Hierarchical Long Short-Term Memory Network for Cyberattack Detection

HAIXIA HOU1, YINGYING XU2, MENGHAN CHEN2, ZHI LIU2,3, (Member, IEEE), WEI GUO3, MINGCHENG GAO1, YANG XIN1, AND LIZHEN CUI3, (Member, IEEE)

1State Key Laboratory of Networking and Switching Technology, Information Security Center, Beijing University of Posts and Telecommunications, Beijing 100876, China
2School of Information Science and Engineering, Shandong University, Qingdao 266237, China
3Joint SDU-NTU Center for Artificial Intelligence Research (C-FAIR), Shandong University, Jinan 250101, China

Corresponding authors: Zhi Liu (liuzhi@sdu.edu.cn) and Yang Xin (yangxin@bupt.edu.cn)

This work was supported in part by the National Key Research and Development Program under Grant 2017YFB1400102, in part by the National Natural Science Foundation of China under Grant 61932005, Grant 61771071, and Grant 61972051, in part by the Key Research and Development Plan of Shandong Province under Grant 2017CXGC1503 and Grant 2018GSF118228, and in part by the Major Fundamental Research of Natural Science Foundation of Shandong Province under Grant ZR2019ZD05.

ABSTRACT With the continuous development of network technology, cyberattack detection mechanisms play a vital role in ensuring the security of computers and network systems. However, with the rapid growth of network traffic, traditional intrusion detection systems (IDSs) are far from being able to quickly and accurately identify complex and diverse network attacks, especially those related to low-frequency attacks. To enhance the overall security of the Internet, an IDS based on hierarchical long short-term memory (HLSTM) networks is proposed. With the introduction of HLSTM, the network can learn across multiple levels of temporal hierarchy over complex network traffic sequences. The system is evaluated on the well-known benchmark data set NSL-KDD for comparison with other existing methods. The experimental results demonstrate that compared with existing start-of-the-art methods, our system has better detection performance for different types of cyberattacks. In addition, the low-frequency network attack types have higher classification accuracy and a lower false detection rate.

INDEX TERMS HLSTM, cyberattack detection, NSL-KDD data set, intrusion detection system.

I. INTRODUCTION
Currently, networks are increasingly integrated with people’s daily lives. With the rapid development and widespread application of the Internet and the Internet of Things (IoT), network security has gradually attracted the attention of enterprises and countries. It is more necessary than ever to defend networks from cyberattacks. As a key part of network security defense, the intrusion detection system (IDS) refers to a network protection system built using certain security policies to detect intrusion behavior. Due to the large traffic volume and complex structure of network data, the processing capability of machine learning is limited. For this reason, traditional IDSs based on conventional machine learning methods generally have some shortcomings, such as a high false positive rate, poor generalization ability, and low real-time performance. Therefore, establishing an IDS that can effectively identify various complex and unknown intrusion attacks is an urgent issue.

In recent years, the excellent representation capability of deep learning has attracted much attention. Deep learning has achieved remarkable results in many areas, such as image recognition and natural language processing (NLP) [1]. To improve the intelligence and accuracy of network intrusion detection, many researchers have successfully applied convolutional neural network (CNN) to cyberattack detection, but they have not yet achieved the expected breakthrough [2], [3]. One of the main reasons for this failure is that network traffic is not in an image data format, so there are some problems in blindly applying CNN in IDS [4]. Considering that network traffic data are usually one-dimensional, an interesting attempt used a recurrent neural network (RNN) for processing. However, the existing few methods [5], [6] have shown that using a simple RNN model does not yield significant performance gains.
Considering the above factors, this paper proposes an intrusion detection method based on hierarchical long short-term memory (HLSTM) [7]. The main contributions of this paper are summarized as follows:

1) A HLSTM-based IDS is proposed, that can learn across multiple levels of temporal hierarchy over complex network traffic sequences. Experimental results show that this method has better detection performance for various network attacks, especially for low frequency network attack.

2) This paper also compares the detection ability of the algorithm to different types of network attacks in detail, and provides a comprehensive analysis of the problems in the detection of low-frequency network attack types.

The rest of this paper is organized as follows. Sect. II briefly introduces the research status and existing problems of network intrusion detection. Sect. III elaborates on the HLSTM-based intrusion detection model proposed in this paper, including the dataset used, data preprocessing method, HLSTM principle, and performance evaluation metrics. Sect. IV gives the specific experimental results and comparative analysis. The full-text summary and further work prospects are presented in Sect. V.

II. RELATED WORK

Traditional intrusion detection systems typically use machine learning algorithms. A similar research account is described in [2], including support vector machine (SVM), decision trees, k-nearest neighbor (KNN), random forests, naïve Bayes networks, and others. However, the actual network operating environment is complex and changeable, and various cyber-intrusion attacks are emerging one after another. Many unknown intrusion attacks are not included in the training data set; for example, 16.6% of the attack types in the test data set do not appear in the training data set for the NSL-KDD data set [8]. Therefore, traditional intrusion detection methods tend to have poor performance.

In recent years, deep learning has gradually become a popular research topic, and many works have applied deep learning methods to the field of cyber attack detection. Labonne et al. [9] proposed a cascade-structured meta-specialist approach based on multilayer perceptron (MLP) for classification in intrusion detection. Kasongo and Sun [10] used a feed-forward deep neural network with a filter for wireless intrusion detection. Yang et al. [11] provided an aggregation approach using the deep belief network (DBN) and modified density peak clustering algorithm (MDPCA). Jia et al. [12] designed an IDS based on a deep neural network (DNN).

With the breakthrough of CNN in the field of image and natural language processing, a series of CNN-based intrusion detection studies [13]–[16] were proposed. Despite these methods having achieved encouraging results in network intrusion detection, there are still some issues that need to be addressed. Considering that the network traffic generated in network communication is one-dimensional byte stream data rather than image data, blindly using CNN for intrusion detection inevitably affects the performance of the model.

Recently, an interesting attempt is to use RNN for intrusion detection, as in the literature [5], [6], [17]. However, all these works show that using only a simple RNN layer as the classifier does not yield significant performance gains in cyberattack detection. To further improve the performance of IDSs, an intrusion detection method based on HLSTM is proposed. Compared with the existing methods, our model has a higher detection accuracy with a lower false detection rate.

III. PROPOSED METHODOLOGIES

This section introduces the proposed HLSTM-based IDS in detail, and the system structure diagram is shown in Fig. 1. The data set used, the data preprocessing method, the proposed intrusion detection model based on HLSTM, and the performance evaluation metrics are sequentially described below.

A. NSL-KDD DATA SET

The NSL-KDD data set is the benchmark data set for intrusion detection in the network security field [8]. It is an improved version of the KDD CUP 99 data set [18], which effectively solves the problem of record redundancy, making the division of the training set and the test set more reasonable [19]. The NSL-KDD data set is widely used as an effective baseline dataset, that can help researchers compare different cyberattack detection methods.

The NSL-KDD data set consists of the training set KDDTrain+, and the test sets KDDTest+ and KDDTest-21, where the KDDTest-21 test set contains more unknown attack types and is, therefore, more difficult to classify than the KDDTest+ test set. The NSL-KDD data set contains five categories of network traffic data: normal, denial of service (DoS), probe, user to root (U2R), and remote to local (R2L). The number and percentage (PCT) of records for each category are shown in Table 1.

| TABLE 1. Distribution of different categories on the NSL-KDD data set [20]. |
|-----------------|-----------------|-----------------|
|                 | KDDTrain+       | KDDTest+        | KDDTest-21      |
| Count           | PCT             | Count           | PCT             | Count           | PCT             |
| Normal          | 67343           | 53.46%          | 9711            | 43.08%          | 2152            | 18.16%          |
| DoS             | 45927           | 36.46%          | 7458            | 33.09%          | 4342            | 36.64%          |
| Probe           | 11656           | 9.25%           | 2421            | 10.74%          | 2402            | 20.27%          |
| R2L             | 995             | 0.79%           | 2754            | 12.80%          | 2754            | 23.25%          |
| U2R             | 52              | 0.04%           | 200             | 0.30%           | 200             | 1.69%           |
| Total           | 125973          | 100%            | 22544           | 100%            | 11850           | 100%            |

Each record in the NSL-KDD data set contains 41 features and a corresponding classification label. These features are divided into four parts [21]: basic features, content features, time-based network traffic statistics features, and host-based network traffic statistics features, as shown in Table 2.
The data preprocessing includes data cleaning, numeralization, and normalization. These steps are briefly introduced below.

1) DATA CLEANING
Data cleaning is an important task in data mining and usually needs to be performed before the model training to ensure the quality of the data. For the NSL-KDD data set, although it has been improved, it can be found that the value of the 20th feature “num_outbound_cmds” is always 0, so it is a useless feature that needs to be removed. After data cleaning, each record in the data set contains 40 features.

2) NUMERALIZATION
After the data are cleaned, every record contains 40 features, including 7 symbolic features and 33 numerical features. To facilitate subsequent processing, the symbolic features should be converted to numerical features. The features “land”, “logged_in”, “is_hot_login”, and “is_guest_login” only contain ‘yes’ and ‘no’ characters, which can be directly encoded as 1 or 0. For the features “protocol_type”, “service”, and “flag”, which contain 3, 70, and 11 characters, respectively, one-hot encoding is used as the numerical method. Taking the feature “protocol_type” as an example, there are three types of protocols, “tcp”, “udp” and “icmp”, and their numeric values are “100”, “010”, and “001”. After numeralization, each record contains 121 values. In addition, the label for each record is also numerically processed using the one-hot encoding method.

3) NORMALIZATION
To reduce the impact of the numerical range of different features on model training, each feature value needs to be scaled to a reasonable range. In this paper, the normalization is used to scale the value to [0, 1], and the normalized value $x'$ is
\[
x' = \frac{(x - x_{\text{min}})}{(x_{\text{max}} - x_{\text{min}})}
\]
where $x$ is the initial value, $x_{\text{max}}$ and $x_{\text{min}}$ is the maximum and minimum value of the feature. In addition, logarithmic normalization would be a better choice for the features “duration”, “src_bytes”, and “dst_bytes”, which have a larger range of values.
C. METHODOLOGY

This subsection begins with a quick overview of the LSTM models. Given a sequence of inputs \( X = \{ x_1, x_2, \ldots, x_n \} \), each input \( x \) is paired with a sequence of outputs to predict \( Y = \{ y_1, y_2, \ldots, y_n \} \). An LSTM defines a distribution over outputs and sequentially predicts tokens using a softmax function:

\[
P(Y|X) = \prod_{t \in [1,n]} p(y_t|x_1, x_2, \ldots, x_t, y_1, y_2, \ldots, y_{t-1})
\]

\[
= \prod_{t \in [1,n]} \frac{\exp(f(h_{t-1}, e_y))}{\sum_{y'} \exp(f(h_{t-1}, e_y'))}
\]  

(2)

where \( f(h_{t-1}, e_y) \) is the activation function between \( e_{t-1} \) and \( e_y \), and \( h_{t-1} \) denotes the representation outputted from the LSTM at time \( t-1 \) [20].

Hierarchical LSTM can learn across multiple levels of temporal hierarchy over a complex sequence [22]. Generally, the first recurrent layer of the HLSTM encodes a sentence (e.g. word vectors) into a sentence vector:

\[
h^w_t = LSTM(h^w_{t-1}, e^w_t)
\]

(3)

where \( LSTM(\cdot) \) is defined as the LSTM operation for simplicity, \( h^w_t \) and \( e^w_t \) denote the hidden vectors from the LSTM model and embedding at the word level, respectively. The second recurrent layer then encodes a sequence of such vectors (encoded by the first layer) into a document vector:

\[
h^d_t = LSTM(h^d_{t-1}, e^d_t)
\]

(4)

where \( h^d_t \) and \( e^d_t \) denote the hidden vectors from the LSTM model and embedding at the sentence level, respectively [23]. The document vector is considered to preserve both the word-level and sentence-level structure of the context.

To obtain better attack detection performance, HLSTM is introduced to our proposed intrusion detection system, as shown in Fig. 2. For the preprocessed data, each record contains 121 values, so it can be converted into an \( 11 \times 11 \) pixel grayscale image. In the HLSTM-based intrusion detection model, the first LSTM layer first encodes each column of pixels of shape \((11, 1)\) to a column vector of shape \((128, 1)\). The second LSTM layer then encodes these 11 column vectors of shape \((11, 128)\) to an image vector representing the whole image. Finally, the full connection layer is added for prediction.

In addition, considering the severe imbalance in the sample size of different categories in the NSL-KDD dataset, it is necessary to rebalance the training data set to obtain better training results. There are two common methods for rebalancing classes: under sampling and over sampling, but they all have some shortcomings [24]. Therefore, the solution we adopted reduces the loss weight of the normal and the DoS categories and increasing the loss weight of the R2L and U2R categories. By weighting the loss function, the classifier does not tend to learn most of the representative classes.

D. EVALUATION METRICS

A commonly used performance evaluation metric is accuracy, which refers to the proportion of correct classification. Using true positive (TP), true negative (TN), false positive (FP), and false negative (FN) in the confusion matrix, the accuracy can be expressed as

\[
\text{Accuracy} = (TP + TN)/(TP + TN + FP + FN)
\]

(5)

In addition, considering the large difference in the number of records in different categories on the NSL-KDD data set, the following indicators are also used to evaluate the performance of the model [25].

\[
\text{Recall} = TP/(TP + FN)
\]

(6)

\[
\text{FPR} = FP/(FP + TN)
\]

(7)

\[
\text{Precision} = TP/(TP + FP)
\]

(8)

\[
F - \text{score} = 2TP/(2TP + FN + FP)
\]

(9)

where the recall, also known as the true positive rate (TPR) or detection rate (DR), is the proportion of correctly detected positives; the false positive rate (FPR), also known as false alarm rate (FAR), is the proportion of negatives that are incorrectly predicted as positive; the precision is the ratio of predicted positives to actual positives; the f-score is the harmonic average of precision and recall.

IV. EXPERIMENT RESULTS AND DISCUSSION

This chapter mainly validates the effectiveness of the proposed HLSTM-IDS on the benchmark data set NSL-KDD and compares it with existing state-of-the-art methods.

A. MODEL TRAINING

All the experiments in this paper are performed using the Keras [26] framework with the backend TensorFlow. The experimental device is a personal computer with a GeForce 1050Ti GPU for accelerating. All models are trained using the KDDTrain+ data set and tested using the KDDTest+ and KDDTest-21 data set. We use the Adam optimizer [27] with a fixed learning rate of 0.001 and train 200 times with a mini-batch size 1024. In addition, other configurations use default parameters.
Besides, considering the large difference in the number of samples of different attack types in the training set, we need to adjust the loss function in the model to avoid the classifier tends to learn the attack categories with a larger number. Specifically, the loss of different types of attack categories needs to be weighted, and the weight is the reciprocal of the ratio of the number of samples in the category to the total number of samples.

B. OVERALL CLASSIFICATION PERFORMANCE

To evaluate the performance of the proposed network model, we performed a multi-classification experiment on the NSL-KDD dataset. At the same time, to prove the superiority of the method, some of the latest deep learning based IDSs are given for comparison. The classification accuracy of each method is shown in Table 3. Our model achieved 83.85% and 69.73% accuracies in the two test sets, which is better than the best classifier in the existing work. The main reason is that HLSTM can learn across multiple levels of temporal hierarchy over the network traffic data. In addition, the specific experimental results are provided in Table 4 in the form of the confusion matrix.

### TABLE 3. The accuracy of our method and other existing methods.

| Method          | KDDTest+ data set | KDDTest-21 data set |
|-----------------|-------------------|---------------------|
| Char-CNN [15]   | 79.05%            | 60.86%              |
| CNN-IDS [24]    | 79.48%            | 60.71%              |
| STL+ SVM [28]   | 80.48%            |                     |
| RNN-IDS [6]     | 81.29%            | 64.67%              |
| Multi-CNN [29]  | 81.33%            | 64.81%              |
| DBN+ DPC [11]   | 82.08%            | 66.18%              |
| HLSTM-IDS       | 83.85%            | 69.73%              |

### TABLE 4. The confusion matrix of the HLSTM-IDS.

| Predicted class | Normal | DoS | Probe | R2L | U2R |
|-----------------|--------|-----|-------|-----|-----|
| Normal          | 9247   | 120 | 214   | 77  | 53  |
| DoS             | 730    | 5991| 35    | 681 | 21  |
| Probe           | 251    | 169 | 1863  | 116 | 22  |
| R2L             | 795    | 8   | 7156  | 188 |     |
| U2R             | 119    | 18  | 16    | 46  |     |

C. CLASSIFICATION PERFORMANCE FOR EACH ATTACK CATEGORIES

To assess the ability of the model to detect different categories of network attacks, we provide detailed performance metrics for each category of attack, along with other existing advanced deep learning-based methods for comparison. Note that most of the literature does not provide their performance on the KDDTest-21 data set, so we only compare performance on the KDDTest+ data set.

1) DoS

The DoS attack is also known as the flood attack. By exhausting the resources of the attacked object, the target computer or network cannot provide normal service or resource access, making it impossible for normal users to access or use the network service. The DoS attack does not include intrusion into the target server or target network device. The detection of such an attack is relatively straightforward and can be easily detected by carefully examining the connection statistics of the attacked object. Therefore, various intrusion detection methods usually achieve a quite good classification results, as shown in Fig. 3.

![FIGURE 3. Detection performance of different intrusion detection methods in the DoS category.](image)

2) PROBE

As an early stage of network intrusion, probing attacks are a growing cybersecurity issue. The original purpose of the probe is to understand the state of the network and to monitor or collect data about network activity. The probe attack attempts to access a target host through a known or potential vulnerability in the computer system. Similarly, the attack can be easily detected by carefully examining the connection statistics on the host. The results in Fig. 4 also show that each model has achieved better classification performance.

3) R2L

An R2L attack is an attack used to gain local access to a vulnerable computer. The premise is that an attacker can send packets to the victim computer over the network. This type of attack is a low-frequency attack, and the number of samples in the training set is also small, so it is difficult to detect. As seen from the results in Fig. 5, the accuracy of each method is greatly reduced. However, the method in this paper can still achieve a better precision with a lower false positive rate.

4) U2R

A user-to-root attack is a set of attacks that intrude through a user with no or low privileges, bypass authentication or
directly obtain root privileges through a vulnerability, and then log in for a series of illegal operations. This type of attack is also a low-frequency attack, and the number of samples in the training set is very small, so detection is very difficult. As seen from the results in Fig. 6, although the accuracy and FPR of each method are good, their precision is very low. Our method has a relatively balanced performance with better precision and a lower FPR.

D. DISCUSSION

From the above experimental results, we can see that the DoS and probe attacks can be easily detected by carefully checking network characteristics and connection statistics, but it is difficult to detect low frequency attacks such as U2R and R2L. The main reasons are as follows:

1) Compared with DoS and Probe attacks, the NSL-KDD data set contains fewer R2L and U2R attack samples, and insufficiently learning these two attacks makes the classifier less suitable for detecting such attacks.

2) Nearly half of the unknown attack types in the test data set KDDTest+ do not exist in the training data set KDDTrain+, which poses a great challenge for intrusion detection, especially R2L and U2R attacks.

3) The connection statistics of low-frequency attacks are very similar to normal connections. The classifier is more inclined to regard such attacks as normal types because of the uneven distribution of data.

4) There is a certain similarity between the behavior of U2R and R2L attacks. Therefore, it is also difficult to distinguish between U2R and R2L. In fact, U2R attacks are one of the variants of the R2L attack [21].

V. CONCLUSION

In this paper, an intrusion detection method based on HLSTM is proposed, which can learn across time levels on complex network traffic sequences. The detection quality of the system for the NSL-KDD data set is greatly improved. The accuracies of multi-classification on KDDTest+ and KDDTest-21 are 83.85% and 69.73%, respectively. It has higher detection precision and a lower false alarm rate than the most advanced methods available, especially in the detection of low-frequency attacks. Note that our method can also be applied to any other intrusion detection data set. For the next step, we need to add more U2R and R2L intrusion samples to the data set to address the increasingly severe low-frequency network intrusion attacks. In summary, the use of deep learning for analysis in the field of cybersecurity remains a challenging and open task.

REFERENCES

[1] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 4700–4708.

[2] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, and K. Han, “Enhanced network anomaly detection based on deep neural networks,” IEEE Access, vol. 6, pp. 48231–48246, 2018.
[3] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, “Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction,” Sensors, vol. 17, no. 4, p. 818, 2017.

[4] G. Feng, B. Li, M. Yang, and Z. Yan, “V-CNN: Data visualizing based convolutional neural network,” in Proc. IEEE Int. Conf. Signal Process., Commun. Comput. (ICSPCC), Sep. 2018, pp. 1–6.

[5] C. Li, J. Wang, and X. Ye, “Using a recurrent neural network and restricted Boltzmann machines for malicious traffic detection,” NeuroQuantology, vol. 16, no. 5, pp. 823–831, 2018.

[6] C. Yin, Y. Zhu, J. Fei, and X. He, “A deep learning approach for intrusion detection using recurrent neural networks,” IEEE Access, vol. 5, pp. 21954–21961, 2017.

[7] C.-H. Liu, D.-C. Juan, X.-A. Tseng, W. Wei, Y.-T. Chen, J.-Y. Pan, and S.-C. Chang, “Hierarchical LSTM: Modeling temporal dynamics and taxonomy in location-based mobile check-ins,” in Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining. Macau, China: Springer, 2019, pp. 217–228.

[8] S. K. Sahu, S. Sarangi, and S. K. Jena, “A detail analysis on intrusion detection datasets,” in Proc. IEEE Int. Advance Comput. Conf. (IACC), Feb. 2014, pp. 1348–1353.

[9] M. Labonne, A. Oliveireau, B. Polve, and D. Zeghlache, “A cascade-structured meta-specialists approach for neural network-based intrusion detection,” in Proc. 16th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC), Jan. 2019, pp. 1–6.

[10] S. M. Kasongo and Y. Sun, “A deep learning method with filter based feature engineering for wireless intrusion detection system,” IEEE Access, vol. 7, pp. 38597–38607, 2019.

[11] Y. Yang, K. Zheng, C. Wu, X. Niu, and Y. Yang, “Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks,” Appl. Sci., vol. 9, no. 2, p. 238, 2019.

[12] Y. Jia, M. Wang, and Y. Wang, “Network intrusion detection algorithm based on deep neural network,” IET Inf. Secur., vol. 13, no. 1, pp. 48–53, Jan. 2019.

[13] W. Wang, M. Zhao, and J. Wang, “Effective Android malware detection with a hybrid model based on deep autoencoder and convolutional neural network,” J. Ambient Intell. Hum. Comput., vol. 10, no. 8, pp. 3035–3043, Aug. 2019.

[14] H. Liu, B. Lang, M. Liu, and H. Yan, “CNN and RNN based payload classification methods for attack detection,” Knowl.-Based Syst., vol. 163, pp. 332–341, Jan. 2019.

[15] S. Z. Lin, Y. Shi, and Z. Xue, “Character-level intrusion detection based on convolutional neural networks,” in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Jul. 2018, pp. 1–8.

[16] Z. Li, Z. Qin, K. Huang, X. Yang, and S. Ye, “Intrusion detection using convolutional neural networks for representation learning,” in Proc. Int. Conf. Neural Inf. Process. Guangzhou, China: Springer, 2017, pp. 858–866.

[17] M. A. Khan, M. R. Karim, and Y. Kim, “A scalable and hybrid intrusion detection system based on the convolutional-LSTM network,” Symmetry, vol. 11, no. 4, p. 583, 2019.

[18] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl., Jul. 2009, pp. 1–6.

[19] K. Siddiqué, Z. Akhtar, F. Aslam Khan, and Y. Kim, “KDD cup 99 data sets: A perspective on the role of data sets in network intrusion detection research,” Computer, vol. 52, no. 2, pp. 41–51, Feb. 2019.

[20] L. Gao, X. Li, J. Song, and H. T. Shen, “Hierarchical LSTMs with adaptive attention for visual captioning,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 5, pp. 1112–1131, May 2019.

[21] P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, “A detailed investigation and analysis of using machine learning techniques for intrusion detection,” IEEE Commun. Surveys Tuts., vol. 21, no. 1, pp. 686–728, 1st Quart., 2019.

[22] Y. Du, W. Wang, and L. Wang, “Hierarchical recurrent neural network for skeleton based action recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2015, pp. 1110–1118.

[23] J. Li, M.-T. Luong, and D. Jurafsky, “A hierarchical neural autoencoder for paragraphs and documents,” 2015, arXiv:1506.01057. [Online]. Available: http://arxiv.org/abs/1506.01057

[24] K. Wu, Z. Chen, and W. Li, “A novel intrusion detection model for a massive network using convolutional neural networks,” IEEE Access, vol. 6, pp. 50850–50859, 2018.

[25] Z. Wang, “Deep learning-based intrusion detection with adversaries,” IEEE Access, vol. 6, pp. 38367–38384, 2018.

[26] F. Chollet. (2015). Keras. [Online]. Available: https://keras.io

[27] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Oct. 1998.

[28] M. Al-Qatif, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, “Deep learning approach combining sparse autoencoder with SVM for network intrusion detection,” IEEE Access, vol. 6, pp. 52843–52856, 2018.

[29] Y. Li, Y. Xu, Z. Liu, H. Hou, Y. Zheng, Y. Xin, Y. Zhao, and L. Cui, “Robust detection for network intrusion of industrial IoT based on multi-CNN fusion,” Measurement, vol. 154, Mar. 2020, Art. no. 107450.