Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 1173065, 9 pages
https://doi.org/10.1155/2021/1173065

Research Article

An Efficient Network Security Situation Assessment Method Based on AE and PMU

Xiao-ling Tao[1,2] **Zi-yi Liu**[1,2] **and Chang-song Yang**[1,2]

[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*

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

Received 22 June 2021; Accepted 23 August 2021; Published 13 September 2021

Academic Editor: Zhuojun Duan

Network security situation assessment (NSSA) is an important and effective active defense technology in the field of network security situation awareness. By analyzing the historical network security situation awareness data, NSSA can evaluate the network security threat and analyze the network attack stage, thus fully grasping the overall network security situation. With the rapid development of 5G, cloud computing, and Internet of things, the network environment is increasingly complex, resulting in diversity and randomness of network threats, which directly determine the accuracy and the universality of NSSA methods. Meanwhile, the indicator data is characterized by large scale and heterogeneity, which seriously affect the efficiency of the NSSA methods. In this paper, we design a new NSSA method based on the autoencoder (AE) and parsimonious memory unit (PMU). In our novel method, we first utilize an AE-based data dimensionality reduction method to process the original indicator data, thus effectively removing the redundant part of the indicator data. Subsequently, we adopt a PMU deep neural network to achieve accurate and efficient NSSA. The experimental results demonstrate that the accuracy and efficiency of our novel method are both greatly improved.

1. Introduction

Network security situation assessment (NSSA) technology is one of the most effective active defense technologies to evaluate the threats of network security, by which the network administrators not only can comprehensively understand the security risk situation but also can understand the security threats which are faced by the current network and information system. Hence, the network administrators can manage the dynamic network security situation and judge the development trend of the network security situation [1, 2]. As a result, NSSA attracts increasing attention.

Nowadays, plenty of NSSA methods have been proposed, while there still exist many inherent deficiencies in the existing methods, resulting in a lot of severe challenges that need to be solved solidly. Firstly, most of the existing NSSA methods pay too much attention to subjective judgment and prior knowledge. Meanwhile, they ignore plenty of other external factors and the time sequence property of the indicator data. As a result, they are not suitable for the long-term assessment over the network security situation. Secondly, the existing NSSA methods are not efficiency attractive. Specifically, the current indicator data is characterized by large scale, multifeature, heterogeneity, high dimensionality, and nonlinearity. Hence, the existing NSSA methods need to pay very expensive computational overhead to process these indicator data before evaluating the network security situation. Last but not least, some existing NSSA methods only consider part of the network security threats and attacks. However, the current network threats and attacks are characterized by diversity and randomness, resulting in very low accuracy in some existing methods.

Therefore, how to design a novel method to effectively and accurately achieve NSSA has become one of the most important problems in the field of network security.

1.1. Contributions. We study an essential but challenging problem in this paper, i.e., accurate and efficient NSSA in large-scale network environment. Then, we design a novel NSSA method based on the autoencoder (AE) and parsimonious memory unit (PMU), which can efficiently and accurately achieve NSSA. Therefore, the main contributions of this paper can be described as the following two aspects.

(1) In the current large-scale network environment, the indicator data is characterized by heterogeneity, large scale, multifeature, high dimensionality, and nonlinearity. Therefore, we design an AE-based data dimensionality reduction method to process the original indicator data, thus reducing the dimensions of the indicator data. Subsequently, we can efficiently extract the situation assessment elements on the premise of guaranteeing the integrity of the data features

(2) We adopt PMU to design a novel NSSA method for the current large-scale network environment, in which PMU is utilized for feature representation and time-varying learning of situation assessment elements. Meanwhile, we offer the theoretical computational complexity comparison. Finally, we implement our novel method and provide the performance evaluation, which can intuitively demonstrate the high efficiency and accuracy of our novel method

1.2. Related Work. NSSA has been extensively studied in both academia and industry, resulting in a rich body of solutions. Generally speaking, the existing NSSA methods can be summarized into three categories: mathematical statistics-based assessment method, knowledge reasoning-based assessment method [3], and machine learning-based assessment method [4, 5].

Wang et al. [6] designed a hierarchical NSSA method based on an analytic hierarchy process (AHP), which used the hierarchical cyber threat situation assessment (CSA) indicator system constructed by AHP to decide the network threat weight values. Wang et al. [7] proposed an AHP-based NSSA and quantification method, which utilized the AHP and hierarchical situation assessment model to simplify the NSSA problem. Li et al. [8] adopted fuzzy optimal clustering criteria combined with c-means clustering to process the indicator data, thus getting the number of clusters and the optimal clustering center. Then, they got the final NSSA results by utilizing AHP to construct an assessment model. Zhang et al. [9] presented a distributed denial of service (DDoS) attack NSSA model based on fuzzy clustering algorithm fusion features, which can effectively evaluate the security status of DDoS attack. Although the above methods can effectively implement NSSA, they not only need a lot of subjective judgments but also are not conducive to long-term assessment.

Yi et al. [10] designed a NSSA method based on fuzzy theory. Their method used fuzzy theory to weaken the index factors with low credibility and eliminate the uncertainty, thus making the assessment results more accurate. Liu et al. [11] utilized D-S evidence theory to fuse the measured indexes for obtaining the device threat value. Then, they utilized AHP to calculate the weights for different devices and finally obtained the network threat situation value by the weighting method. Codetta-Raiteri et al. [12] designed a NSSA method based on decision networks (DN), which can achieve a reasonable tradeoff between computational complexity and analysis efficiency. To quantitatively assess the network security risk, Wang et al. [13] designed a NSSA model based on the Bayesian approach. Fan et al. [14] presented a security evaluation method based on a software-defined network (SDN), which used multiple observation hidden Markov model (HMM) to obtain the security evaluation value of SDN, by quantifying the network state. To more completely describe the network security situation, Liao et al. [15] designed a NSSA method based on the extended HMM. Although the above knowledge reasoning-based methods can improve the NSSA accuracy, they rely on too much prior knowledge and have no advantages in efficiency.

Nowadays, the support vector machine (SVM) [16] and neural network [17, 18] are also widely used in NSSA. Chen et al. [19] adopted the SVM and gravitational search algorithm (GSA) to design a NSSA method, which has a better global optimization function. Qiang et al. [20] utilized an optimized cuckoo search back propagation neural network (BPNN) to design a new NSSA method. In their method, they used a cuckoo search (CS) algorithm based on conjugate gradient to optimize the initial parameters of BPNN and increase the training efficiency of the neural network. Shi and Chen [21] utilized a dual-SVM model for data learning and parameter estimation in command information system security situation samples, thus evaluating the command information system security situation. Gao et al. [22] designed the SVM information system security risk assessment model, which was optimized by an artificial fish swarm algorithm (AFSA). In their method, the AFSA was used to optimize the SVM, resulting in great accuracy and fast convergence. Han et al. [23] adopted convolutional neural networks (CNN) to design a quantitative network security situation evaluation method for an intelligent robot cluster under the wireless connection. Yang et al. [24] adopted a deep autoencoder (DAE) and deep neural networks (DNN) to study NSSA. Subsequently, they designed a new method to improve the network attack identification accuracy and the NSSA flexibility. Although the above methods can improve the accuracy, they cannot learn the correlation of time series. Therefore, they cannot be suitable for the NSSA over the indicator data which is characterized by the time sequence.

1.3. Organization. The rest of the structure of this paper is as follows. In Section 2, we describe the AE and PMU. In Section 3, we adopt AE and PMU to propose a novel NSSA method. Then, we implement the proposed method and

provide the experiment results in Section 4. Finally, we simply summarize this paper in Section 5.

2. Preliminaries

2.1. Autoencoder. An autoencoder (AE) [25, 26] is a common unsupervised learning algorithm, which utilizes the original input data as a reference for self-supervised learning to reduce dimension. AE generates information elements with more obvious features and lower dimensions than the original data element. AE maps the original data to the encoding layer to achieve encoding, then maps the encoded data to the decoding layer for decoding, and takes the final decoded data as the output data (see Figure 1).

We utilize $X = [X_1, X_2, \cdots, X_l]^T \in R^{l\times 1}$ and $W_e = [W_e^{(1)}, W_e^{(2)}, \cdots, W_e^{(n)}]^T \in R^{n\times 1}$ to represent the input data and the weight of the encoder, respectively. At the same time, we denote by $h_e = [h_e^{(1)}, h_e^{(2)}, \cdots, h_e^{(n)}]^T \in R^{n\times 1}$ the output of the encoding layer. Then, we could utilize $W_d = [W_d^{(1)}, W_d^{(2)}, \cdots, W_d^{(n)}]^T \in R^{n\times 1}$ to represent the weight of the decoder. The output result can be expressed as $X' = [X_1', X_2', \cdots, X_l^t]^T \in R^{l\times 1}$, where l represents the data dimensions. Note that the AE requires that the final input result is almost equal to the output result, that is, $X = X'$.

The encoding process of the AE can be expressed as follows:

$$h_e = f^e(W_e X + b^e), \tag{1}$$

where the bias of the encoding part can be represented as b^e and the activation function of the encoding part can be represented as f^e.

The decoding process of the AE decoding layer can be expressed as follows:

$$X' = f^d\left(W_d h_e + b^d\right), \tag{2}$$

where the bias of the decoding part can be represented as b^d and the activation function of the decoding part can be represented as f^d.

The MSE loss function is usually used in AE training, and it can be expressed as

$$L\left(X, X'\right) = \frac{1}{2k}\sum_{j=1}^{k}\left(X_j - X_j'\right)^2, \tag{3}$$

where X represents the input variable, X' represents the output variable, $L(X, X')$ represents the loss function, and k represents the number of samples.

2.2. Parsimonious Memory Unit. A parsimonious memory unit (PMU) is a new recurrent neural network, which can be viewed as an improved version of a gated recurrent unit (GRU) [27]. PMU is characterized by better managing the latent relations between short- and long-term dependencies [28]. Note that there are two gate structures in the GRU model, i.e., reset gate and update gate. However, there is only one gate structure in PMU, i.e., unit gate, as seen in Figure 2. Specifically, PMU integrates the update gate and the reset gate of GRU into a new unit gate, resulting in fewer parameters in PMU. Moreover, due to the fact that the PMU can better manage the latent relations between short- and long-term dependencies, PMU has better convergence and speed in training.

In Figure 2, we utilize U_t to represent the unit gate, which is used to control the learning of long-term correlation and short-term correlation of the data. When U_t is 1, PMU learns the long-term dependence of the data, while when U_t is 0, PMU learns the short-term dependence of the data. The learning mode of PMU can be described as follows.

Firstly, the state of the unit gate is obtained by the evaluation state h_{t-1} transmitted from the previous node and the input x_t of the current node:

$$U_t = \sigma(W_u \bullet [h_{t-1}, x_t]). \tag{4}$$

Secondly, the state of the current time memorized on the current candidate set $\widetilde{h_t}$ can be expressed as

$$\widetilde{h_t} = \tanh\left(W_{\tilde{h}} \bullet [U_t \times h_{t-1}, x_t]\right). \tag{5}$$

Thirdly, in the stage of updating memory, PMU updates h_t through the following formula:

$$h_t = (1 - U_t) \times h_{t-1} + \widetilde{h_t}. \tag{6}$$

Finally, the output of forward propagation is

$$y_t = \text{softmax}(W_o \bullet h_t). \tag{7}$$

In the forward propagation process of PMU, we need to learn the following three parameters: W_u, $W_{\tilde{h}}$, and W_o, where

$$\begin{aligned} W_u &= W_{ux} + W_{uh}, \\ W_{\tilde{h}} &= W_{\tilde{h}x} + W_{\tilde{h}h}, \\ W_o &= W_o. \end{aligned} \tag{8}$$

Then, the output y_t is the network domain security situation score value.

3. Method

In this section, we initially describe the system structure. Then, we introduce our AE-PMU-based NSSA method in detail.

3.1. System Structure. With the rapid development of modern network technology, the complexity of the current network environment is continuously increasing. In particular, with the development of cloud computing [29–31] and

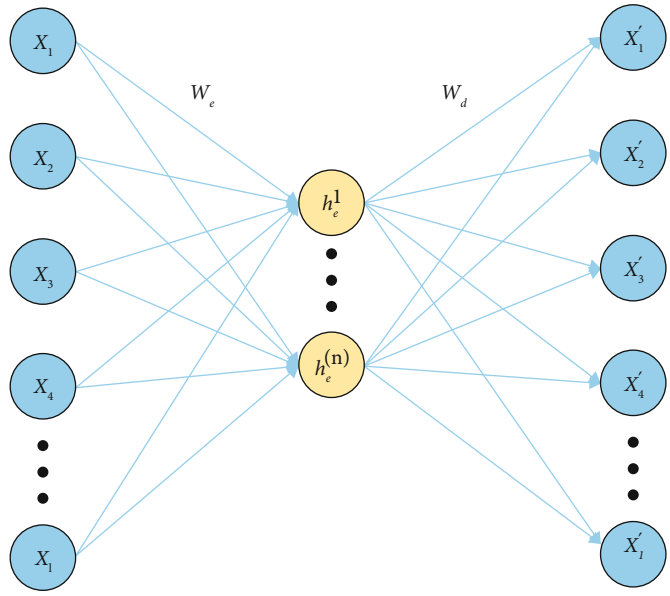

Figure 1: Basic structure of AE.

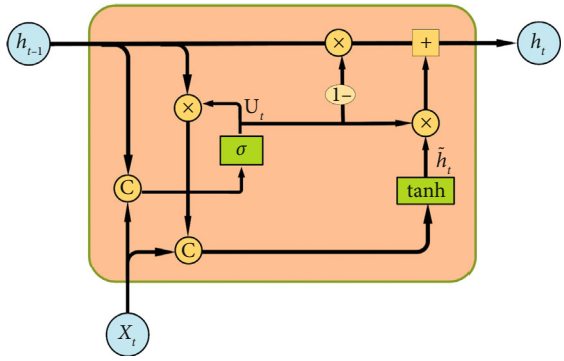

Figure 2: The construction of PMU.

Internet of things [32, 33], the modern network is characterized by new features, such as dynamic virtualized management methods and multilevel service models. As a result, the network threats have the characteristics of diversity and randomness. Meanwhile, the indicator data is large-scale and has heterogeneity, resulting in plenty of new problems in NSSA. Specifically, the large volume of indicator data seriously affects the efficiency of the assessment method. Moreover, the diversity and randomness of network threats directly determine the accuracy and the universality of the assessment method. To handle the above challenges, we propose a novel NSSA method (as seen in Figure 3), which is mainly composed of AE-based data dimensionality reduction and PMU-based assessment method. Specifically, we first adopt AE to process the original indicator data to achieve data dimension reduction. Then, we extract the situation assessment elements efficiently. By taking AE, our method can greatly improve the efficiency and reduce the data loss. Then, we adopt PMU to design an efficient NSSA method. Compared with other deep neural networks (e.g., GRU), PMU is more suitable for managing the latent relations between short- and long-term dependencies. Meanwhile, our PMU-based assessment method is more efficient than the GRU-based assessment method.

3.2. AE-PMU-Based NSSA Method. In this part, we provide the detailed description of our proposed AE-PMU-based NSSA method. The algorithm pseudocode is shown in Algorithm 1.

As described in Algorithm 1, M represents the dimension of data dimensionality reduction, n represents the training period, and C represents the situation value after the evaluation of the test set. The main processes are as follows.

(1) Initialize the data dimension reduction dimension M and the number of training period n

(2) Use AE to extract the situation assessment elements from the initialized overall indicator dataset to achieve data dimensionality reduction

(3) Input the situation assessment elements of the training set into the PMU-based NSSA training model to implement the model training

(4) Input the situation assessment elements of the testing set into the PMU-based NSSA model to get the situation value C

4. Time Complexity Analysis and Experiment

We initially analyze the time complexity in this section. Then, we implement our AE-PMU-based NSSA method and provide the experimental results, including the precision, the efficiency, and the fit between the assessed indicator value and the real indicator value.

4.1. Time Complexity Analysis. Time complexity is a significant index to judge the merits of the algorithm. We will

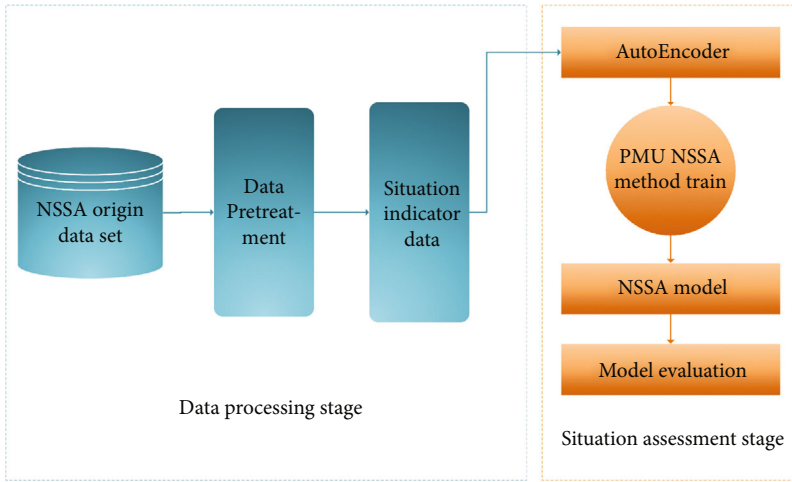

Figure 3: The system model.

Input: X-train dataset, Y-test dataset, Real-label, M-dimension, n-number of epoch.
Output: evaluation results.
1 Initialize the size of M and n.
2 $X \longleftarrow AutoEncoder(X - train, M)$
3 $Y \longleftarrow AutoEncoder(Y - test, M)$
4 **for** $i = 0$ **to** n **do**
5 $PMU \longleftarrow PMU_i(X)$
6 **end for**
7 $C \longleftarrow PMU(Y)$
8 **return** result $\longleftarrow C$

Algorithm 1: AE-PMU-based NSSA method.

analyze and compare the forward propagation time complexity of the GRU-based NSSA method and the PMU-based NSSA method.

We can assume that the dimension of the data input is m and the number of PMU hidden units is n. Firstly, according to formula (4), the number of operations for U_t can be represented as $T(n \times m + n^2 + n)$. Secondly, according to formula (5), the number of operations for calculating the current state candidate set is $T(n \times m + 2 \times n^2 + n)$. Thirdly, according to formula (6), the number of operations in the memory update phase is $T(n^2 + 2 \times n)$. Finally, the total number of operations of PMU is $T(2 \times n \times m + 4 \times n^2 + 4 \times n)$. Overall, the time complexity is $O(n^2)$.

Compared with PMU, GRU has one more gate structure, which has the same number of operations as the U_t gate of PMU. In addition, the memory update phase of the GRU is different from that of PMU. GRU uses Z_t to control whether the candidate set state is added to the memory update phase in this state. Therefore, the number of operations in the memory update phase of the GRU is $T(2 \times n^2 + n)$. The operation time of other parts of GRU is the same as that of PMU. Therefore, the total number of GRU operations is $T(3 \times n \times m + 6 \times n^2 + 4 \times n)$. In summary, the time complexity is $O(n^2)$.

As shown in Table 1, in general, although the time complexity of PMU is the same as that of the GRU, the total number of operations in PMU is much less than that in GRU. Therefore, the PMU-based NSSA is much more efficient.

4.2. Experimental Settings

4.2.1. Experimental Environment and Dataset. In our simulation experiment, the public dataset UNSW-NB15 is utilized as the experimental dataset [34, 35]. In UNSW-NB15, there are 9 different modern attacks. Meanwhile, every data record contains 43 elements and a corresponding label. UNSW-NB15 is divided into 4 different Comma-Separated Values (CSV) files, which contain a total of 2540044 data records. Moreover, there are 300000 abnormal traffic data records, as shown in Table 2.

As shown in Table 2, the dataset covers 9 different attack categories; the detailed categories are as follows:

(1) Analysis: an intrusion method that penetrates web applications through email, web scripts, ports, etc.

(2) Backdoors: an intrusion method that bypasses the system security mechanism through technical secrets to evaluate the computer or its data

(3) DoS: a method of deliberately attacking the implementation defects of the network protocol or directly

Table 1: PMU and GRU time complexity.

	Total number of operations	Time complexity
PMU	$T(2 \times n \times m + 4 \times n^2 + 4 \times n)$	$O(n^2)$
GRU	$T(3 \times n \times m + 6 \times n^2 + 4 \times n)$	$O(n^2)$

Table 2: UNWS-NB15 dataset data type.

Type	Quantity
Normal	2218761
Analysis	2677
Backdoors	2329
DoS	16353
Exploits	44525
Fuzzers	24246
Generic	215481
Reconnaissance	13987
Shellcode	1511
Worms	174

using brute force to exhaust the resources of the attacked object, so as to achieve an attack that makes the target network unable to use services or resources

(4) Exploits: a type of attack that exploits the attacker's knowledge of security vulnerabilities in the operating system or software

(5) Fuzzers: an attack type in which an attacker provides a large number of random numbers to the program or the network to make it down

(6) Generic: use hash functions for conflicts regardless of password configuration

(7) Reconnaissance: attacks used to collect computer information, also called probes

(8) Shellcode: the attacker uses shell commands and a small amount of code to control the attack mode of the attacked host

(9) Worms: worm attack, a virus attack that can replicate itself to the control host without any operation

For the convenience of experiment, we make statistics every ten minutes according to the time stamp in all the extracted dataset. A total of 144 sample data composed of the situation value is generated; among them, 100 are intercepted as the training set and 44 as the testing set. In addition, the network security situation value is 0-10. In the process of establishing the situation risk level, we will denote the situation value of 0-2 as safe, the situation value of 3-4 as low risk, and the situation value of 5-6 as medium risk. The situation value of 7-8 indicates a high risk, and the situation value of 9-10 indicates an emergency.

Table 3: Confusion matrix.

	Positive	Negative
True	TP	FP
False	FN	TN

4.2.2. Experimental Criteria. In this experiment, Accuracy, Precision, Recall, and $F1\_score$ are used to evaluate the effectiveness of our NSSA method and some of these concepts are defined as follows.

True positive (TP): TP means that the positive samples are evaluated as positive samples

False positive (FP): FP means that the negative samples are evaluated as positive samples

True negative (TN): TN means that the negative samples are evaluated as negative samples

False negative (FN): FN means that the positive samples are evaluated as negative samples

We use a confusion matrix to represent TP, FP, TN, and FN, as shown in Table 3.

Then, the Accuracy, Precision, Recall, and $F1\_score$ are defined as follows:

$$\begin{aligned} \text{Accuracy} &= \frac{\text{TN} + \text{TP}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \\ \text{Precision} &= \frac{\text{TP}}{\text{FP} + \text{TP}}, \\ \text{Recall} &= \frac{\text{TP}}{\text{FN} + \text{TP}}, \\ F1_{\text{score}} &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \end{aligned} \tag{9}$$

Accuracy represents the proportion of the number of correctly identified samples in the total sample. Precision represents the proportion of actual positive samples among the number of positive samples identified. Recall represents the percentage of positive examples in the sample that are predicted to be correct. However, it is unreasonable to evaluate the performance of the model only from Precision or Recall. To make the evaluation be more convincing, except for Precision and Recall, it is generally necessary to use $F1\_score$ as the model evaluation standard.

4.3. Evaluation Experimental Result. In this part, we evaluate the proposed AE-PMU-based assessment method, the PMU-based assessment method, the GRU-based assessment method, and BPNN-based assessment method from the points of effectiveness, fitting degree, and efficiency.

4.3.1. Effectiveness Evaluation. We compare the effectiveness of our AE-PMU-based assessment method, PMU-based assessment method, GRU-based assessment method, and BPNN-based assessment method, as shown in Figure 4.

Figure 4 measures the effectiveness of four different assessment methods from the Accuracy rate, Precision rate, Recall rate, and $F1\_score$. Among them, the AE-PMU-based assessment method has the best performance. This is

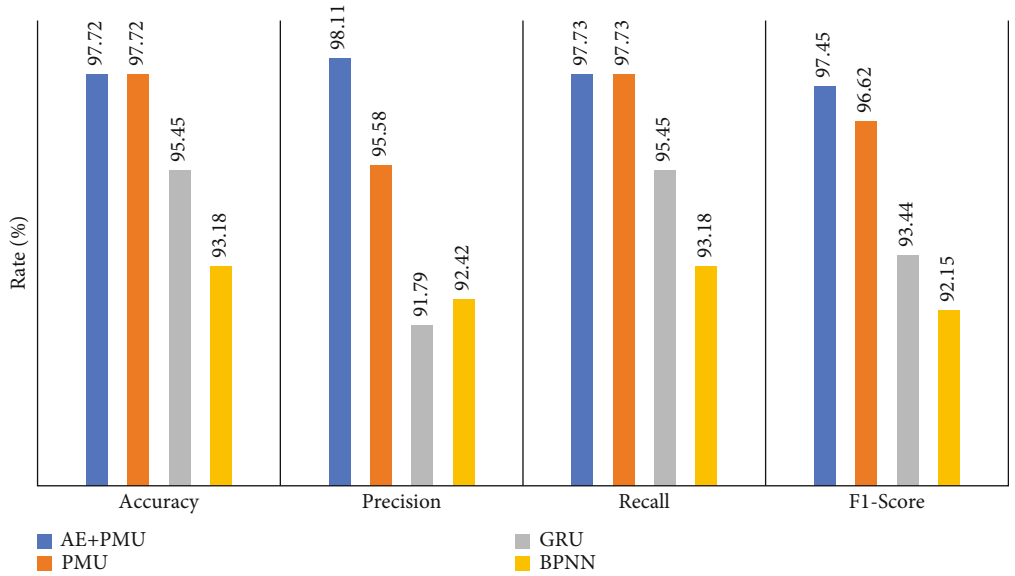

Figure 4: Comparison of evaluation effectiveness of different algorithms.

because BPNN-based assessment method does not consider the time sequence of the indicator data; thus, it could not evaluate the indicator data better. Although the GRU-based assessment method can consider the timing of the indicator data, compared with the PMU-based assessment method, GRU cannot effectively manage gates based on the latent relation between short- and long-term dependencies, so its effectiveness is inferior to that of the PMU-based assessment method. Because the data after AE dimensionality reduction removes the redundant part, the effectiveness of the AE-PMU-based assessment method is better than that of the PMU-based assessment method. This shows that the AE-based dimensionality reduction data fully retains the effectiveness of the indicator data, and the effectiveness of the PMU-based assessment method is better than that of the GRU-based assessment method.

4.3.2. Goodness of Fit. We utilize a polyline graph to intuitively show the comparison of the fit between the assessment value and the real value, as shown in Figure 5.

From Figure 5, we can see that when the sample numbers are 3, 13, 31, and 33, the network situation value fluctuates significantly, indicating that the network threats are relatively strong at these moments. In the third sample, a warning of "medium-risk" level appeared, indicating that the network is being threatened by a higher level attack, and security defense countermeasures should be taken. The "high-risk" level warnings appeared in the samples no. 13 and no. 31, indicating that the network suffers from extremely great security threats, and timely protection or rescue is required. According to the two fitting curves of the real value and the assessment value, it can be seen that the situation assessment result obtained by the proposed method basically fits the real security situation. Except for sample no. 31, which misjudged "high risk" as "safety," all other samples were correctly judged, which can more accurately fit the real security situation of the current network.

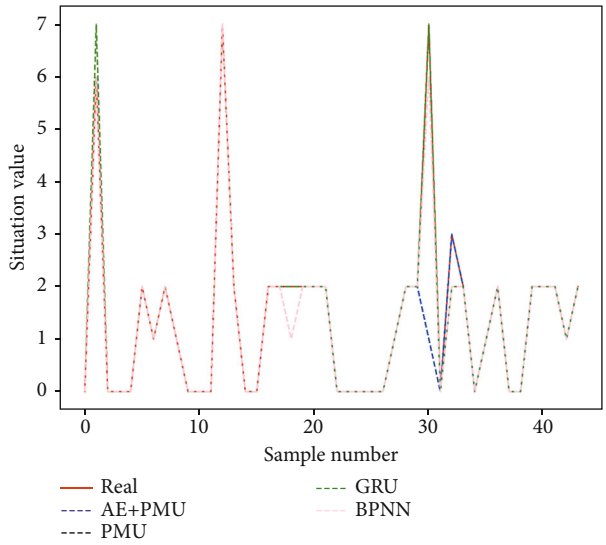

Figure 5: The fitted polyline of the assessment value and the real value.

Among other methods, the PMU-based assessment method makes a mistake once, the GRU-based assessment method makes a mistake twice, and the BPNN-based assessment method makes a mistake three times.

Our analysis of the reasons is consistent with the above "effectiveness evaluation" reasons. According to the above experimental results, it can be shown that the AE-PMU-based NSSA method can adapt to the NSSA under the modern network environment and can more accurately fit the real network security situation changes.

4.3.3. Performance Evaluation. Among the above four methods, although the network structure of BPNN is simple, it is rarely used in practical applications because it cannot fully characterize the data characteristics by using the time

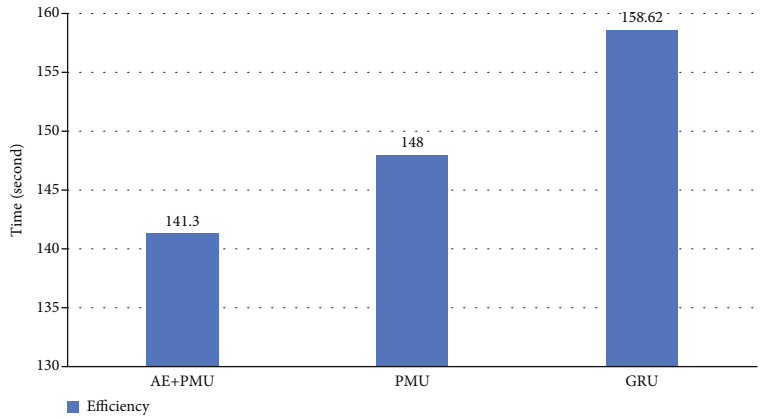

Figure 6: Efficiency comparison of different methods.

sequence of the data. Therefore, here we only compare the performance (assessment time) of NSSA methods based on PMU, GRU, and AE+PMU, as seen in Figure 6.

We can see from Figure 6 that the running time of the AE-PMU-based assessment method is the smallest, the running time of the PMU-based assessment method is the second, and the running time of the GRU-based assessment method is the longest. This is because the GRU has two gate structures. The reset gate helps the GRU decide which past information needs to be forgotten, and the update gate helps the GRU decide which past information needs to be passed to the future. However, PMU uses a gate to complete the calculation tasks of the GRU update gate and reset gate, thus reducing the amount of calculation. Meanwhile, AE reduces the dimension of the original indicator data. Hence, the computational efficiency of the AE-PMU-based assessment method is the best.

5. Conclusions

In large-scale network environment, the diversity of network threats and the high dimensionality of indicator data make the NSSA become more difficult. In this paper, we studied the NSSA in large-scale network environment and then proposed a novel NSSA method based on AE and PMU. Specifically, we first used AE for data dimensionality reduction to remove the redundant data. Then, we utilized PMU to achieve NSSA. By taking the advantage of PMU, the proposed method can effectively improve the performance of the model. Finally, we implemented the proposed method and provided the performance evaluation. The experimental results can show that compared with the existing methods, our method had significant advantages in efficiency, accuracy, and fit degree.

Data Availability

The dataset UNSW-NB15 can be got by sending e-mail to the corresponding author or 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 National Natural Science Foundation of China (No. 61962015), the Natural Science Foundation of Guangxi (No. 2020GXNSFBA297132), the Science and Technology Program of Guangxi (No. AD20297028), the Innovation Project of GUET Graduate Education (No. 2021YCXS061), and the Opening Project of Shanghai Key Laboratory of Integrated Administration Technologies for Information Security (No. AGK2020005).

References

[1] Y. B. Leau, S. Manickam, and Y. W. Chong, "Network security situation assessment: a review and discussion," in *Information Science and Applications*, K. Kim, Ed., vol. 339 of Lecture Notes in Electrical Engineering, pp. 407-–4414, Springer, Berlin, Germany, 2015.

[2] X. Tao, Y. Liu, F. Zhao, C. Yang, and Y. Wang, "Graph database-based network security situation awareness data storage method," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, 2018.

[3] P. Thagard, "Frames, knowledge, and inference," *Synthese*, vol. 61, no. 2, pp. 233–259, 1984.

[4] T. M. Mitchell, *Machine Learning*, McGraw Hill, Burr Ridge, IL, 1997.

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

[6] Y. F. Wang, J. Wang, Z. B. Xu, and H. Li, "Assessing cyber-threats situation for electric power information networks," in *2013 Ninth International Conference on Natural Computation (ICNC)*, pp. 1557–1562, Shenyang, China, 2013.

[7] H. Wang, Z. F. Chen, Z. Feng et al., "Research on network security situation assessment and quantification method based on analytic hierarchy process," *Wireless Personal Communications*, vol. 102, no. 2, pp. 1401–1420, 2018.

[8] F. W. Li, S. C. Yang, and J. Zhu, "Improved network security situation assessment method based on fuzzy hierarchy

method," *Journal of Computer Applications*, vol. 42, no. 9, pp. 2622–2626, 2014.

[9] R. Zhang, J. Cheng, X. Tang, Q. Liu, and X. He, "DDoS attack security situation assessment model using fusion feature based on fuzzy C-means clustering algorithm," in *International Conference on Cloud Computing and Security (ICCCS)*, X. Sun, Z. Pan, and E. Bertino, Eds., vol. 11064 of Lecture Notes in Computer Science, pp. 654–669, Springer, Cham, 2018.

[10] B. Yi, Y. P. Cao, and Y. Song, "Network security risk assessment model based on fuzzy theory," *Journal of Intelligent & Fuzzy Systems*, vol. 38, no. 4, pp. 3921–3928, 2020.

[11] Z. H. Liu, Z. Bin, Z. Ning, and L. Li, "Hierarchical network threat situation assessment method for DDoS based on DS evidence theory," in *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pp. 49–53, Beijing, China, 2017.

[12] D. Codetta-Raiteri and L. Portinale, "Decision networks for security risk assessment of critical infrastructures," *ACM Transactions on Internet Technology*, vol. 18, no. 3, pp. 1–22, 2018.

[13] J. Wang, M. Neil, and N. Fenton, "A Bayesian network approach for cybersecurity risk assessment implementing and extending the FAIR model," *Computers & Security*, vol. 89, article 101659, 2020.

[14] Z. Fan, Y. Xiao, A. Nayak, and C. Tan, "An improved network security situation assessment approach in software defined networks," *Peer-to-Peer Networking and Applications*, vol. 12, no. 2, pp. 295–309, 2019.

[15] Y. W. Liao, G. S. Zhao, J. Wang, and S. Li, "Network security situation assessment model based on extended hidden Markov," *Mathematical Problems in Engineering*, vol. 2020, Article ID 1428056, 13 pages, 2020.

[16] W. S. Noble, "What is a support vector machine?," *Nature Biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2006.

[17] Y. Liang, Z. P. Cai, J. G. Yu, Q. L. Han, and Y. S. Li, "Deep learning based inference of private information using embedded sensors in smart devices," *IEEE Network Magazine*, vol. 32, no. 4, pp. 8–14, 2018.

[18] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[19] X. Y. Chen, X. C. Yin, and A. Sun, "Network security situation assessment model based on GSA-SVM," in *Proceedings of 2018 International Conference on Computer, Communication and Network Technology (CCNT)*, pp. 414–420, Zhejiang, China, June 2018.

[20] J. Qiang, F. Wang, and X. L. Dang, "Network security based on DS evidence theory optimizing CS-BP neural network situation assessment," in *2018 5th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2018 4th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, pp. 153–159, Shanghai, China, June 2018.

[21] L. L. Shi and J. Chen, "Assessment model of command information system security situation based on twin support vector machines," in *2017 International Conference on Network and Information Systems for Computers (ICNISC)*, pp. 135–139, Shanghai, China, April 2017.

[22] Y. Y. Gao, Y. J. Shen, G. D. Zhang, and S. Zheng, "Information security risk assessment model based on optimized support vector machine with artificial fish swarm algorithm," in *2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pp. 599–602, Beijing, China, September, 2015.

[23] W. H. Han, Z. H. Tian, Z. Z. Huang, D. Q. Huang, and Y. Jia, "Quantitative assessment of wireless connected intelligent robot swarms network security situation," *IEEE Access*, vol. 7, no. 99, pp. 134293–134300, 2019.

[24] H. Y. Yang, R. Y. Zeng, G. Q. Xu, and L. Zhang, "A network security situation assessment method based on adversarial deep learning," *Applied Soft Computing*, vol. 102, no. 8, article 107096, 2021.

[25] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[26] G. E. Hinton, S. Osinder, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[27] M. Mohamed, "Parsimonious memory unit for recurrent neural networks with application to natural language processing," *Neurocomputing*, vol. 314, no. 7, pp. 48–64, 2018.

[28] L. Landberg, G. Giebel, H. A. Nielsen, T. Nielsen, and H. Madsen, "Short-term prediction? An overview," *Wind Energy*, vol. 6, no. 3, pp. 273–280, 2003.

[29] 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.

[30] C. Yang, X. Tao, F. Zhao, and Y. Wang, "A new outsourced data deletion scheme with public verifiability," in *Wireless Algorithms, Systems, and Applications. WASA 2019*, E. Biagioni, Y. Zheng, and S. Cheng, Eds., vol. 11604 of Lecture Notes in Computer Science, pp. 631–638, Springer, Cham, 2019.

[31] 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.

[32] X. Zheng and Z. P. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 968–979, 2020.

[33] K. Y. Li, G. C. Luo, Y. Ye, W. Li, S. Ji, and Z. Cai, "Adversarial privacy-preserving graph embedding against inference attack," *IEEE Internet of Things*, vol. 8, no. 8, pp. 6904–6915, 2021.

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

[35] 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, pp. 18–31, 2016.