Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.047143

ARTICLE

ResNeSt-biGRU: An Intrusion Detection Model Based on Internet of Things

Yan Xiang[1,2], Daofeng Li[1,2,*], Xinyi Meng[1,2], Chengfeng Dong[1,2] and Guanglin Qin[1,2]

[1]School of Computer and Electronics Information, Guangxi University, Nanning, 530004, China

[2]Guangxi Colleges and Universities Key Laboratory of Multimedia Communications and Information Processing, Guangxi University, Nanning, 530004, China

*Corresponding Author: Daofeng Li. Email: ldf-0123@gxu.edu.cn

Received: 26 October 2023 Accepted: 01 March 2024 Published: 25 April 2024

ABSTRACT

The rapid expansion of Internet of Things (IoT) devices across various sectors is driven by steadily increasing demands for interconnected and smart technologies. Nevertheless, the surge in the number of IoT device has caught the attention of cyber hackers, as it provides them with expanded avenues to access valuable data. This has resulted in a myriad of security challenges, including information leakage, malware propagation, and financial loss, among others. Consequently, developing an intrusion detection system to identify both active and potential intrusion traffic in IoT networks is of paramount importance. In this paper, we propose ResNeSt-biGRU, a practical intrusion detection model that combines the strengths of ResNeSt, a variant of Residual Neural Network, and bidirectional Gated Recurrent Unit Network (biGRU). Our ResNeSt-biGRU framework diverges from conventional intrusion detection systems (IDS) by employing this dual-layered mechanism that exploits the temporal continuity and spatial feature within network data streams, a methodological innovation that enhances detection accuracy. In conjunction with this, we introduce the PreIoT dataset, a compilation of prevalent IoT network behaviors, to train and evaluate IDS models with a focus on identifying potential intrusion traffics. The effectiveness of proposed scheme is demonstrated through testing, wherein it achieved an average accuracy of 99.90% on the N-BaIoT dataset as well as on the PreIoT dataset and 94.45% on UNSW-NB15 dataset. The outcomes of this research reveal the potential of ResNeSt-biGRU to bolster security measures, diminish intrusion-related vulnerabilities, and preserve the overall security of IoT ecosystems.

KEYWORDS

Internet of Things; cyberattack; intrusion detection; internet security

1 Introduction

The ubiquitous deployment, smart capabilities, and interconnected nature of the Internet of Things, augmented by rapid advancements in optical transport networks [1,2], have led to IoT devices becoming integral in various sectors, including those utilized by end-users [3], terminal manufacturers [4] and developers [5]. However, the expansive growth of IoT has introduced significant threats, resulting in numerous security intrusion events in IoT. Threats range from spoofing attacks, denial of service (DoS) attacks, distributed DoS (DDoS) attacks, jamming, eavesdropping, and malware [6,7].

Such intrusions can result in severe outcomes, including information leakage, fraudulent charging, malicious advertising inserts, and even loss of control over the server [8,9]. Information security threats related to IoT devices have ramifications that surpass the confines of individual devices or localized networks, profoundly influencing the larger digital ecosystem, which includes not only interconnected IoT appliances but also websites, servers, and cloud-based services [10]. So, the security of IoT is of great significance for the reliable and stable operation of the entire Internet and needs to be addressed urgently.

To fortify the security of IoT, IDS is commonly employed in IoT. As the primary defense mechanism for IoT security [11,12], IDS monitors network attacks [13], recognizes intrusion traffic, and generates management reports [14]. IDS ensures the confidentiality, integrity, availability, security, and privacy of IoT traffic [15,16]. The accuracy of detection largely dictates the efficacy of an IDS; insufficient accuracy compromises the system's ability to prevent intrusions effectively.

To enhance detection accuracy and recognize potential intrusion traffic, this paper proposes ResNeSt-biGRU, an intrusion detection framework that combines ResNeSt and biGRU. The proposed scheme addresses two issues: (1) improving detection accuracy, which is vital for the practical application of the model; and (2) identifying potential intrusion traffic to preemptively combat intrusions. Considering that intrusions often commence with an "information collection" phase where early-stage traffic is non-damaging, we introduce the PreIoT dataset to encapsulate traffic from this preliminary phase. To examine the effectiveness of our proposed scheme, we have compared it against various algorithms that were recently presented in the literature, utilizing N-BaIoT and PreIoT datasets. Our experiments demonstrate that the results from ResNeSt-biGRU are comparable to those of other research efforts on the N-BaIoT dataset; it also surpasses the performance of these alternative models when evaluated on the PreIoT dataset. The main contributions of this paper can be summarized as follows:

1. Enhanced detection rate. By integrating the advantages of ResNeSt and biGRU, our model adeptly extracts both the temporal and spatial feature map of traffic to improve detection accuracy.

2. Detection of potential attacks: Employing the PreIoT dataset, the ResNeSt-biGRU framework demonstrates the ability to recognize both potential threats.

The structure of this paper is as follows: Section 2 shows the related works; Section 3 introduces the related knowledge of the intrusion detection model of IoT; the model is illustrated in Section 4; Section 5 presents the experiment and makes an analysis of the results; in the end, a summary of the whole paper in Section 6.

2 Related Work

A review of recent work is presented in this section. Research has concentrated on machine learning-based and hybrid algorithm-based intrusion detection models in IoT networks.

Machine learning-based solutions have been a focus for classification in IoT intrusion detection research work. Hussain et al. [17] proposed a Residual Network (ResNet) to detect DoS and DDoS intrusion traffic in IoT networks. They achieved an accuracy score of 99.99% on the CICDDoS2019 [18] dataset, though their scheme required extensive preprocessing to convert data into pictures. Almiani et al. [19] designed a multi-layer recurrent neural network (RNN) for IoT. This scheme had high sensitivity on the NSL-KDD [20] dataset to DoS attack, with a 98.27% detection rate to DoS attack but gained an overall accuracy score of 92.18%. Saurabh et al. [21] proposed a semi-supervised Deep Learning approach, which utilized Semi-supervised Generative Adversarial Networks (SGAN) for IoT botnet detection on the N-BaIoT dataset. It should be noted, however, that inaccuracies in the labeled data could affect models learning patterns. Okur et al. [22] examined 23 different machine learning models on the N-BaIoT regarding detection accuracy, such as Naive Bayes, Logistic, Hoeffding Tree, Random Forest, Random Tree, and so on. The Random Forest model demonstrated the highest correct detection rate at 99.92%.

Hybrid intrusion detection models combining multiple algorithmic approaches can often improve performance by leveraging the strengths of different algorithms, effectively reducing false positive rates and enhancing detection accuracy [23]. Saba et al. [24] proposed an integrated classifier combining a Support Vector Machine and a Decision Tree for IoT intrusion detection. They used a bootstrap aggregating approach to merge the two classifiers, obtaining an accuracy of 99.8% on the NSL-KDD dataset [20]. However, the NSL-KDD dataset is not specifically focused on IoT intrusion traffic. Cao et al. [25] put forward an IoT intrusion detection framework based on a Convolutional Neural Network (CNN) and Gate Recurrent Unit (GRU). They addressed the imbalance of positive and negative samples in the original dataset with a hybrid sampling algorithm. The scheme was assessed using the UNSW_NB15 [26], NSL-KDD [20], and CIC-IDS2017 [27] datasets, with a classification accuracy of 86.25%, 99.69%, 99.65%, respectively, illustrating improvements in classification accuracy and class balance. Javeed et al. [28] introduced a deep-learning Software Defined Network-enabled intelligent framework and a hybrid classifier, Long Short-Term Memory (LSTM)-GRU and Bidirectional LSTM (Cu-LSTMGRU + Cu-BLSTM). The proposed model achieved a high detection accuracy with low false-positive rate on N-BaIoT datasets [29]. Liu et al. [30] presented the combination of CNN and LSTM model, CNN-LSTM for IoT intrusion detection. Their study confirmed that the cascading CNN and LSTM model methods were more stable than the separate CNN and LSTM methods, achieving a 99.98% dichotomous classification accuracy rate on the N-BaIoT dataset. A summary table comparing methods for IoT intrusion detection is given in Table 1.

Table 1: Comparison of methods for IoT intrusion detection

Work	Year	Scheme	Dataset	Evaluation metrics	Attack
[17]	2020	ResNet	CICDDoS2019	accuracy, precision, recall, F1 value	DoS, DDoS
[19]	2020	RNN	NSL-KDD	accuracy, precision, false positive rate, false negative rate, F1 value, Mathew correlation coefficient, Cohens' Kappa coefficient	denial-of-service, remote to the local, user to root, surveillance or probe
[21]	2022	SGAN	N-BaIoT	accuracy	botnet attack for IoT
[22]	2023	23 different machine-learning models	N-BaIoT	accuracy	botnet attack for IoT

(Continued)

Table 1 (continued)

Work	Year	Scheme	Dataset	Evaluation metrics	Attack
[24]	2021	an integrated classifier of Support Vector Machine and Decision Tree	NSL-KDD	accuracy, confusion matrix	denial-of-service, remote to the local, user to root, surveillance or probe
[25]	2022	CNN and GRU	UNSW_NB15, NSL-KDD and CIC-IDS2017	accuracy, precision, recall, F1 value	fuzzers, analysis, backdoors, generic, shellcode, worm, Dos, user to root, surveillance or probe, brute force, heart-bleed, botnet
[28]	2022	Cu-LSTMGRU + Cu-BLSTM	N-BaIoT	precision, recall, accuracy, and F1 value, confusion matrix	botnet attack for IoT
[30]	2023	CNN-LSTM	N-BaIoT	accuracy, precision, recall, F1 value	botnet attack for IoT

3 Preliminaries

In this section, some background of ResNeSt and GRU network will be reviewed before detailing ResNeSt-biGRU construction.

3.1 ResNeSt

ResNeSt [31] is a Convolutional Neural Network and an extension of ResNet [32]. It aims to enhance the network's expressive power and optimize its performance across various visual tasks. ResNeSt introduces 'Split-Attention' blocks to the ResNet architecture, providing an efficient and powerful feature extraction and representation mechanism. This modification enables ResNeSt to deliver high-precision performance across various computer vision tasks, particularly those involving a large number of categories and complex scenarios. The performance of ResNeSt has been proved to often surpass those of the original ResNet model and other variants such as Selective Kernel Network (SKNet), Squeeze-and-Excitation Network (SENet), etc. At present, the downstream work in the field of computer vision (such as object detection and image segmentation) still chooses ResNet as the backbone network [33]. ResNeSt model uses the foundational building blocks of the ResNet model, thus allowing seamless integration into many existing downstream applications [31].

3.2 The Bidirectional GRU

GRU [34] is a special version of RNN. LSTM and GRU are the most Common RNNs. GRU is a variant of LSTM. GRU model is simpler and faster to compute, which can save a significant amount of time, especially when a dataset is large [35]. In traditional RNNs, information flows only in one direction, from the past to the future. BiGRU comprises two one-way GRU layers: One layer processes the sequence from start to end, while the other processes it from end to start. That is, biGRU enables information to traverse in both directions, allowing them to handle not only past information but also future information. This bidirectional capability makes GRUs particularly effective and powerful in processing sequential data.

3.3 Detection Model

Combining with the ResNeSt and the biGRU, this paper presents a hybrid algorithm-based intrusion detection model. It is important to note that both the ResNeSt and biGRU are feasible for IoT intrusion detection. Furthermore, the proposed model not only has a formidable capacity for feature extraction but also holds significant potential, providing robust support for future research. The specific feasibility analysis is as follows:

1. The same goal. ResNeSt, biGRU, and IoT intrusion detection models share a common goal, which is to make classifications. The primary function of ResNeSt and biGRU are commonly used for classification. The goal of deploying an IoT intrusion detection model is to detect attack traffic flows and distinguish attack traffic flows from normal traffic flows.

2. Consistent input formats. During the data preprocessing stage, different types of data are standardized and uniformly converted into tensor formats that are suitable for neural networks, enabling subsequent training.

3. Intrusion traffic can be viewed as a sequence. For example, in the context of intrusion detection, an individual traffic flow may be benign, but a pattern of similar flows could indicate a coordinated attack. Without considering context, intrusion detection models may suffer from low accuracy and high false positive rates.

4 The Proposed Model

In this section, our proposed detection model will be demonstrated in detail. The overall detection process is shown in Fig. 1: Firstly, the raw dataset transformed is processed through some steps such as standardization and feature selection. The processed dataset is then wrapped appropriately for use with the model. Afterward, the ResNeSt-biGRU model is trained and tested. Finally, two fully connected layers and an active layer are used to complete the classification task.

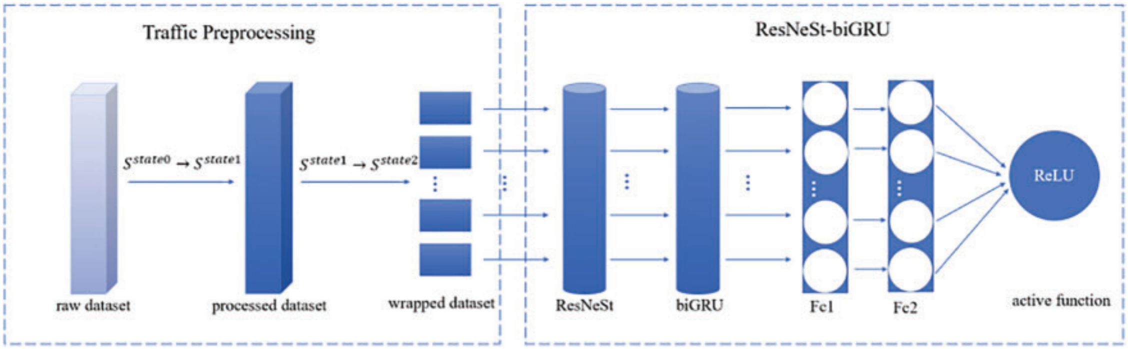

Figure 1: Detection model architecture

Define 1. The entity state is defined as:

$$S^{state0} = \{S_1, S_2, \ldots, S_n\} \tag{1}$$

where S^{state0} represents the raw traffic dataset of IoT. $S_1 = \{s_1, s_2, \ldots, s_n\}, n \in N^*$ and s_i represent the features of traffic flow.

Define 2. The categories of IoT traffic are defined as:

$$FT = \{NT, IT_1, IT_2, \ldots, IT_n\}, n \geq 1 \tag{2}$$

where NT represents normal traffic of IoT, and IT_i represents various intrusion traffic types of IoT. The relationship between NT and IT_i is as follows:

$$NT \cup \cup_{i=1}^{N^*} IT_i = S^{state0}, NT \cap \cup_{i=1}^{N^*} IT_i = \varnothing \tag{3}$$

Since the cost of the detector is related to the size of S^{state0}, reasonable actions are taken to generate entity state flow to reduce computational complexity.

Define 3. The state transition of traffic flow is defined as:

$$SF = \left\{S^{temp} \middle| S^{state0} \rightarrow S^{state1} \rightarrow S^{state2}\right\} \tag{4}$$

There are two state transitions: $S^{state0} \rightarrow S^{state1}$ and $S^{state1} \rightarrow S^{state2}$. The raw dataset is transformed into the processed dataset after $S^{state0} \rightarrow S^{state1}$. The state transition $S^{state1} \rightarrow S^{state2}$ involves converting the dataset into a format suitable for the model. S^{state2} represents the final output, which is then used in the subsequent ResNeSt-biGRU stage.

4.1 Traffic Preprocessing

The first state transition, $S^{state0} \rightarrow S^{state1}$, includes Z-score Normalization and Random Forest. The data in each column from the raw dataset exhibits significant dispersion and covers a wide numerical range, which may lead to troubles, such as long training time, inability to converge, and low accuracy. Then, Z-score normalization is used to address the issue of the numerical span. Another problem of the raw dataset is a redundancy of features, accordingly, opting for a Random Forest for feature selection. Models with fewer features will be improved in time consumption and as well as in accuracy. The transformation applied to the dataset is portrayed in Fig. 2. Originally, the maximum difference in data values reached about 1,000,000,000. However, following the data processing—which included normalization and feature selection—this difference was reduced to a scale of approximately 4, and many irrelevant features were removed.

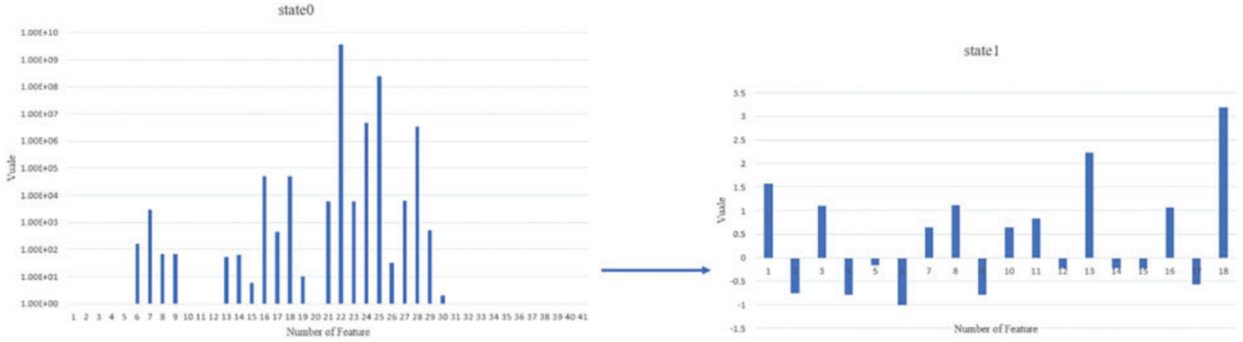

Figure 2: Changes in datasets from $\boldsymbol{S^{state0}} \rightarrow \boldsymbol{S^{state1}}$

The last state transition, $S^{state1} \rightarrow S^{state2}$, is encapsulating dataset because the neural network exclusively accepts input in a Specified format. The Torch.utils.data.TensorDataset and

Torch.utils.data.DataLoader functions are used to convert the dataset into an acceptable format for the model. Initially, the features and class of the training set are separated into two different files. Then, convert the two files into tensor format, respectively. It is noteworthy that PyTorch defaults to use 32-bit float (single-precision) tensors for training due to their balance between precision and memory usage, whereas 16-bit float tensors may lose precision and 64-bit float tensors, despite offering little accuracy improvement, consume considerably more memory. Subsequently, the Torch.utils.data.The TensorDataset function is used to combine the feature tensors and label tensors into a single dataset. At long last, torch.utils.data.DataLoader is responsible for setting values of the batch size, whether to shuffle the dataset, and the memory size of the results obtained in the previous step. The whole preprocessing algorithm is shown in Algorithm 1.

Algorithm 1: Traffic preprocessing

```
Input: dataset $A = \{(x_1, y_1) \dots (x_N, y_N)\}$; $M$; random number $(a, b)$
Output: new dataset B
$X \leftarrow$ Z-score Normalization(X)
$weight(X_N) \leftarrow$ random forest(A)
for $i = 1 \dots M$ do
    select highest $(weight(X_N))$
end for
$X \leftarrow$ reshape $X$ into ($X$.shape[0], 1, a, b)
$X \leftarrow$ convert to torch float 32 format (from numpy to torch format ($X$))
$Y \leftarrow$ convert to torch float 32 format (from numpy to torch format ($Y$))
$B \leftarrow$ zip ($X$, $Y$) based on tensor dataset function
$B \leftarrow$ Dataloader ($B$, batchsize, shuffle)
return $B$
```

4.2 ResNeSt-biGRU

The main inspiration of the ResNeSt-biGRU algorithm originates from various time-influenced and spatially bound objects and phenomena in the material world. For example, humans, and animals, among others, are all influenced by time and space, occupying specific locations in the physical world and existing over a certain period. For instance, considering tree growth:

1. Trees rely on roots in the soil to absorb nutrients and utilize components like leaves, stems, and other structures to carry out processes such as photosynthesis, reproduction, and propagation. These tasks require physical space.

2. Trees require time to mature and bear fruit.

The growth and development of trees are influenced by the interaction of time and space, inspiring me to multidimensional feature extraction. Extracting features from both spatial and temporal dimensions carries a metaphorical similarity to understanding different aspects of data. First of all, the dataset itself has spatial and physical characteristics, which differentiate each data. The process of extracting useful data or information from the dataset naturally involves extracting features from space. Secondly, feature extraction from the time dimension considers the influence of the antecedent traffic and the subsequent traffic. In the ResNeSt-biGRU algorithm, ResNeSt is deployed to extract spatial feature maps, and biGRU is employed to extract temporal feature maps. This combination allows for an enriched representation of information from the original data, enhancing the expressiveness of the original data. Specifically, ResNeSt-biGRU enhances IoT traffic feature extraction by:

1. Utilizing Split-Attention block in ResNeSt for superior spatial feature discernment.

2. Leveraging biGRU's ability to integrate temporal context, considering both antecedent and subsequent IoT traffic in its analysis.

First, in the spatial dimension, the importance of the Split-Attention block and shortcut connection is emphasized, as Fig. 3 shows. Split-Attention block is a core computational unit. This structure consists of group convolution and a channel-wise soft attention mechanism. The IoT traffic is divided into different groups, allowing for feature learning independently across different feature maps. Group convolution is grouping input IoT traffic by channel dimension twice: *cardinality* and *radix*. The $cardinality_k, k \in (1, 2, \ldots, K)$ is the number of groups in the grouped convolution and $radix_r, r \in (1, R)$ is the branches within each $cardinality_k$ group that will individually undergo convolution (con).

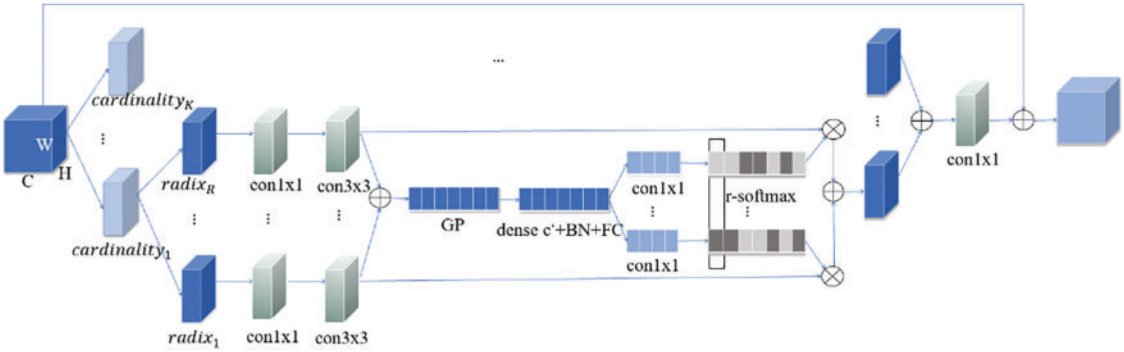

Figure 3: ResNeSt architecture

These separate branches then undergo global average pooling (GP) to capture the holistic information present before proceeding through a sequence of dense, batch normalization (BN) and fully connected (FC) layers. This sequence enables the model to learn the feature map. Subsequently, the softmax function calculates the attention weight for the R feature maps to strengthen the feature representation ability. The resulting weights from softmax indicate the significance of each feature group, which in turn informs the model of the prioritized features for the given task. The final stage involves fusing the soft assignment weights with the corresponding feature map, followed by concatenating these weighted results along the channel dimension and combining them with the shortcut connection. This fusion is mathematically executed as outlined in Eqs. (5) and (6).

$$V_c^k = \sum\nolimits_{i=1}^{R} a_i^k\left(r\right) U_{R(k-1)+i} \tag{5}$$

$$Y = V + T(X) \tag{6}$$

where V_c^k represents the weighted fusion of $c'th$ channel, $a_i^k\left(r\right)$ represents the soft assignment weight of $c'th$ channel, $U_{R(k-1)+i}$ represents the groups of $c'th$ channel. Y represents the ultimate result, and V is the concatenated weighted fusion result. Function T is a strided convolution or a combined convolution-with-pooling and X represents the input IoT traffic.

Second, in the temporal dimension, ResNeSt-biGRU focuses on the relationship between the previous time, the current time, and the subsequent time. As visualized in Fig. 4, a GRU cell, which is employed as the primary functional unit, applies a gating mechanism to discern pertinent data for retention from that which should be excluded from IoT traffic.

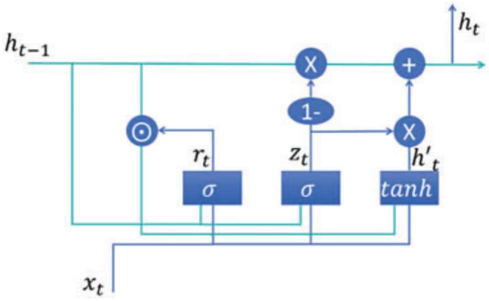

Figure 4: GRU cell

Update gate (z_t) regulates how much previous state information is retained. A value approaching 1 directs the model to preserve more of the former hidden state, mitigating the vanishing gradient issue by carrying forward past information. Conversely, a value near 0 induces greater forgetfulness. Reset gate (r_t) influencing the extent to which the model should forget previous state information, allowing for the discarding of irrelevant data. As the value approaches 0, the network forgets more information from the previous state information. Candidate hidden state (h'_t) blends current input with past state details of past candidate hidden state, tempered by tanh non-linearity, thereby managing gradients and enriching model expressivity. Current hidden state (h_t), is a weighted sum of the hidden state at the previous hidden state and the current candidate hidden state, modulated by the update gate. The interactions of the update gate, reset gate, candidate hidden state, and current candidate hidden state are numerically formulated in Eqs. (7)–(10), respectively.

$$z_t = \sigma\left(W_{zh} \cdot h_{t-1} + W_{zx} \cdot x_t\right) \tag{7}$$

$$r_t = \sigma\left(W_{rh} \cdot h_{t-1} + W_{rx} \cdot x_t\right) \tag{8}$$

$$h'_t = tanh\left(W_{hh} \cdot \left(r_t \odot h_{t-1}\right) + W_{hx} \cdot x_t\right) \tag{9}$$

$$h_t = \left(1 - z_t\right) \odot h_{t-1} + z_t \odot h'_t \tag{10}$$

where W_{zh}, W_{zx}, W_{rh}, W_{rx}, W_{hh}, W_{hx} are weight matrix. $t - 1, t$ stand for the previous moment and current moment, respectively. The operator $\odot$ represents Hadamard product. σ is a sigmoid active function.

As depicted in Fig. 5, biGRU comprises four distinct layers: Input layer, forward propagation layer, reverse propagation layer, and output layer. The forward layer processes the data in sequence—from the inaugural to the final time step—preserving past data influences and producing a forward hidden state hf_t. Conversely, the reverse propagation layer begins with the sequence's final time step, progressing inversely to retain future data implications, generating a backward hidden state hb_t. At each sequential step, dual hidden states are generated: One from the forward propagation layer and another from the reverse. As Eq. (11) shows, the corresponding states are typically combined, such as through concatenation, ensuring each temporal output integrates preceding and subsequent information.

$$h_t = hf_t + hb_1 \tag{11}$$

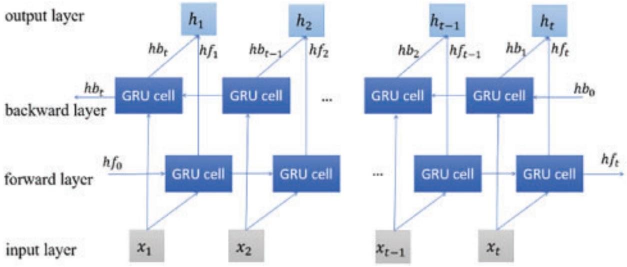

Figure 5: BiGRU architecture

Finally, two fully connected layers and a Rectified Linear Unit (ReLU) activation function are implemented. The fully connected layers map the feature vectors, extracted by ResNeSt-biGRU, to the label value. Afterward, the activation function classifies based on the probabilities assigned to each node in the output generated by the fully connected layer. The ReLU activation function, the mathematical form of which is specified in Eq. (12), features simple computation and rapid execution. With inputs greater than 0, the derivative of the input remains constant at 1, which effectively prevents the gradient from diminishing, thereby mitigating the issue of gradient vanishing to some degree [36]. Algorithm 2 shows the flow of the ResNeSt-biGRU algorithm.

$$f(x) = max(0, x) \tag{12}$$

Algorithm 2: ResNeSt-biGRU

```
Input: dataset $B$; cardinality $K$; Radix $R$
Output: $H$
$cardinality_k, k \in (1, 2, \ldots, K) \leftarrow$ divided $B$ in channel dimension
$radix_r, r \in (1, R) \leftarrow$ divide each $cardinality_k$ in channel dimension
$\{F_1, F_2, F_3 \ldots F_G\}, G = KR \leftarrow$ each $radix$ block
$U_i \leftarrow F_i(X), i \in \{1, 2, 3 \ldots G\}$
$U'^k \leftarrow \sum_{j=R(k-1)+1}^{RK} U_j, j \in \{R(k-1)+1, r(k-1)+2 \ldots RK\}, U'^k \epsilon R^{X \times Y \times Z/K}$, where $X \times Y \times Z$ are the block output feature sizes.
$s_c^k \leftarrow \frac{1}{X \times Y} \sum_{i=1}^{X} \sum_{j=1}^{Y} U_c'^k(i, j)$
$V_c^k \leftarrow \sum_{i=1}^{R} a_i^k(c) U_{R(k-1)+i}, a_i^k(c) \leftarrow \begin{cases} \frac{\exp(\theta_i^c(s^k))}{\sum_{j=0}^{R} \exp(\theta_i^c(s^k))}, R > 1 \\ \frac{1}{1 + \exp(-\theta_i^c(s^k))}, R = 1 \end{cases}$, $\theta_i^c$ represents the weight of the $radix_c$ determined by $s_c^k$.
$V \leftarrow concat\{V_1, V_2, \ldots, V_K\}$, concatenate $V$ along the channel dimension
$O \leftarrow V + T(X)$, $T$ is a strided convolution or combined convolution with pooling.
$zf_t \leftarrow \sigma(W_{zhf} \cdot hf_{t-1} + W_{zxf} \cdot o_t)$ //$o_t$ is the information entered at the current time.
$rf_t \leftarrow \sigma(W_{rhf} \cdot hf_{t-1} + W_{rxf} \cdot o_t)$ //$W_{zhf}$, $W_{zxf}$, $W_{rhf}$, $W_{rxf}$ are weight matrices.
$h'f_t \leftarrow tanh(W_{hhf} \cdot (rf_t \odot hf_{t-1}) + W_{hxf} o_t)$ //$W_{hhf}$, $W_{hxf}$ are weight matrices.
$hf_t \leftarrow (1 - zf_t) \odot hf_{t-1} + zf_t \odot h'f_t$ //output from the forward layer
$zb_t \leftarrow \sigma(W_{zhb} \cdot hb_{t-1} + W_{zxb} \cdot o_1)$ //$\sigma$ is the sigmoid activation function
```

(Continued)

Algorithm 2 (continued)

```
$rb_t \leftarrow \sigma\,(W_{rhb} \cdot hb_{t-1} + W_{rxb} \cdot o_1)$ //$W_{zhb}, W_{zxb}, W_{rhb}, W_{rxb}$ are weight matrices.
$h'b_t \leftarrow tanh\,(W_{hhb} \cdot (rb_t \odot hb_{t-1}) + W_{hxb}o_1)$ //$W_{hhb}, W_{hxb}$ are weight matrices.
$hb_t \leftarrow (1 - zb_t) \odot hb_{t-1} + zf_t \odot h'b_t$ //output from backward layer
$h_t \leftarrow concat\{hb_1, hf_t\}$
$H \leftarrow \{h_1, h_2, \ldots, h_t\}$
return $H$
```

5 Experiment and Analysis of the Proposed Scheme

In this section, some experiments are conducted to evaluate the performance of the proposed ResNeSt-biGRU.

5.1 Datasets

Two datasets are used to validate the model: The N-BaIoT and the PreIoT dataset. The N-BaIoT dataset is well-known in IoT intrusion detection. The PreIoT dataset is captured using Wireshark.

5.1.1 N-BaIoT

The dataset comprises traffic from 9 different IoT devices, recording both normal and botnet traffic. Table 2 details the amount of traffic for each set.

Table 2: The traffic selection from the N-BaIoT dataset

	Normal flows	Botnet flows
Train set	40,000	360,000
Test set	10,000	90,000

5.1.2 PreIoT

The PreIoT dataset is collected by the Wireshark tool from a total of four IoT devices, including a rice cooker, printer, smart switch (controlling devices like a projector, a curtain lift in a computer classroom), and a camera. The PreIoT traffic is categorized into four types: Normal traffic (BenTra), sensitive directory scanning (SenTra), sniffing hosts or services (SniTra), and botnet traffic (BonTra). Sniffing hosts or services and sensitive directory scanning are the prevalent techniques for information gathering.

Over the span of one hour, a total of 2,916,560 traffic flows were collected using the Wireshark tool. From these captured data, 41 features are extracted using the Tshark tool from the original pcapng (Packet Capture Next Generation) format file. Table 3 shows the traffic numbers of each set.

Table 3: The traffic selection from the PreIoT dataset

	Normal flows	SniTra	SenTra	BonTra
Train set	48,000	120,000	80,000	40,000
Test set	13,000	30,000	19,000	10,000

5.1.3 UNSW-NB15

The original UNSW-NB15 dataset consists of a hybrid of real modern normal activities and synthetic contemporary attack behaviors. It covers nine attack types, with each instance comprising 47 feature attributes and a single label attribute. Considering the challenges in comparisons, primarily binary or selective type classification, we have tailored our comparative dataset to align with the attack types used in the most recent relevant study [37]. Consistent with work [37], our dataset includes five categories: Normal traffic along with four attack traffic types. Among these four categories of attacks, Dos and Reconnaissance represent more prevalent attacks, while Shellcode and Worms are indicative of the rarer attacks. The specifics of the dataset utilized in our study are delineated in the Table 4 below.

Table 4: The traffic selection from the UNSW-NB15 dataset

	Normal	Dos	Reconnaissance	Shellcode	Worms
Train set	56,000	12,000	12,000	1,200	120
Test set	1,000	1,000	1,000	300	50

5.2 Evaluation Method

To evaluate the effectiveness of the method objectively from multiple perspectives, four evaluation indicators are employed in this paper, which are accuracy, precision, recall, and F1 score.

In binary classification, accuracy (acc) represents the proportion of records in the total sample where the model's classification results match the true labels. Precision (pre) signifies the probability of correctly predicting intrusion instances among all samples predicted as an intrusion. Recall indicates the probability of correctly predicting intrusion instances among all true intrusion samples. The F1 score is used to calculate the Harmonic mean of precision and recall. The False Positive Rate (FPR) is the proportion of incorrect intrusion traffic predictions out of all the actual benign traffic. The calculation formula of accuracy (acc), precision (pre), recall, and F1 score are shown in Eqs. (13)–(17), respectively. In multi-classification, taking PreIoT as an example, when calculating metrics for the BenTra class, the other three classes are considered collectively, and so forth.

$$acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{13}$$

$$pre = \frac{TP}{TP + FP} \tag{14}$$

$$recall = \frac{TP}{TP + FN} \tag{15}$$

$$F1\,score = \frac{2 * pre * recall}{pre + recall} \tag{16}$$

$$FPR = \frac{FP}{FP + TN} \tag{17}$$

where TP, TN, FN, and FP represent true positives, true negatives, false negatives, and false positives, respectively.

5.3 Experimental Results and Analysis

5.3.1 Comparation 1: PreIoT dataset

To demonstrate the effectiveness of the hybrid ResNeSt-biGRU scheme in improving detection accuracy, this paper conducts a comparative analysis, comparing it with its component architectures—ResNeSt and GRU—as well as with the enriched variant, biGRU. To ensure the fairness of the comparisons, identical training and testing datasets are utilized for each model.

The PreIoT dataset categorizes network traffic into four types for more accurate identification of potential security threats, including, but not limited to, traffic associated with widely-used information gathering techniques such as sensitive directory scanning, which aims to discover potential vulnerabilities in network services or applications, and sniffing hosts or services, often malicious, intended to identify active hosts or services within the network, which could result in the collection of further detailed information. The early detection of such malicious activities is crucial as it often signifies the initial preparation work of potential security threats. Therefore, the capabilities provided by the PreIoT dataset in recognizing traffic related to information gathering are vital for defense systems, playing a significant role in defending against future security incidents, as well as providing proactive protection by analyzing and understanding patterns of attack. Table 5 shows the results on PreIoT. The ResNeSt model shows high performance, with a 98.35% accuracy rate, 97.91% precision, 98.44% recall, and 98.10% F1 value. GRU obtains an accuracy of 95.33%, precision of 94.72%, recall of 94.51%, and F1 value of 94.36%. BiGRU achieves higher accuracy than GRU at 95.60%, with corresponding precision, recall, and F1 values of 94.99%, 94.96%, and 94.73%, respectively. The proposed scheme yielded the highest metrics of 99.90% accuracy, 99.86% precision, 99.90% recall, and 99.88% F1 value.

Table 5: The experiment results on PreIoT

Method	Acc (%)	Pre (%)	Recall (%)	F1 value (%)	FPR (%)	Time (s)
ResNeSt	98.35	97.91	98.44	98.10	1.65	117
GRU	95.33	94.72	94.51	94.36	4.67	**114**
biGRU	95.60	94.99	94.96	94.73	4.40	139
Proposed scheme	**99.90**	**99.86**	**99.90**	**99.88**	**0.11**	180

Although the proposed model requires 180 s to process, it achieved an impressive accuracy rate of 99.90%. For the vast scope of IoT network traffic data, completing the training and testing of a 360,000-entry dataset within this timeframe is acceptable. Moreover, FAR is a critical metric in evaluating security system performance. High FPR not only triggers unnecessary alarms, affecting user trust and satisfaction, but also burdens system administrators with additional verification tasks. In environments with numerous IoT devices, this can be especially problematic. The proposed achives FPR of 0.11%, significantly lower than the others.

The outstanding performance of the proposed scheme can be attributed to a thorough methodology that includes traffic preprocessing and model construction. The traffic preprocessing ensures the quality of the data, decreases computational complexity, avoids the risk of overfitting, and transforms the data to better suit the needs of the model. The model combines ResNeSt and BiGRU to significantly advance its ability to fit and represent underlying patterns. ResNeSt is capable of effectively extracting feature maps from traffic data through the use of convolution layers, pooling, and activation functions, employing Split-Attention mechanisms to capture diverse patterns in the data via nonlinear transformations. BiGRU, particularly suited for processing sequential data, enhances context comprehension by considering information in both forward and reverse directions. The combination of ResNeSt and biGRU within ResNeSt-biGRU, combining the strengths of the aforementioned models, facilitates the extraction of more diverse features, leading to improved model accuracy as Table 4 shows. This confirms the efficacy of integrating these two robust neural network architectures for the task of intrusion detection in IoT networks.

Fig. 6 offers the ResNeSt-biGRU confusion matrix. The “true_0”, “true_1”, “true_2”, and “true_3” represent BenTra, SenTra, SniTra, and BonTra, respectively. Corresponding predicted labels are “pred_0”, “pred_1”, “pred_2”, and “pred_3”. Out of 72,000 instances, 71,925 instances of IoT traffic are correctly classified. It achieves 100% accuracy in BenTra and BonTra, 99.62% accuracy in SenTra, and 99.99% accuracy in SniTra. Despite the model’s overall high precision, it shows a relative weakness in identifying SenTra, as 71 out of 75 misclassified instances pertain to SenTra; this indicates a challenge in differentiating it from BenTra due to variable directory paths with no apparent pattern, such as “/config.php”, “/FCKeditor/”, “/password.log” among others. In general, the proposed scheme demonstrates high predictive accuracy and shows promise for practical IoT security application deployment.

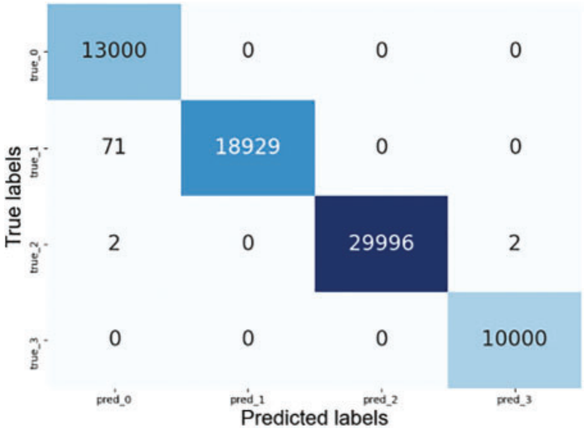

Figure 6: The confusion matrix on the PreIoT dataset by ResNeSt-biGRU

5.3.2 Comparation 2: N-BaIoT Dataset

In this section, we conduct a comparative analysis of our proposal against four recent methodologies evaluated on the N-BaIoT dataset, a common comparison framework employed by most studies.

Table 6 presents the results of the scheme proposed in this paper along with the methods described in references [21,22,28,30] as mentioned in the Related Work section. Reference [21] boasted an impressive accuracy of 99.88% through semi-supervised methods, yet the lack of precision, recall, and F1 score data limits a comprehensive evaluation. High accuracy indicates strong reliability, but without full performance metrics, methodological robustness cannot be fully ascertained. Reference [22] found that the Random Forest model had the best performance out of 23 different models, achieving a noteworthy accuracy of 99.92%. This can be attributed to its decision tree aggregation capabilities, which capture complex feature relationships effectively. Yet, as with the prior method, the absence of additional performance metrics constrains full evaluation. The recall of reference [28] is slightly lower than other metrics, indicating that some positive samples are misrecognized. Reference [30] showed optimal performance, with perfect precision, recall, and F1 value, but the accuracy is lower than this paper's proposed scheme by 0.01%. This paper's proposed scheme achieves marginally superior accuracy in these references at 99.99%, coupled with robust precision, recall, and F1 scores of 99.96%, 99.98%, and 99.97%, respectively. This can be attributed to its effective combination of ResNeSt for robust spatial feature extraction and biGRU for proficient handling of sequential data. This combination allows the model to accurately identify complex patterns typical in IoT intrusion scenarios.

Table 6: The experiment results on the N-BaIoT dataset

Method source	Acc (%)	Pre (%)	Recall (%)	F1 value (%)
Reference [21]	99.88	–	–	–
Reference [22]	99.92	–	–	–
Reference [28]	99.45	99.34	98.49	99.47
Reference [30]	99.98	**100.00**	**100.00**	**100.00**
Proposed scheme	**99.99**	99.96	99.98	99.97

The confusion matrix of ResNeSt-biGRU is displayed in Fig. 7. Rows are labeled with predicted labels, while columns correspond to true labels. The "true_0" and "pred_0" represent normal traffic, while "true_1" and "true_1" represent botnet traffic. Out of 100,000 instances, 99,989 instances of IoT traffic are correctly identified. For normal traffics, the model achieves an accuracy of 99.97%, with only 3 samples misclassified as botnet traffics. In the case of botnet traffics, the proposed scheme achieves an accuracy of 99.99%, with only 8 samples misclassified as normal traffics. affirming notably high-performance metrics of ResNeSt-biGRU and its capability to discriminate between normal and botnet traffic effectively.

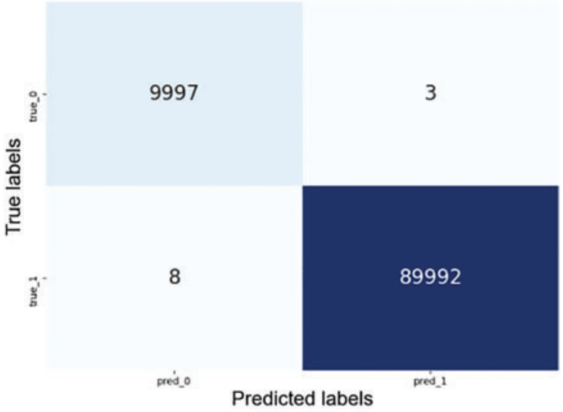

Figure 7: The confusion matrix on the N-BaIoT dataset by ResNeSt-biGRU

5.3.3 Comparation 3: UNSW-NB15 Dataset

Furthermore, we compare the performance of TMG-IDS [37] and MAGENTO [38], alongside the proposed scheme, on the imbalanced UNSW-NB15 dataset. Both TMG-IDS and MAGENTO are intrusion detection methods that have improved their performance by incorporating data augmentation in the preprocessing phase. The experimental outcomes are presented in Table 7.

Table 7: The experiment results on the UNSW-NB15 dataset

	TMG-IDS			MAGENTO			proposed scheme		
	pre	recall	F1 value	pre	recall	F1 value	pre	recall	F1 value
normal	0.9935	0.9498	0.9711	0.9919	0.9492	0.9701	1.0000	0.9940	0.9970
Dos	0.7944	0.9129	0.8496	0.8103	0.9139	0.8590	0.8514	0.9970	0.9185
Recon-naissance	0.6995	0.8616	0.7721	0.7111	0.8581	0.7777	0.9895	0.8510	0.9151
Shell-code	0.4527	0.7090	0.5526	0.3877	0.7354	0.5078	0.9966	0.9667	0.9814
Worms	0.6410	0.5682	0.6024	0.3962	0.4773	0.4330	0.9412	0.6400	0.7619
Macro-	0.7162	0.8003	0.7496	0.6594	0.7868	0.7095	**0.9557**	**0.8897**	**0.9148**

The findings highlight the F1 value is higher when the training samples are abundant but decreases when the sample size is reduced. For normal traffic conditions with approximately 56,000 real samples, TMG-IDS, MAGENTO, and the proposed scheme perform nearly identically, boasting precision rates over 99% and F1 scores above 97%. This result demonstrates that, with a sufficient number of training samples, all three models are capable of learning robust features of normal traffic and accurately applying these characteristics to predict new samples. In contrast, as the sample size contracts to roughly 12,000, a noticeable decrement in model performance is observed, as presented by the results from Dos and Reconnaissance attacks. Although the characteristics of Dos attacks and Reconnaissance traffic can be learned to some extent from a relatively smaller sample set, the models' generalization capabilities are affected as the number of samples decreases. The decline in model efficacy is most severe when addressing Worm attacks, where only about 120 real samples are available. The suggested model accurately ascertained a mere 32 of 50 instances, accentuating the quandary confronted by models in extrapolating from negligible datasets. However, under the conditions of Shellcode attack traffic, despite having only 1,200 real training samples, the proposed solution achieves an unexpectedly high F1 score of 98.14%. This surprising result may be due to the nature of Shellcode attacks, where the series of actions attackers take to gain the highest privileges after obtaining execution control of the shell are highly similar, exhibiting significant consistency in intrusion behavior. This consistent and distinctive characteristic allows for the creation of accurate detection models even with a limited number of samples through learning these clear attack pattern features. If the attack exhibits highly consistent and identifiable patterns, then excellent detection performance can be achieved through proper data processing and model design, even with a smaller sample size.

6 Conclusion

With the rising complexity and frequency of network attacks, IDS has become crucial for safeguarding computer networks and systems from unauthorized access and damage. In this paper, we introduce an IoT intrusion detection model based on ResNeSt and biGRU. This model leverages the spatial feature extraction capabilities of ResNeSt to discern valuable spatial features of IoT device data, while the biGRU component efficiently processes sequential data to analyze dependencies based on preceding and subsequent traffic. The incorporation of these technologies allows the proposed ResNeSt-biGRU model to significantly enhance the accuracy of abnormal traffic detection within IoT environments. Additionally, we present PreIoT, a dataset comprising traffic from the "information gathering" phase, which includes two prevalent types of network probes. Utilizing PreIoT, the model is positioned to monitor potential attacks proactively. Experimental results validate that our ResNeSt-biGRU model surpasses existing detection methods on various standard IoT security datasets, exhibiting a notably high detection rate and a low false-positive rate on both PreIoT and N-BaIoT datasets.

Currently, thresholds of some intrusion detection systems might depend too heavily on subjective assessment, which introduces variability and uncertainty. To overcome this issue, future research might concentrate on developing objective standardization of threshold determination, effectively diminishing the subjectivity and enhancing the dependability of the IDS feature selection. Moreover, to reduce the higher rate of false positives associated with particular types of attacks, such as SenTra of PreIoT, it would be important to refine the detection algorithms. Enhancing these algorithms will ensure they are adaptable to discern varying traffic patterns and minimize instances of false positives.

Acknowledgement: The authors thank everyone who helped during the research and preparation of the article.

Funding Statement: This research was funded by the National Natural Science Foundation of China (No. 61662004).

Author Contributions: The authors confirm contribution to the paper as follows: Study conception and design: Yan Xiang, Daofeng Li; data collection: Yan Xiang; analysis and interpretation of results: Yan Xiang, Xinyi Meng, Chengfeng Dong, Guanglin Qin; draft manuscript preparation: Yan Xiang, Daofeng Li. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The PreIoT is shared at https://github.com/Magret-n/IoT-PreDataset.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] D. T. Hai, "Optical-computing-enabled network: An avant-garde architecture to sustain traffic growth," *Results Opt.*, vol. 13, no. 2, pp. 100504, 2023. doi: 10.1016/j.rio.2023.100504.

[2] D. T. Hai, "Optical networking in future-land: From optical-bypass-enabled to optical-processing-enabled paradigm," *Opt. Quant. Electron.*, vol. 55, no. 10, pp. 864, Jul. 2023. doi: 10.1007/s11082-023-05123-x.

[3] A. Kumari and S. Tanwar, "A reinforcement-learning-based secure demand response scheme for smart grid system," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 2180–2191, Jun. 2021. doi: 10.1109/JIOT.2021.3090305.

[4] A. Kumar *et al.*, "Revolutionary strategies analysis and proposed system for future infrastructure in Internet of Things," *Sustainability*, vol. 14, no. 1, pp. 71, Dec. 2021. doi: 10.3390/su14010071.

[5] K. Koteish, H. Harb, M. Dbouk, C. Zaki, and J. C. Abou, "AGRO: A smart sensing and decision-making mechanism for real-time agriculture monitoring," *J. King Saud. Univ.-Comput. Inf. Sci.*, vol. 34, no. 9, pp. 7059–7069, Jun. 2022. doi: 10.1016/j.jksuci.2022.06.017.

[6] R. Roman, J. Y. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed internet of things," *Comput. Netw.*, vol. 57, no. 10, pp. 2266–2279, Jul. 2013. doi: 10.1016/j.comnet.2012.12.018.

[7] S. Z. Chen, H. Xu, D. Liu, B. Hu, and H. C. Wang, "A vision of IoT: Applications, challenges, and opportunities with China perspective," *IEEE Internet Things J.*, vol. 1, no. 4, pp. 349–359, Jul. 2014. doi: 10.1109/JIOT.2014.2337336.

[8] B. W. Zhao, J. M. Yuan, X. M. Liu, Y. D. Wu, H. H. Pang and R. H. Deng, "SOCI: A toolkit for secure outsourced computation on integers," *IEEE Trans. Inf. Forens. Secur.*, vol. 17, pp. 3637–3648, Oct. 2022. doi: 10.1109/TIFS.2022.3211707.

[9] A. Kumari, S. Tanwar, S. Tyagi, and N. Kumar, "Verification and validation techniques for streaming big data analytics in internet of things environment," *IET Netw.*, vol. 8, no. 3, pp. 155–163, May 2019. doi: 10.1049/iet-net.2018.5187.

[10] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for IoT security based on learning techniques," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 3, pp. 2671–2701, Jan. 2019. doi: 10.1109/COMST.2019.2896380.

[11] E. Biermann, E. Cloete, and L. M. Venter, "A comparison of intrusion detection systems," *Comput. Secur.*, vol. 20, no. 8, pp. 676–683, Dec. 2001. doi: 10.1016/S0167-4048(01)00806-9.

[12] P. Rana *et al.*, "Intrusion detection systems in cloud computing paradigm: Analysis and overview," *Complexity*, vol. 2022, no. 9, pp. 1–14, May 2022. doi: 10.1155/2022/3999039.

[13] N. Scaife, H. Carter, P. Traynor, and K. R. Butler, "CryptoLock (and Drop It): Stopping ransomware attacks on user data," presented at the 2016 Int. Conf. Distrib. Comput. Syst., Nara, Japan, Jun. 27–30, 2016, pp. 303–312.

[14] J. Jabez and B. Muthukumar, "Intrusion detection system (IDS): Anomaly detection using outlier detection approach," *Procedia Comput. Sci.*, vol. 48, no. 5–6, pp. 338–346, 2015. doi: 10.1016/j.procs.2015.04.191.

[15] B. K. Mohanta, D. Jena, U. Satapathy, and S. Patnaik, "Survey on IoT security: Challenges and solution using machine learning, artificial intelligence and blockchain technology," *Internet Things*, vol. 11, no. 8, pp. 100227, Sep. 2020. doi: 10.1016/j.iot.2020.100227.

[16] A. Kumari, S. Tanwar, S. Tyagi, N. Kumar, M. Maasberg and K. R. Choo, "Multimedia big data computing and Internet of Things applications: A taxonomy and process model," *J. Netw. Comput. Appl.*, vol. 124, no. Suppl. C, pp. 169–195, Dec. 2018. doi: 10.1016/j.jnca.2018.09.014.

[17] F. Hussain, S. G. Abbas, M. Husnain, U. U. Fayyaz, F. Shahzad and G. A. Shah, "IoT DoS and DDoS attack detection using ResNet," presented at the 2020 IEEE Int. Multitop. Conf., Bahawalpur, Pakistan, Nov. 5–7, 2020, pp. 1–6.

[18] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," presented at the 2019 Int. Carnahan Conf. Sec. Technol., Chennai, India, Oct. 1–3, 2019, pp. 1–8.

[19] M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, and A. Razaque, "Deep recurrent neural network for IoT intrusion detection system," *Simul. Model. Pract. Theory*, vol. 101, pp. 102031, May 2020. doi: 10.1016/j.simpat.2019.102031.

[20] L. Dhanabal and S. Shantharajah, "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 4, no. 6, pp. 446–452, Jun. 2015. doi: 10.17148/IJARCCE.2015.4696.

[21] K. Saurabh, A. Singh, U. Singh, O. P. Vyas, and R. Khondoker, "GANIBOT: A network flow based semi supervised generative adversarial networks model for IoT botnets detection," presented at the 2022 IEEE Int. Conf. Omni-layer Intell. Syst., Barcelona, Spain, Aug. 1–3, 2022, pp. 1–5.

[22] C. Okur, A. Orman, and M. Dener, "DDOS intrusion detection with machine learning models: N-BaIoT data set," presented at the Int. Conf. Artif. Intell. Appl. Math. Eng., May, 2023, pp. 607–619.

[23] C. Zhi, G. Dong, Y. Wang, Z. Zhu, and Y. Yang, "Trade-off-oriented impedance optimization of chiplet-based 2.5-D integrated circuits with a hybrid MDP algorithm for noise elimination," *IEEE Trans. Circuits Syst.*, vol. 69, no. 12, pp. 5247–5258, Dec. 2022. doi: 10.1109/TCSI.2022.3200410.

[24] T. Saba, T. Sadad, A. Rehman, Z. Mehmood, and Q. Javaid, "Intrusion detection system through advance machine learning for the internet of things networks," *IT Prof.*, vol. 23, no. 2, pp. 58–64, Mar. 2021. doi: 10.1109/MITP.2020.2992710.

[25] B. Cao, C. H. Li, Y. F. Song, Y. Y. Qin, and C. Chen, "Network intrusion detection model based on CNN and GRU," *Appl. Sci.*, vol. 12, no. 9, pp. 4184, Apr. 2022. doi: 10.3390/app12094184.

[26] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," presented at the 2015 Mil. Commun. Inf. Syst. Conf., Canberra, ACT, Australia, Nov. 10–12, 2015, pp. 1–6.

[27] A. Rosay, E. Cheval, F. Carlier, and P. Leroux, "Network intrusion detection: A comprehensive analysis of CIC-IDS2017," presented at the Int. Conf. Inf. Syst. Secur. and Priv., 2022, pp. 25–36.

[28] D. Javeed, T. Gao, M. T. Khan, and D. Shoukat, "A hybrid intelligent framework to combat sophisticated threats in secure industries," *Sens.*, vol. 22, no. 4, pp. 1582, Feb. 2022. doi: 10.3390/s22041582.

[29] Y. Meidan *et al.*, "N-BaIoT—network-based detection of IoT botnet attacks using deep autoencoders," *IEEE Pervasive Comput.*, vol. 17, no. 3, pp. 12–22, Oct. 2018. doi: 10.1109/MPRV.2018.03367731.

[30] Y. R. Liu, Z. M. Lv, and Z. J. Liu, "Research on abnormal traffic detection of Internet of Things based on feature selection," presented at the 2023 Int. Conf. Comput., Netw. Int. Things, Xiamen, China, May, 2023, pp. 576–582.

[31] H. Zhang *et al.*, "Resnest: Split-attention networks," presented at the IEEE/CVF Conf. Compt. Vision Pattern Recognit. Workshops, New Orleans, LA, USA, Jun. 19–20, 2022, pp. 2736–2746.

[32] Z. F. Wu, C. H. Shen, and A. V. D. Hengel, "Wider or deeper: Revisiting the resnet model for visual recognition," *Pattern Recognit.*, vol. 90, no. 3, pp. 119–133, Nov. 2019. doi: 10.1016/j.patcog.2019.01.006.

[33] S. H. Gao, M. M. Cheng, K. Zhao, X. Y. Zhang, M. H. Yang, and P. Torr, "Res2Net: A new multi-scale backbone architecture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 652–662, Apr. 2019. doi: 10.1109/TPAMI.2019.2938758.

[34] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014. doi: 10.48550/arXiv.1412.3555.

[35] J. X. Chen, D. M. Jiang, and Y. N. Zhang, "A hierarchical bidirectional GRU model with attention for EEG-based emotion classification," *IEEE Access*, vol. 7, pp. 118530–118540, 2019. doi: 10.1109/ACCESS.2019.2936817.

[36] G. F. Lin and S. Wei, "Research on convolutional neural network based on improved Relu piecewise activation function," *Procedia Comput. Sci.*, vol. 131, no. 4, pp. 977–984, May 2018. doi: 10.1016/j.procs.2018.04.239.

[37] H. Ding, Y. Sun, N. Huang, Z. Shen, and X. Cui, "TMG-GAN: Generative adversarial networks-based imbalanced learning for network intrusion detection," *IEEE Trans. Inf. Forens. Secur.*, vol. 19, no. 27, pp. 1156–1167, Nov. 2023. doi: 10.1109/TIFS.2023.3331240.

[38] G. Andresini, A. Appice, L. D. Rose, and D. Malerba, "GAN augmentation to deal with imbalance in imaging-based intrusion detection," *Futur. Gener. Comp. Syst.*, vol. 123, no. 5, pp. 108–127, May 2021. doi: 10.1016/j.future.2021.04.017.