical and challenging problem, finding ways to increase the accuracy of intrusion detection in IoT network environments. Specifically, we design a new deep neural network model, namely, supplement gate recurrent unit (SGRU). Then, we apply the SGRU to design a novel intrusion detection method that can achieve efficient and accurate intrusion detection. Therefore, the three main contributions of this paper are summarized as follows:

(i) Most of the existing neural networks do not work well on high-dimensional time-sequential. Meanwhile, they need to perform a large number of nonlinear transformations, which leads to the problem of feature loss in characterization learning. To address the above problem, based on GRU, we design a new neural network model, namely, SGRU. Compared with the traditional GRU, SGRU can not only learns and characterizes data through the data's time-sequential, but also alleviates the loss of feature information caused by the nonlinear transformations

(ii) We design a SGRU-based intrusion detection method for the IoT network environment. Specifically, we utilize the SGRU to learn the characterization of network traffic data and give a theoretical basis. Our proposed SGRU-based intrusion detection method judges whether the network is in a secure state by analyzing the characteristics of network traffic data. Hence, the network administrators can accurately learn the security threats faced by information and networks systems, enabling them to take effective safeguard in time

(iii) We analyze the time complexity of our proposed SGRU-based intrusion detection method and three other different intrusion detection methods. It can be seen from the comparative analysis results that the time complexity of our proposed SGRU-based intrusion detection method is the same as that of the three other intrusion detection method. Moreover, a prototype system is developed, and a performance evaluation is provided. Compared with the existing intrusion detection methods, it can be seen that the intrusion detection method based on SGRU has a better effect

1.2. Related Work. Because of its prime performance, intrusion detection has been extensively studied both in industry and academia. Generally, the existing intrusion detection methods can be summarized as statistics-analysis-based intrusion detection methods [11], time-series-based intrusion detection methods [12, 13], and machine-learning-based intrusion detection methods [14, 15].

1.2.1. Statistics-Analysis-Based Intrusion Detection Methods. Gu et al. [16] developed a network traffic anomaly detection system that compared current baseline distributions with the entropy value of network traffic utilizations, which could effectively detect network anomalies, such as port scans and different types of synchronous attacks. Mazel et al. [17] introduced a method that combined interclass and subspace clustering result associations to achieve unsupervised network anomaly detections. Song and Liu [18] presented a dynamic k-nearest-neighbor (KNN) distance anomaly detection method based on cumulative storms. Compared with other methods, their method was more effective in anomaly detection. Mohammadi et al. [19] utilized a filter and wrapper to design a method for intrusion detection

using feature selection and clustering algorithm, which improved the intrusion detection performance. Zhou et al. [20] designed a new intrusion detection method, which is implemented by ensemble learning and feature selection techniques. In their method, a heuristic algorithm was used for dimensionality reduction. In addition, they utilized the voting technique and probability distribution of the base learner to identify attacks. Moustafa et al. [21] presented an integrated intrusion detection method to mitigate malicious events, which generated new statistical flow features from the protocol based on the analysis of the latent properties in the network. Unfortunately, the above methods require a lot of mathematical calculations. Therefore, the statistics-analysis-based intrusion detection methods are not efficient and accurate in the face of large-scale, multifeatured network traffic data.

1.2.2. Time-Series-Based Intrusion Detection Methods. Han and Zhang [22] used weighted self-similar parameters for detection in order to achieve network activity anomaly detection. Ye et al. [23] designed an anomaly detection method that was immune to nonstationary time series, which could achieve better evaluation performances by using the Hurst parameter estimation algorithm and the fractional Fourier transform (FRFT) algorithm. Yu et al. [24] improved the anomaly detection method using the autoregressive integrated moving average (ARIMA) model, which was improved for the imbalance and nonstationary characteristics unique to wireless sensor networks (WSNs). Pérez et al. [25] presented a new intrusion detection method. By combining time series analysis and multiplexed networks, their method could calculate the probability of an IP address being an attacker at a specific time. Abaeian et al. [26] designed a time series based intrusion detection method, which utilized the generalized autoregressive moving average (GARMA) method to study time series properties. To effectively reduce the false-positive rate, Bozdal et al. [27] proposed a wavelet-based method, which could localize the behavioral changes in the controller area network (CAN) traffic by analyzing the transmission patterns of a CAN network. However, the time-series-based intrusion detection method only uses time as an analysis factor, resulting in a low accuracy in intrusion detection.

1.2.3. Machine-Learning-Based Intrusion Detection Methods. Gu and Lu [28] presented a naive Bayesian (NB) feature embedding and a support vector machine– (SVM–) based intrusion detection method. Iwendi et al. [29] used the correlation-based feature selection approach to extract data features and then analyzed the dimensionality-reduced data through an integrated classifier, thereby constructing an intrusion detection system. Mittal et al. [30] used the low energy adaptive clustering hierarchy protocol for Levenberg-Marquardt neural networks (LEACH-LMNN) to analyze the network lifetime and the use of the gating mechanisms in wireless sensor networks. Through comparative experiments, it can be seen that this method has improved the detection accuracy. Xiao et al. [31] proposed a convolutional neural network– (CNN–) based intrusion detection method. They first used different dimensionality reduction methods to remove the redundant features of the network traffic data. Then, they utilized CNN to extract features from the data. Devan and Khare [32] designed an intrusion detection method that used XGBoost technology for feature selection and then utilized DNN to classify the network intrusions. Muhammad et al. [33] presented an intrusion detection method based on stacked autoencoders (SAE), which improved the classification accuracy. Imrana et al. [34] proposed an intrusion detection method based on bidirectional long-term and short-term memory (BiLSTM). Although the above methods improved the accuracy of the intrusion detections, they did not consider the loss of the feature information caused by the nonlinear changes in the neural network.

1.3. Organization. We introduce the work in the following Sections of this paper as follows. In Sections 2 and 3, we introduce the structure of GRU and SGRU, respectively. Then, we introduce the implementation of SGRU-based intrusion detection method in Section 4. Subsequently, we present a computational complexity comparison in Section 5. Next, we develop a prototype implementation of our proposed method and conduct comparative experiments in Section 6. Finally, we provide a brief and prospects for future work in Section 7.

2. Gate Recurrent Unit

In 2014, to address the ineffective transfer of long-term memory information and the gradient disappearance in backpropagation, Cho et al. designed a new recurrent neural network, namely, recurrent unit (GRU) [35]. Specifically, a GRU has two gate structure units, the reset gate R_t and update gate Z_t, as shown in Figure 1. The R_t gate is used to control the flow of the hidden state information from the previous moment in the current candidate set to the current moment set of the candidate hidden states. The Z_t gate is used to control how much unrelated content of the current candidate state needs to be forgotten at the previous moment and to determine how much of the current candidate set hidden state is retained.

As shown in Figure 1, in this paper, we use R_t to denote the reset gate and Z_t to denote the update gate. Then, the learning model of GRU can be described as follows.

First, in a GRU, the reset gate and update gate are determined by past information h_{t-1} and current information x_t. Then, the formulas are as follows:

$$R_t = \sigma(W_R \bullet [h_{t-1}, x_t]), \tag{1}$$

$$Z_t = \sigma(W_Z \bullet [h_{t-1}, x_t]). \tag{2}$$

Second, the candidate set of a GRU is controlled by the reset gate, and the formula can be expressed as follows:

$$\widetilde{h_t} = \tanh\left(W_{\tilde{h}} \bullet [R_t \times h_{t-1}, x_t]\right). \tag{3}$$

Third, in the update memory phase, a GRU updates h_t through the following formula:

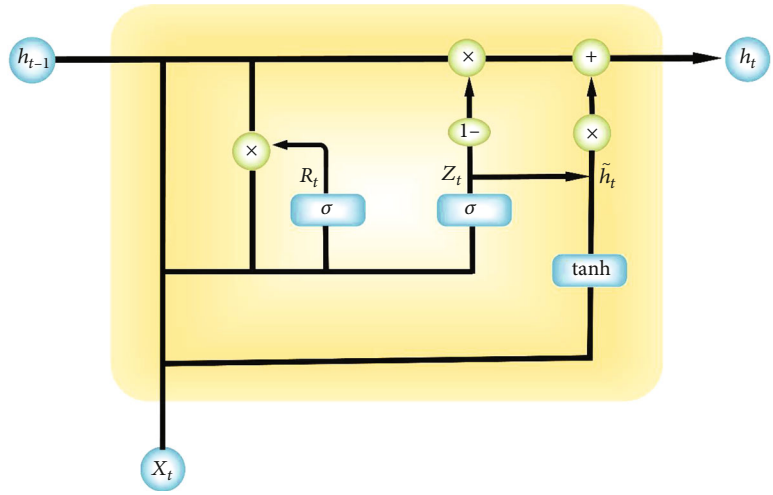

Figure 1: Structure of the GRU model.

$$h_t = (1 - Z_t) \times h_{t-1} + Z_t \times \widetilde{h_t}. \tag{4}$$

Finally, the output of the forward propagation is y_t, which can be computed through the following formula:

$$y_t = \text{softmax}(W_o \bullet h_t). \tag{5}$$

3. Supplement Gate Recurrent Unit

Generally, a GRU is capable of learning and characterization based on the temporal nature of the network traffic data. However, a GRU contains a large number of nonlinear transformations, which might lead to feature information loss in learning and characterizing the network traffic data. Therefore, we design a new neural network model based on GRU, namely, supplement gate recurrent unit (SGRU), as shown in Figure 2. Unlike the original GRU, the SGRU uses the loss compensation principle to alleviate the loss of the feature information when the SGRU executes nonlinear transformations. Therefore, the SGRU has more advantages in learning and characterizing of the IoT network traffic data.

As shown in Figure 2, the SGRU model consists of two GRUs, the OGRU and DGRU. In the SGRU, the OGRU is used for learning and characterizing the input data. Then, the DGRU is used to decode and restore the feature data after learning and characterizing by the OGRU. Without loss of generality, we use x_{input} to represent the input data, x_t to represent the characterizing data learned by the OGRU, x_{out} to represent the data restored by the DGRU, lc to represent the loss data, and lc_{out} to represent the loss compensation data. The specific implementation process of the SGRU is as follows.

First, we use the OGRU to perform the learning and characterization of the input data to obtain x_t. The specific formula is as follows:

$$x_t = OGRU\left(x_{input}\right). \tag{6}$$

Second, we use the DGRU to restore x_t to obtain x_{out}. Then, x_{input} minus x_{out} to obtain the loss data lc. The specific formulas are as follows:

$$\begin{aligned} x_{\text{out}} &= DGRU(x_t), \\ lc &= x_{input} - x_{\text{out}}. \end{aligned} \tag{7}$$

Finally, the loss data lc are subject to learning and characterization through the OGRU again to obtain the loss compensation data lc_{out}. Then, lc_{out} is added to x_t so that the loss of the feature information in x_t is supplemented, and the intrusion detection accuracy can be improved. The formulas are as follows:

$$\begin{aligned} lc_{\text{out}} &= OGRU(lc), \\ x_t &= x_t + lc_{\text{out.}} \end{aligned} \tag{8}$$

In addition, we also use the pseudocode in Algorithm 1 to describe the specific internal implementation process of the above SGRU.

In Algorithm 1, we first input the preprocessed x_{input} to the OGRU for learning and characterizing the x_t. Then, we input x_t into the DGRU for resotring to obtain x_{out}. Subsequently, we subtract the resotredx_{out} from the original input x_{input} and input the resulting loss data lc into the OGRU for learning and characterization to obtain lc_{out}. Finally, we supplement lc_{out} to x_t to obtain the output data x_t after the supplementary learning.

4. Our Proposed Method

In this section, we establish the system model of our proposed SGRU-based intrusion detection method and introduce the method.

4.1. System Model. The design of an accurate intrusion detection method is important for IoT network environments. In particular, with the widespread popularity of cloud

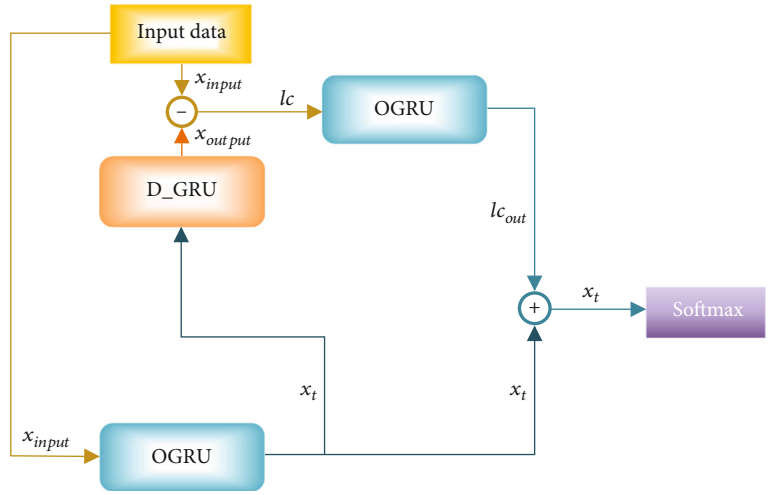

Figure 2: Structure of SGRU.

Input: x_{input}.
Output: x_t.
1 Preprocessing and other operations on x_{input} data.
2 $x_t \leftarrow OGRU(x_{input})$
5 $x_{\text{out}} \leftarrow DGRU(x_t)$
6 $lc \leftarrow (x_{input} - x_{\text{out}})$
7 $lc_{\text{out}} \leftarrow OGRU(lc)$
8 **return** $x_t \leftarrow (x_t + lc_{\text{out}})$

Algorithm 1: SGRU method implementation.

computing [36, 37], IoT [38, 39], and wireless networks [40–42], network traffic data are instilled with the characteristics of diversity, series timing, randomness, and high dimensionality, which creates many new problems for the existing intrusion detection methods. Specifically, it directly affects the accuracy and universality of the intrusion detection methods. Moreover, there are a large number of nonlinear transformations in neural networks. Hence, when the network traffic data are learned and characterized, many features are lost. To solve the above problems, we propose a SGRU-based intrusion detection method, whose system model is shown in Figure 3.

In our proposed intrusion detection method, we first design a new recurrent neural network, namely, SGRU. Then, we adopt SGRU to build a new intrusion detection method. Compared with other DNNs, a SGRU not only learns and characterizes network traffic data through time-sequential but also uses the loss compensation mechanism to reduce the feature loss caused by a large number of nonlinear transformations. As a result, the performance of our proposed SGRU-based intrusion detection method is more attractive.

4.2. SGRU-Based Intrusion Detection Method. We use Algorithm 2 to introduce our proposed SGRU-based intrusion detection method in detail.

In Algorithm 2, we utilize $X - train$ as the network data training set, $Y - test$ as the network data test set, R_{label} as the real attack label, n as the training epoch, S as the intrusion detection value, and R to represent the comparison result.

In the above Algorithm 2, we first input $X - train$ into the SGRU model for the training of the SGRU model. Then, the trained SGRU model is obtained through n epoch of training. Subsequently, $Y - test$ is input into the SGRU to obtain the corresponding detection result S. Finally, we compare S with the true label R_{label} and get the comparison result R.

5. Computational Complexity Analysis

We compare the time complexity of our proposed SGRU-based intrusion detection method with GRU, BiLSTM, and SAE-BPNN-based intrusion detection methods in this section and show them using Table 1.

For simplicity, we use m as the input dimension and n as the dimension of the hidden layer. To facilitate the calculation of the time complexity of SGRU, we first calculate the time complexity of GRU. From Formulas (1)-(4) presented in Section 2, we find that for the GRU, the total operations time overhead is $T(3 \times n \times m + 6 \times n^2 + 4 \times n)$, so the time complexity of GRU can be described as $O(\ n^2)$. Subsequently, for the SGRU, the total operations time overhead is $T(3 \times (3 \times n \times m + 6 \times n^2 + 4 \times n) + 2 \times n + 2 \times m)$, so the time complexity of SGRU is the same as that of GRU, which is $O(n^2)$. For LSTM, the total operations time overhead is $T(4 \times n \times m + 7 \times n^2 + 4 \times n,)$ and the time complexity can be expressed as $O(n^2)$. Meanwhile, the total operations time overhead of BiLSTM is $T(2 \times (4 \times n \times m + 7 \times n^2 + 4 \times n))$, and the time complexity can be expressed as $O(n^2)$. Moreover, since both SAE and BPNN are two-layer fully connected layer structures in our reproduction experiments, the total operations time overhead is $T(3 \times n \times m + 3 \times n^2)$, and the time complexity is $O(n^2)$.

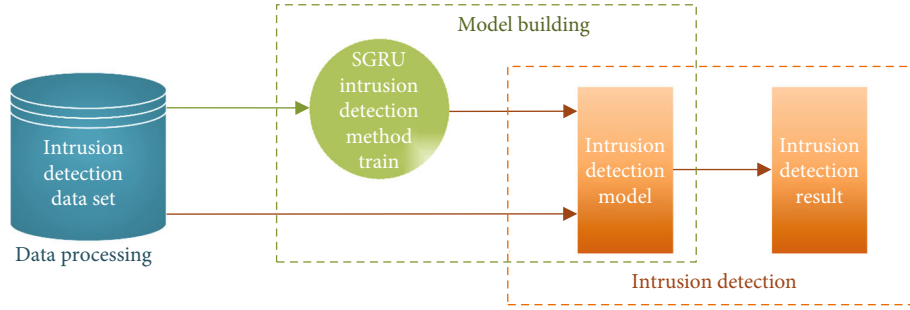

Figure 3: System model.

Input: $X - train$, $Y - test$, R_{label}, n-number of epoch.
Output: intrusion detection results R.
1 Initialize the network dataset.
2 **for** $i = 0$ **to** n **do**
5 $SGRU \leftarrow SGRU_i(X - train)$
6 **end for**
7 $S \leftarrow SGRU(Y - test)$
8 **return** $R \leftarrow Compare(S, R_{label})$

Algorithm 2: SGRU-based intrusion detection method.

6. Experimental Settings

In this section, we first describe the environment, datasets, and experimental standards required for the comparative experiments. Then, we give the results and analysis of the comparative experiments.

6.1. Experimental Environment and Dataset. The desktop hardware devices we used in this experiment mainly include AMD Ryzen 5 3500X CPU, 16G of main memory, and NVIDIA RTX 2060S graphics card. At the same time, this desktop also has Windows10 system, cuDNN7.4.2, and CUDA10.0 driver.

In the experiments of this paper, we use the public dataset UNSW-15NB as the experimental data [43, 44]. The UNSW-15NB dataset includes 2,540,044 pieces of data and 9 different anomaly types. The anomaly types are specifically described as follows:

(1) Analysis: web pages are hacked using tools such as network ports and scripts

(2) Backdoors: a method of attacking through holes in computer reservations or defenses

(3) DoS: using a large-scale traffic attack on the attacker will exhaust the computing power of the computer and make various computer services unusable

(4) Exploits: a means of attacking through vulnerabilities in the attacker's computer system

(5) Fuzzers: an attack method that paralyzes the victim's system by sending a large number of random numbers

(6) Generic: a method suitable for attacking block ciphers

(7) Reconnaissance: an attack that uses probing to gather information about an attack target

(8) Shellcode: an attack method that uses Shell commands to control the victim's host

(9) Worms: the self-replication method increases the computing overhead of the victim's computer, resulting in low computer efficiency and inability to work properly

For simplicity, we randomly intercept 550,000 pieces of data as experimental data, of which 50,000 are used as the test set and 500,000 are used as the training set.

6.2. Experimental Criteria. In the simulation experiments, *Accuracy*, *Precision*, *Recall*, *F1_score*, and *FRR* are used to verify the effectiveness of our proposed SGRU-based intrusion detection method.

True positive (*TP*): The actual is positive, and the detection result is also positive

False positive (*FP*): The actual is negative, while the detection result is positive

True negative (*TN*): The actual is negative, and the detection result is also negative

False negative (*FN*): The actual is positive, while the detection result is negative

Accuracy. The proportion of samples that are correctly detected in the total sample and the calculation formula is as follows:

$$Accuracy = \frac{TN + TP}{TP + TN + FP + FN}. \tag{9}$$

Precision. The proportion of samples that are detected as positive and turn out to be positive and the calculation formula is as follows:

$$Precision = \frac{TP}{FP + TP}. \tag{10}$$

Recall. The proportion of samples that were detected as positive among the samples were actually positive, and the calculation formula is as follows:

Table 1: Time complexity.

	Total operations time overhead	Time complexity
SGRU	$T(9 \times n \times m + 18 \times n^2 + 14 \times n + 2 \times m)$	$O(n^2)$
GRU [35]	$T(3 \times n \times m + 6 \times n^2 + 4 \times n)$	$O(n^2)$
BiLSTM [34]	$T(8 \times n \times m + 14 \times n^2 + 8 \times n)$	$O(n^2)$
SAE-BPNN [33]	$T(3 \times n \times m + 3 \times n^2)$	$O(n^2)$

$$Recall = \frac{TP}{FN + TP}. \tag{11}$$

F1_score. After a comprehensive evaluation of the intrusion detection results using the *Recall* and *Precision* indicators, the calculation formula is as follows:

$$F1_{score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \tag{12}$$

FRR. The proportion of samples that detected negative results among the samples were actually positive, and the calculation formula is as follows:

$$FRR = \frac{FN}{FN + TP}. \tag{13}$$

6.3. Experimental Results and Analysis. We implement our proposed intrusion detection method in this section and compare the method with the experimental results of intrusion detection methods based on GRU [35], BiLSTM [34], and SAE-BPNN [33] analyzed.

6.3.1. Effectiveness Evaluation. We use *Accuracy*, *Precision*, *Recall*, $F1_{score}$, and *FRR* metrics to compare the effectiveness of our proposed method with GRU-based intrusion detection methods, BiLSTM-based intrusion detection methods, and SAE-BPNN-based intrusion detection methods. Figure 4 shows the comparison of the four metrics, *Accuracy*, *Precision*, *Recall*, and *F1_score*.

Figure 4 demonstrates the performance results of four different intrusion detection methods from different perspectives using four different metrics. It is not difficult to see that our proposed SGRU-based intrusion detection method performs better in all aspects compared to the other three methods. Thus, network administrators can more accurately grasp the current network security status, thus greatly enhancing the effectiveness of network security protection.

It can been seen from Figure 5 that our proposed SGRU-based intrusion detection method has lower *FRR* than the other three methods. That is, our proposed method can provide network administrators with less *FRR*. Therefore, our proposed method can reduce the waste of network resources caused by *FRR* while improving the protection performance of the intrusion detection. The reason is that our proposed method considers the loss of the feature information caused by a large number of nonlinear transformations and alleviates this problem by means of a loss compensation.

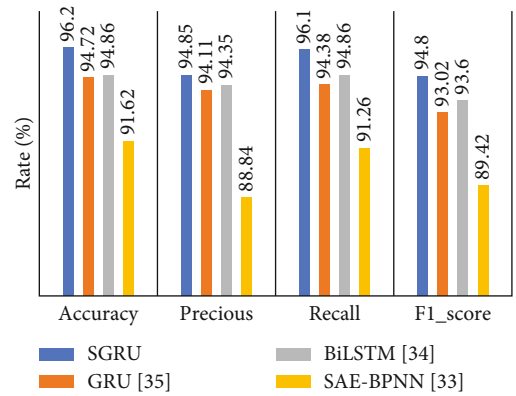

Figure 4: Comparison of effectiveness.

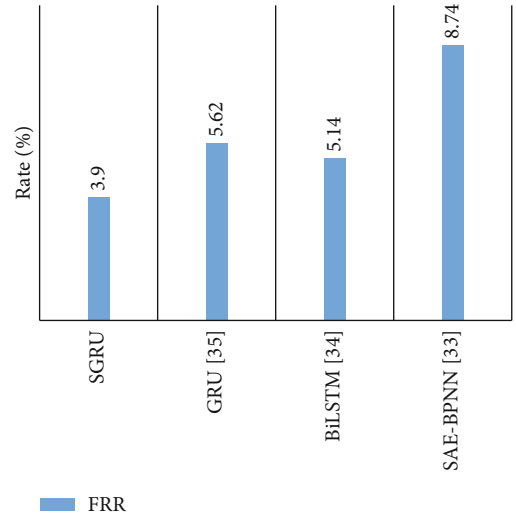

Figure 5: Comparison of *FRR*.

6.3.2. Efficiency Analysis. In this part of the section, we provide an efficiency comparison, as presented in Figures 6 and 7. Figures 6 and 7 show the total time overhead and the time overhead of the test, respectively. We can see that although our proposed method has a higher overhead in time compared to the other three intrusion detection methods, but compared with the improvement in accuracy of our proposed intrusion detection method, this part of the time overhead is acceptable. The experimental results are consistent with the time complexity analysis in Section 5.

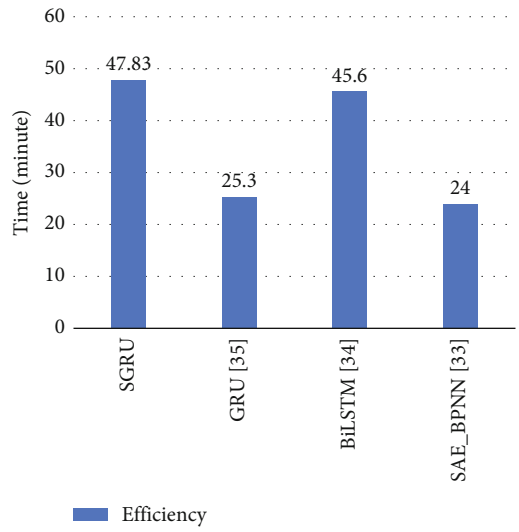

Figure 6: Total time overhead comparison.

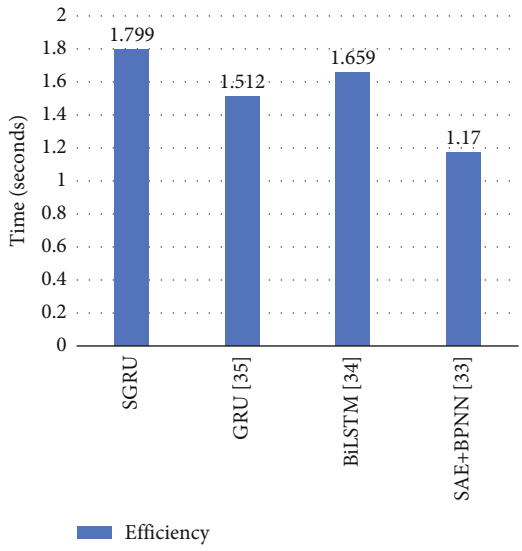

Figure 7: Efficiency comparison of the test.

7. Conclusions and Future Work

In this paper, we studied the design of an accurate intrusion detection method for the IoT network environment. First, we proposed a neural network model named SGRU by improving GRU. Then, we utilized the SGRU to propose a novel intrusion detection method. This method could greatly improve the effectiveness of intrusion detection. Finally, we used simulation experiments to implement our proposed SGRU-based intrusion detection method and evaluated the detection performance. The experimental results showed that compared with some existing intrusion detection methods, our proposed SGRU-based intrusion detection method could achieve a substantial improvement in effectiveness and accuracy.

In the future, we plan to conduct further research on intrusion detection. For example, we will explore the possibility of improving the efficiency of intrusion detection by proposing a simpler neural network structure.

Data Availability

The dataset UNSW-NB15 can be downloaded from https://research.unsw.edu.au/projects/unsw-nb15-dataset.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the Science and Technology Program of Guangxi grant (AD20297028), the Guangxi Key Laboratory of Cryptography and Information Security grant (GCIS202128), and the Natural Science Foundation of Guangxi grant (2020GXNSFBA297132).

References

[1] J. Azar, A. Makhoul, R. Couturier, and J. Demerjian, "Robust IoT time series classification with data compression and deep learning," *Neurocomputing*, vol. 398, pp. 222–234, 2020.

[2] K. N. Qureshi, O. Kaiwartya, G. Jeon, and F. Piccialli, "Neurocomputing for internet of things: object recognition and detection strategy," *Neurocomputing*, vol. 485, pp. 263–273, 2022.

[3] S. Jingyao, S. Chandel, Y. Yunnan, Z. Jingji, and Z. Zhipeng, "Securing a network: how effective using firewalls and VPNs are?," in *Advances in Information and Communication. FICC 2019*, K. Arai and R. Bhatia, Eds., vol. 70 of Lecture Notes in Networks and Systems, Springer, Cham, 2019.

[4] V. H. Dixit, S. Kyung, Z. Zhao, A. Doupé, Y. Shoshitaishvili, and G.-J. Ahn, "Challenges and preparedness of SDN-based firewalls," in *Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*, pp. 33–38, Tempe, Arizona, USA, 2018.

[5] D. Esposito, M. Rennhard, L. Ruf, and A. Wagner, "Exploiting the potential of web application vulnerability scanning," in *ICIMP 2018 the Thirteenth International Conference on Internet Monitoring and Protection*, pp. 22–29, Barcelona, Spain, 2018.

[6] N. Schagen, K. Koning, H. Bos, and C. Giuffrida, "Towards Automated Vulnerability Scanning of Network Servers," in *Proceedings of the 11th European Workshop on Systems Security*, pp. 1–6, Porto, Portugal, 2018.

[7] M. H. Saracevic, S. Z. Adamovic, V. A. Miskovic et al., "Data encryption for internet of things applications based on catalan objects and two combinatorial structures," *IEEE Transactions on Reliability*, vol. 70, no. 2, pp. 819–830, 2021.

[8] R. Almadhoun, M. Kadadha, M. Alhemeiri, M. Alshehhi, and K. Salah, "A user authentication scheme of IoT devices using blockchain-enabled fog nodes," in *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, pp. 1–8, Aqaba, Jordan, 2018.

[9] C. Iwendi, J. H. Anajemba, C. Biamba, and D. Ngabo, "Security of things intrusion detection system for smart healthcare," *Electronics*, vol. 10, no. 12, article 1375, 2021.

[10] M. T. Nguyen and K. Kim, "Genetic convolutional neural network for intrusion detection systems," *Future Generation Computer Systems*, vol. 113, pp. 418–427, 2020.

[11] J. Aitchison, "The statistical analysis of compositional data," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 44, no. 2, pp. 139–160, 1982.

[12] C. W. J. Granger and M. J. Morris, "Time series modelling and interpretation," *Journal of the Royal Statistical Society. Series A (General)*, vol. 139, no. 2, pp. 246–257, 1976.

[13] D. S. Broomhead and R. Jones, "Time-series analysis," *Proceedings of the Royal Society of London A Mathematical and Physical Sciences*, vol. 423, no. 1864, pp. 103–121, 1989.

[14] M. I. Jordan and T. M. Mitchell, "Machine learning: trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[15] T. G. Dietterich, "Machine-learning research," *AI Magazine*, vol. 18, no. 4, pp. 97–97, 1997.

[16] Y. Gu, A. McCallum, and D. Towsley, "Detecting anomalies in network traffic using maximum entropy estimation," in *Proceedings of the 5th ACM SIGCOMM conference on Internet measurement - IMC '05*, p. 32, California, USA, 2005.

[17] J. Mazel, P. Casas, Y. Labit, and P. Owezarski, "Sub-space clustering, inter-clustering results association & anomaly correlation for unsupervised network anomaly detection," in *2011 7th International Conference on Network and Service Management*, pp. 1–8, Paris, France, October 2011.

[18] R. Song and F. Liu, "Real-time anomaly traffic monitoring based on dynamic k-NN cumulative-distance abnormal detection algorithm," in *2014 IEEE 3rd International Conference on Cloud Computing and Intelligence Systems*, pp. 187–192, Shenzhen, China, November 2014.

[19] S. Mohammadi, H. Mirvaziri, M. Ghazizadeh-Ahsaee, and H. Karimipour, "Cyber intrusion detection by combined feature selection algorithm," *Journal of Information Security and Applications*, vol. 44, pp. 80–88, 2019.

[20] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Computer Networks*, vol. 174, article 107247, 2020.

[21] N. Moustafa, B. Turnbull, and K. R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4815–4830, 2018.

[22] J. Han and J. Z. Zhang, "Network traffic anomaly detection using weighted self-similarity based on EMD," in *2013 Proceedings of IEEE Southeastcon*, pp. 1–5, Jacksonville, USA, 2013.

[23] X. Ye, J. Lan, and W. Huang, "Network traffic anomaly detection based on self-similarity using FRFT," in *2013 IEEE 4th International Conference on Software Engineering and Service Science*, pp. 837–840, Beijing, China, 2013.

[24] Q. Yu, L. Jibin, and L. Jiang, "An improved ARIMA-based traffic anomaly detection algorithm for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2016, Article ID 9653230, 2016.

[25] S. I. Pérez, S. Moral-Rubio, and R. Criado, "A new approach to combine multiplex networks and time series attributes: building intrusion detection systems (IDS) in cybersecurity," *Chaos, Solitons & Fractals*, vol. 150, article 111143, 2021.

[26] V. Abaeian, A. Abdullah, T. Pillai, and L. Cai, "Intrusion detection forecasting using time series for improving cyber defence," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 3, no. 1, pp. 28–33, 2015.

[27] M. Bozdal, M. Samie, and I. K. Jennions, "WINDS: a wavelet-based intrusion detection system for controller area network (CAN)," *IEEE Access*, vol. 9, pp. 58621–58633, 2021.

[28] J. Gu and S. Lu, "An effective intrusion detection approach using SVM with naïve Bayes feature embedding," *Computers & Security*, vol. 103, article 102158, 2021.

[29] C. Iwendi, S. Khan, J. H. Anajemba, M. Mittal, M. Alenezi, and M. Alazab, "The use of ensemble models for multiple class and binary class classification for improving intrusion detection systems," *Sensors*, vol. 20, no. 9, p. 2559, 2020.

[30] M. Mittal, C. Iwendi, S. Khan, and A. R. Javed, "Analysis of security and energy efficiency for shortest route discovery in low-energy adaptive clustering hierarchy protocol using Levenberg-Marquardt neural network and gated recurrent unit for intrusion detection system," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 6, article 3997, 2021.

[31] Y. Xiao, C. Xing, T. Zhang, and Z. Zhao, "An intrusion detection model based on feature reduction and convolutional neural networks," *IEEE Access*, vol. 7, pp. 42210–42219, 2019.

[32] P. Devan and N. Khare, "An efficient XGBoost–DNN-based classification model for network intrusion detection system," *Neural Computing and Applications*, vol. 32, no. 16, pp. 12499–12514, 2020.

[33] G. Muhammad, M. S. Hossain, and S. Garg, "Stacked autoencoder-based intrusion detection system to combat financial fraudulent," *IEEE Internet of Things Journal*, 2020.

[34] Y. Imrana, Y. Xiang, L. Ali, and Z. Abdul-Rauf, "A bidirectional LSTM deep learning approach for intrusion detection," *Expert Systems with Applications*, vol. 185, article 115524, 2021.

[35] K. Cho, M. B. Van, C. Gulcehre et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, https://arxiv.org/abs/1406.1078.

[36] C. Yang, F. Zhao, X. Tao, and Y. Wang, "Publicly verifiable outsourced data migration scheme supporting efficient integrity checking," *Journal of Network and Computer Applications*, vol. 192, article 103184, 2021.

[37] C. Yang, X. Tao, F. Zhao, and Y. Wang, "Secure data transfer and deletion from counting bloom filter in cloud computing," *Chinese Journal of Electronics*, vol. 29, no. 2, pp. 273–280, 2020.

[38] E. Tanghatari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Distributing DNN training over IoT edge devices based on transfer learning," *Neurocomputing*, vol. 467, pp. 56–65, 2022.

[39] Y. Lu, S. Wu, Z. Fang, N. Xiong, S. Yoon, and D. S. Park, "Exploring finger vein based personal authentication for secure IoT," *Future Generation Computer Systems*, vol. 77, pp. 149–160, 2017.

[40] W. Guo, N. Xiong, H. C. Chao, S. Hussain, and G. Chen, "Design and analysis of self-adapted task scheduling strategies in wireless sensor networks," *Sensors*, vol. 11, no. 7, pp. 6533–6554, 2011.

[41] X. Wang, Q. Li, N. Xiong, and Y. Pan, "Ant colony optimization-based location-aware routing for wireless sensor networks," in *Wireless Algorithms, Systems, and Applications. WASA 2008*, Y. Li, D. T. Huynh, S. K. Das, and D. Z. Du, Eds., vol. 5258 of Lecture Notes in Computer Science, pp. 109–120, Springer, Berlin, Heidelberg, 2008.

[42] R. Wan, N. Xiong, and N. T. Loc, "An energy-efficient sleep scheduling mechanism with similarity measure for wireless sensor networks," *Human-centric Computing and Information Sciences*, vol. 8, no. 1, pp. 1–22, 2018.

[43] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, pp. 1–6, Canberra, Australia, November 2015.

[44] 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," *Information Security Journal: A Global Perspective*, vol. 25, no. 1-3, pp. 18–31, 2016.