A Network Intrusion Detection Method Based on Stacked Autoencoder and LSTM

[†]Yu Yan, [†]Lin Qi, [†]Jie Wang, [†]Yun Lin, [‡]*Lei Chen
[†]*College of Information and Communication Engineering, Harbin Engineering University*
Harbin, P.R.China
[‡]*Department of Information Technology, College of Engineering and Computing, Georgia Southern University*
Georgia, U.S.A.
Corresponding Author: LChen@georgiasouthern.edu

***Abstract*—Nowadays, network intrusions have brought greater impact in a large scale. Intrusion Detection Systems (IDS) have been a recent research hotspot for both the industry and the academic. However, due to the dynamic characteristics of network traffic, it is challenging to extract significant features and identify the traffic types. This paper focuses on applying deep learning methods to feature extraction. Specifically, an IDS model is proposed based on autoencoder and long short-term memory (LSTM) cell. The overall architecture of the intrusion detection model includes a feature extractor, a classifier, and an evaluation block. Different structures of the feature extraction model have been discussed and researched. Experiments conducted on the UNSW-NB15 dataset produce satisfactory result. A number of selected metrics such as accuracy and false alarm rate are adopted to evaluate the detection performance. Simulation results indicate that our model works better than competing machine learning methods and achieves accuracy of over 92%.**

***Index Terms*—Feature Reduction, Autoencoder, LSTM, Intrusion Detection**

I. Introduction

Today, the use of the Internet has penetrated into every aspect of people's daily life and work. While computer networks and systems enhance efficiency and convenience, they have also become the targets of malicious attacks. Hacker intrusions and privacy leakages frequently erupt, bringing various degrees of losses to governments, businesses and individuals [1]. Methods to detect network intrusions are receiving close attention as cybersecurity becomes an increasingly critical issue worldwide. With the continuous deterioration of the network environment, the corresponding network protection strategies, such as anti-malware systems, firewalls, and intrusion detection systems, have also been emerging. Among them, intrusion detection systems, as an active defense technology, has been paid much attention by network developers and administrators [2].

Generally, intrusion detection methods can be divided into two categories, namely host-based and network-based, according to the source of information [3]. The former is applied in a single host system and only serves for this host system on which IDS is located. It analyses the log information and evaluates the security performance of the host system. The latter usually monitors the traffic in the network which has a larger range of monitoring activities compared with the former. Network-based approach uses some certain traffic tools to capture traffic data and analyses it using detection algorithms.

From another perspective, IDSs can also be differentiated in accordance with the process of detection, 1)anomaly-based IDS, 2)misuse-based IDS, and 3)hybrid IDS [4]. Anomaly-based IDS judges the traffic only as either normal or attack. It can be regarded as a coarse-grained intrusion detection. Any activity that performs differently from the normal traffic will be considered as an attack. Usually, to train such an anomaly-based model, only normal data is used when an unsupervised method is applied, such as autoencoder and clustering. By discriminating the difference between input data and normal data, the anomaly-based intrusion detection methods can be able to detect the unknown attack. However, this method can easily make the intrusion detection system face a high false alarm rate. That is because some intrusions act similarly to the normal activities [5]. Misuse-based method is also called signature-based approach. It builds a database that stores different existing attacks. For the current traffic, the category can be obtained by matching it with the existing templates in the database. The advantage of misuse-based method is concrete detection of the already-known abnormal behaviors. However, However, it is not suitable for unknown attack detection. [6]. The hybrid method combines the misuse-based and anomaly-based methods. Usually, the database is adopted as prior information and the final result is a binary classification.

Common methods to build an intrusion detection model include machine learning, statistic knowledge, data mining, and expert rules [7]. Deep learning is a star member of the machine learning community and also has been applied in many fields such as medical diagnosis, automatic driving, etc. In this paper, the deep learning strategy is adopted to build the detection model. The stacked autoencoder serves as the first feature extractor, and the output becomes the input of the following feature extraction engine. The long short-term memory units, aiming to find the time relationship, are the second important component of the model. Finally, several fully-connected layers are joined after the LSTM layer. The aim of this block is to compress the dimension as well as obtain deep representation of traffic data. The extracted features will be used as input to the classifier.

978-1-7281-5089-5/20/$31.00 ©2020 IEEE

The remainder of this paper includes following parts. Section II provides related works about intrusion detection using machine learning methods. In Section III, we introduce the basic components and the details of the proposed method. Section IV describes the experimental results and analysis. Section V draws the conclusion and casts our future work.

II. Related Works

Network traffic data has high dimension, which makes machine learning methods easy to suffer from dimension curse. Many studies have focused on the problem of feature reduction, aiming to provide a simple and effective set of features. Generally, feature selection and feature extraction can achieve the purpose of feature reduction. In the feature selection process, certain original features will be selected according to rules. In contrast, the result of feature reduction is low-dimension data through feature mapping or other transformations.

In [8], Association Rule Mining (ARM) is adopted to obtain the best subset of UNSW-NB15. To begin, the most frequent value for each feature is determined and fed to the following ARM algorithm. Highly ranked features are obtained in this way. Use three algorithms to test the classification performance, namely The Expectation Maximization (EM) clustering, Logistic Regression and Nave Bayes, and achieve the accuracy of 77.2%, 83.0% and 79.5% respectively.

In [9], Agrawal constructs a hybrid detection model with the idea of binary tree. This method is especially useful to detect known attack. Each node in the binary tree is the best classifier for corresponding attack. In other words, the most critical step in building this model is to train the best classifier for each attack category. As for unknown attacks, SVM is adopted for anomaly detection. The accuracy of this method can be achieved of 88.55%.

In [10], Mustapha et al. present a two-stage classifier based on the RepTree algorithm. In the first phase, the traffic is judged as normal or abnormal. To improve the performance, traffic data first can be divided into three different parts based on different protocols: TCP, UDP, or others. If the traffic is detected to be abnormal, in the second stage, a Reduced Error Pruning Tree is applied for classification. The best accuracy of this method reaches 88.95%.

In [11], the authors propose a framework combining five different types of feature selection strategies. The Decision Tree classifier with Gain Ratio method achieves the highest 88% accuracy with 18 features.

The pioneer work based on principal component analysis, transforming high-dimensional features into low-dimensional data, is presented in [12], with a focus on the detection of Denial-of-Service attacks and Network Probe attacks. In [13], the authors use an approach combining an SOM network with k-means to detect anomaly over the KDDCUP99 dataset. SOM is capable of obtaining a preliminary clustering result, based on which k-means can be used for secondary clustering to further enhance the performance.

In order to better handle dimensional disasters, we apply autoencoder to the IDS model, and compare the model with the existing methods [10] [11] [17] [18] [19].

III. Proposed Work

In this section, the description of the proposed framework for intrusion detection is provided in detail. To begin, the basic components in proposed model are introduced followed by an illustration of the idea and the content of proposed framework. The evaluation metrics for the performance comparison are also presented in this section.

A. Basic Components

The autoencoder (AE) is usually used as a feature compression algorithm achieved by neural network [14], in which the compression and decompression are lossy and data-related. It is extremely popular in image reconstruction, clustering, machine translation, among others.

Generally speaking, the simplest structure of an AE only contains the input layer, the hidden layer, and the output layer. Three basic steps are required to build an autoencoder: building an encoder, building a decoder, and setting a loss function. Most encoding phases will result in feature reduction, mapping high-dimension data into low-dimension space. This function is realized by a bottleneck structure in the autoencoder. A concise representation of the encoding process is as follow:

$$H = f_\theta(X) = \sigma(W_{ij}X + b_{ij}) \tag{1}$$

where X is the input data and σ is the activation function. W_{ij} is the weight while b is the bias of the neuron unit. In this paper, $Tanh$ activation is adopted and computed by:

$$Tanh(t) = \frac{1 - e^{-2t}}{1 + e^{-2t}} \tag{2}$$

The decoding process is to reconstruct the data so that it can be fed back to the neural network with the loss function for network training. In general, decoding can be regarded as the inverse of the encoding process. A simplified expression of decoding is presented as follow:

$$Y = g_\theta(X) = \sigma(W_{jk}X + b_{jk}) \tag{3}$$

The loss function is an essential element of the gradient descent process. Due to the reconstruction operation in the decoding phase, the Mean Squared Error (MSE) is usually adopted in the autoencoder model. The loss will be propagated back to the hidden layer to update the weights and bias of the neuron units. So the entire loss function for the network can be presented as:

$$J(W, b) = \frac{1}{2N}\sum_{n=1}^{N} \|Y_n - X_n\|^2 \tag{4}$$

where N indicates the total number of samples.

To enhance the ability of AE, many autoencoders are stacked together to form a stacked autoencoder (SAE) as shown in Fig. 1. The advantage of the stacked self-encoder is that the feature extraction process gradually deepens as the number of layers increases. Usually, to train such model, layer-wise pre-training and fine-tuning are required.

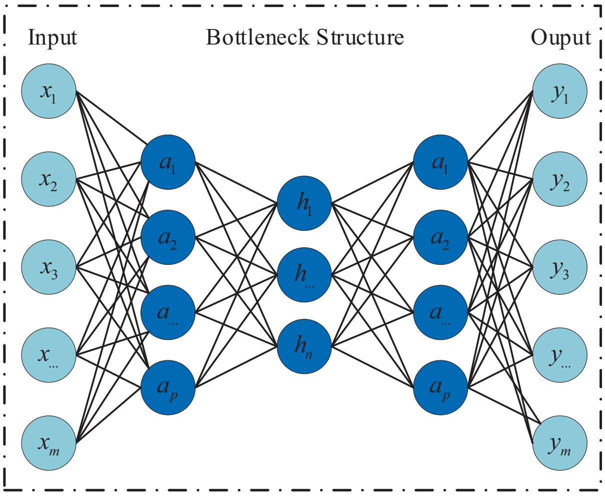


Fig. 1. Structure of Stacked Autoencoder.

The recurrent neural network (RNN) is created based on the idea that human cognition is based on past memory. The unique feature of RNN is the cyclic structure of neurons, which is useful for extracting time-related characteristics. Assuming a sequence $X = x_0, x_2, ..., x_{T-1}$, the hidden state h_t at time t can be represented as:

$$h_t = \sigma(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{5}$$

where σ is a non-linear function, W_{xh} and W_{hh} are weight matrices, and b is a bias vector.

Unfortunately, the continous multiplication of the weight matrix W will cause the gradient to disappear or explode. LSTM is an enhanced variant of RNN and aims to address gradient vanishing issues [15] which is shown in Fig. 2. The gate mechanism, which enables LSTM to perform better in long sequence problems, contains an input gate i_t, a forget gate f_t, and an output gate o_t.

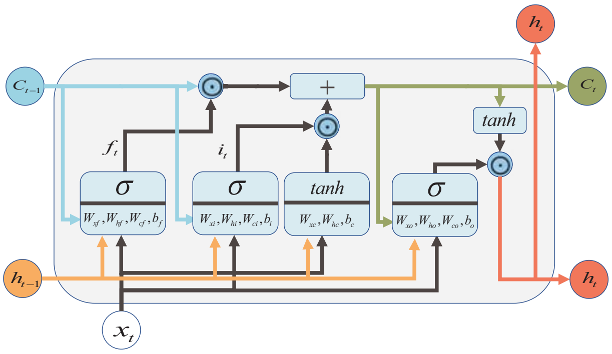


Fig. 2. The Structure of the Long Short-term Memory Cell.

The process to obtain an updated activation of a neural unit is as follow: The input gate combines the history information and the current input.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{6}$$

The forget gate then determines whether or not to pass the previous memory h_{t-1}.

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \tag{7}$$

c_t is an internal memory cell that combines the forget gate and the input gate. The output gate o_t shows the process to obtain the output of an LSTM cell from c_t.

$$\begin{cases} c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\ o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_{t-1} + b_o) \end{cases} \tag{8}$$

The activation of the hidden unit is updated as follow:

$$h_t = o_t \odot \tanh(c_t) \tag{9}$$

where $\odot$ denotes the Hadamard product.

B. Proposed Model

The overall framework of the proposed model based on the stacked autoencoder and LSTM is shown in Fig. 3. The model consists of a feature extractor, a classifier, and an evaluation block.

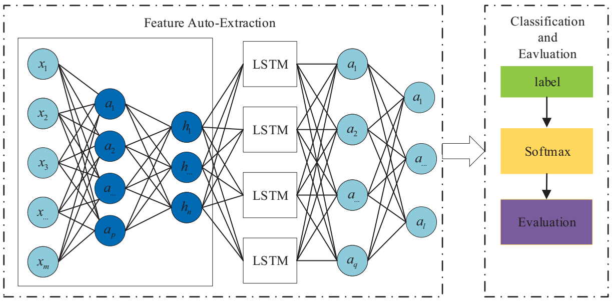


Fig. 3. Proposed Framework for Intrusion Detection using Deep Learning.

Before feature extraction, the input data should be pre-processed. In this paper, this step includes the feature transformation and normalization. Feature transformation is used to transform the nominal feature into numerical values for satisfying the requirement of input value type. Here, the one-hot encoding is adopted. Besides, the Min-Max skill is thereafter used to confine the encoded data range in [0,1].

$$x^* = \frac{x - min}{max - min} \tag{10}$$

The autoencoder can help reduce the feature dimension through the bottleneck structure. After that, the LSTM and the multi-layer neural network allow for deeper and more precise representations of features. Finally, the $softmax$ classifier classifies the extracted features. $softmax$ is widely used in dealing with the multiple categories tasks. The classified category is determined by the max probability of output cells.

$$S_i = \frac{e^{V_i}}{\sum_{i=1}^{n} e^{V_i}} \tag{11}$$

C. Evaluation Metrics

To better evaluate the model, different metrics are adopted based on the confusion matrix, which is illustrated in Table I.

TABLE I
CONFUSION MATRIX FOR BINARY CLASSIFICATION

Reality Predicted	Attack	Normal
Attack	TP	FP
Normal	FN	TN

Based on the above definition, the accuracy and far can be obtained as follows [16]:

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{12}$$

$$far = \frac{1}{2}(\frac{FP}{FP + TP} + \frac{FN}{FN + TN}) \tag{13}$$

It's obvious that the high-performance detection will be achieved, when the higher the accuracy value and the lower the FAR value.

IV. EXPERIMENTS

A. Dataset

Experiments are conducted using the UNSW-NB15 dataset which was developed by Moustafa et al. [16] with the details of this dataset introduced in Table II. One of the advantages of UNSW-NB15 is the rich data types, reflecting the characteristics of modern network traffic data. Moreover, the distribution is balanced both in the datasets. The official website provides a pair of training and testing datasets. In the training set,there are 82,332 records with a normal percentage of 45% and an abnormality of 55%. The testing set has 175,341 records, where 32% are normal and 68% are abnormal. It is good for model training. And then the trained model can better recognize attack label of each record.

B. Model Configuration

In this subsection, the details of the model configuration are described. One-hot encoding is the first step which causes the input dimension to change from 42 to 196. Generally, the number of nodes in the next layer will be relatively reduced to, for example 128 or 81. The model structure and related parameters are shown in Fig. 4.

To construct the bottleneck structure, the number of neurons in the Dense layer decreases with the deepening of layers. There is a Dropout layer connected to each Dense layer in the autoencoder part and the dropout rate is set to be 0.5. Additionally, the LSTM layer is joined to the former encoder

TABLE II
UNSW-NB15 DATASET

Number	Description	Number	Description
1	dur	23	dwin
2	proto	24	tcprtt
3	service	25	synack
4	state	26	ackdat
5	spkts	27	smean
6	dpkts	28	dmean
7	sbytes	29	trans_depth
8	dbytes	30	response_body_len
9	rate	31	ct_srv_src
10	sttl	32	ct_state_ttl
11	dttl	33	ct_dst_ltm
12	sload	34	ct_src_dport_ltm
13	dload	35	ct_dst_sport_ltm
14	sloss	36	ct_dst_src_ltm
15	dloss	37	is_ftp_login
16	sinpkt	38	ct_ftp_cmd
17	dinpkt	39	ct_flw_http_mthd
18	sjit	40	ct_src_ltm
19	djit	41	ct_srv_dst
20	swin	42	is_sm_ips_ports
21	stcpb	43	attack_cat
22	dtcpb	44	label

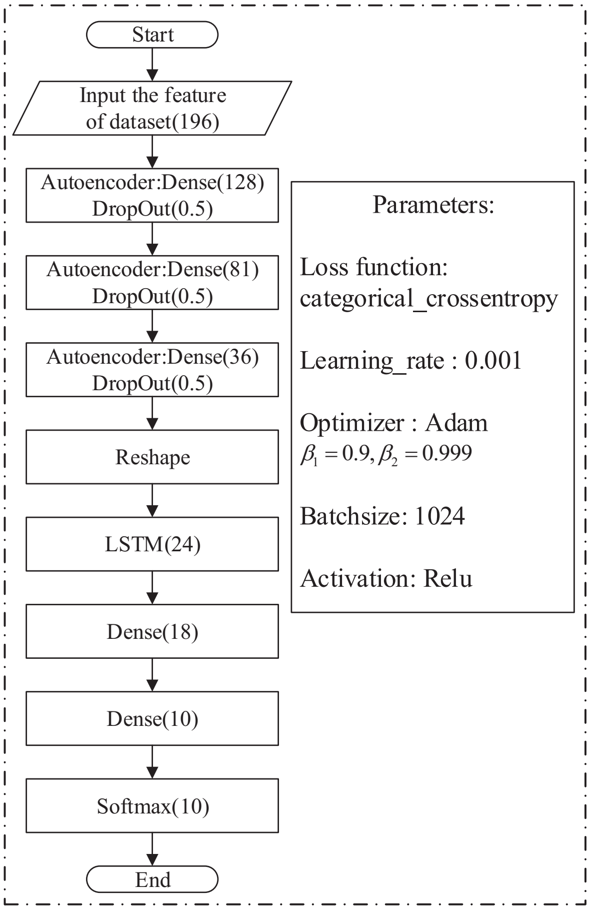

Start
Input the feature of dataset(196)
Autoencoder:Dense(128) DropOut(0.5)
Autoencoder:Dense(81) DropOut(0.5)
Autoencoder:Dense(36) DropOut(0.5)
Reshape
LSTM(24)
Dense(18)
Dense(10)
Softmax(10)
End
Parameters:
Loss function: categorical_crossentropy
Learning_rate : 0.001
Optimizer : Adam
$\beta_1 = 0.9, \beta_2 = 0.999$
Batchsize: 1024
Activation: Relu

Fig. 4. The flow chart of system model and related configuration.

model using the Reshape layer as interface. The LSTM layer has the same number of units with the last layer of the autoencoder. In our model, the number of LSTM layer is set to be 1. Furthermore, two fully-connected layers are joined to the LSTM layer and the output layer is used as the classifier with $softmax$ activation. Based on experience, the batchsize is 1024 and the initial value of learning_rate is 0.001. The Adam method is adopted as optimizer with $\beta_1 = 0.9, \beta_2 = 0.999$.

C. Result and Analysis

In order to determine an effective autoencoder architecture, different structures of autoencoder model are studied. For a fair comparison, we retain the same value for each parameter across different structures, with only two exceptions being the number of layers and neurons of the autoencoder. Different structures are listed in the first column in Table III.

TABLE III
THE ACCURACY AND FAR OF STACKED AUTOENCODER

SAE	Acc(train)	Acc(test)	FAR(train)	FAR(test)
[81, 36]	0.9493	0.8713	0.0608	0.1335
[100, 50]	0.9486	0.8874	0.0604	0.1177
[128,64,32]	0.9499	0.8892	0.0607	0.1176
[128,32,32]	0.9462	0.8871	0.0596	0.1229
[128,81,36]	0.9501	0.8929	0.0573	0.1081

In the training phase, the autoencoder is first trained with layer-wise strategy, and the final model is trained using fine-tuning method. The optimizer here is $Adam$. The results of simulation on different structures are shown in Table III. It is apparent that the [128,81,36] structure performs better than other structures and it is therefore regarded as the base for following experiments.

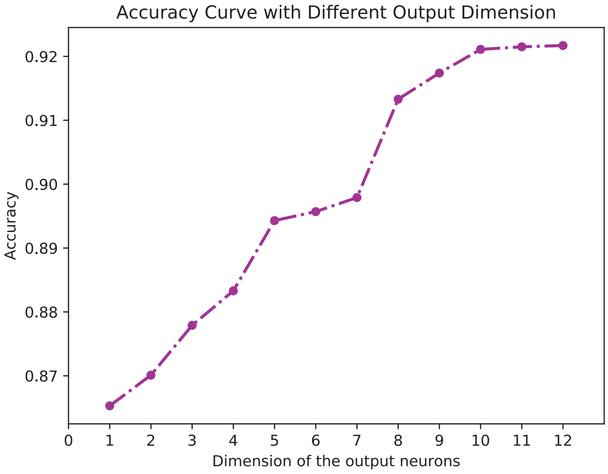


Fig. 5. The Accuracy Curve with Different Dimension of the Last Hidden Layer.

The structure of fully-connected layers also influences the performance of the model. We conduct research on the output dimension of the Dense layer. The number of neurons in the last hidden layer ranges from 1 to 12. For each situation, 100 Monte Carlo experiments were conducted and the average accuracy was obtained. The result is shown in Fig. 5. With the increase of unit dimension in the last hidden layer, the performance of the model improves. The more the number of units is, the better the non-linear expression will be. However, when the dimension is greater than 10, the trend of growth has slowed-down; in other words, the model is gradually converged. This is because 10 neurons are sufficient to characterize the data. Even if the number of neurons keeps increasing, the performance will not further improve, despite a risk of overfitting. Therefore 10 would be a proper dimension.

We compare our work with some of recent research as shown in Fig. 6. In order to make the comparison more comprehensive and meaningful, the existing feature reduction model and existing feature selection method are both considered. Among them, machine learning technology for feature reduction is adopted in [17], [18], and [19]. Feature selection cases are [10] and [11].

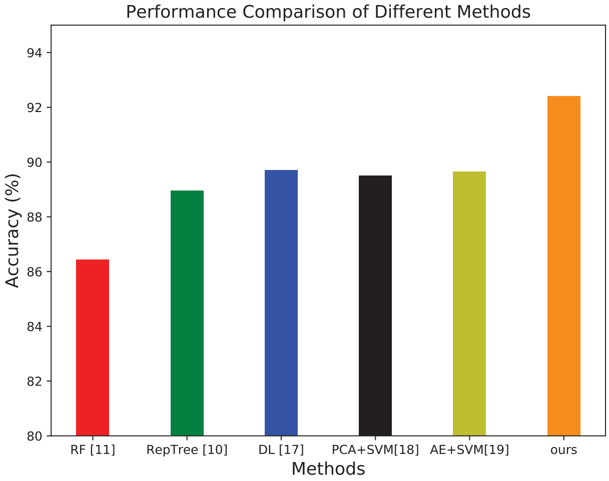
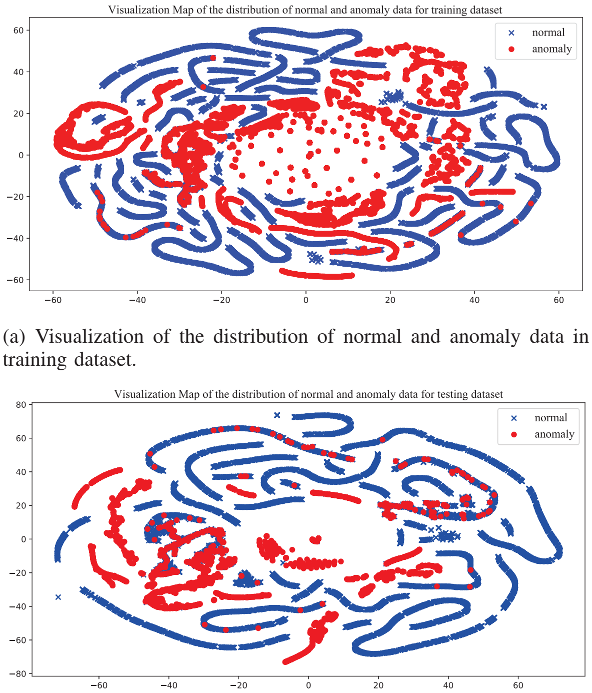


Fig. 6. Accuracy Comparison among Different Methods.

In Fig. 6, it is apparent that our model performs better than the combination of AE and SVM. This is because the LSTM cell excels in extracting more latent information. Additionally, the AE+SVM model performs better than the PCA+SVM model. This is because the autoencoder plays a role of enhanced PCA. PCA is a linear feature reduction method but the autoencoder not only works in a linear way but also in a non-linear way. Also shown in the figure, the RF and RepTree methods were not able to achieve satisfactory performance. It may be due to the reason that the selected feature subset is insufficient for the representation of the data. Overall, our proposed method would provide a satisfactory solution to intrusion detection problems.

Fig. 7 provides a visualization of the distribution of normal and anomaly by using $TSNE$ dimension reduction. The red dots represent the anomaly instances and the blue Xs represent the normal data. The upper part of the figure represents the situation in training dataset and the lower part is for the testing dataset. The number of samples accounts for 30%. Although there exist mixed points, the overall classification is accurate for most samples. The visualization is only for binary classification. The result for multi-classification is below satisfactory because attacks types mix with each other. In future work, the difference among attacks will be further investigated.



(a) Visualization of the distribution of normal and anomaly data in training dataset.

(b) Visualization of the distribution of normal and anomaly data in testing dataset.

Fig. 7. The Visualization Map for Training and Testing Dataset.

V. Conclusion

In this paper, we study the method of feature extraction for intrusion detection problem. A detection model based on autoencoder and LSTM is proposed. Experiments are conducted on the UNSW-NB15 dataset. The structure of the autoencoder is studied and a satisfactory autoencoder model is obtained. The proposed model achieves accuracy of over 92%. In addition, the output dimension is also considered as a factor to influence the model performance. The experiment results show that 10 dimension would be a proper selection. However,it is noteworthy that our proposed model also has the disadvantage of high false alarm rate. In future work, we will try to adopt other deep learning models and apply the trained IDS model into actual network environment.

Acknowledgment

This paper is funded by the International Exchange Program of Harbin Engineering University for Inovation-oriented Talents Cultivation. This work is also supported by the National Natural Science Foundation of China (61771154) and the Fundamental Research Funds for the Central Universities (HEUCFG201830).

Meantime, all the authors declare that there is no conflict of interests regarding the publication of this article. We greatly appreciate the helpful input and suggestions from the reviewers.

References

[1] Z. Inayat, A. Gani, N. B. Anuar, M. K. Khan, and S. Anwar, “Intrusion response systems: Foundations, design, and challenges,” *Journal of Network and Computer Applications*, vol. 62, pp. 53–74, 2016.

[2] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, “Network anomaly detection: Methods, systems and tools,” *IEEE Commun. Surv. Tuts.*, vol. 16, no. 1, pp. 303–336, 2014.

[3] F. Gumus, C. O. Sakar, Z. Erdem, and O. Kursun, “Online naive bayes classification for network intrusion detection,” in *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, pp. 670–674, IEEE, 2014.

[4] W. Bul’ajoul, A. James, and M. Pannu, “Improving network intrusion detection system performance through quality of service configuration and parallel technology,” *Journal of Computer and System Sciences*, vol. 81, no. 6, pp. 981–999, 2015.

[5] I. S. Thaseen and C. A. Kumar, “Intrusion detection model using fusion of chi-square feature selection and multi class svm,” *Journal of King Saud University-Computer and Information Sciences*, vol. 29, no. 4, pp. 462–472, 2017.

[6] A. L. Buczak and E. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2015.

[7] M. Ahmed, A. N. Mahmood, and J. Hu, “A survey of network anomaly detection techniques,” *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016.

[8] N. Moustafa and J. Slay, “A hybrid feature selection for network intrusion detection systems: Central points,” *arXiv preprint arXiv:1707.05505*, 2017.

[9] A. Agrawal, S. Mohammed, and J. Fiaidhi, “Developing data mining techniques for intruder detection in network traffic,” *International Journal of Security and Its Applications*, vol. 10, no. 8, pp. 335–342, 2016.

[10] M. Belouch, S. El Hadaj, and M. Idhammad, “A two-stage classifier approach using reptree algorithm for network intrusion detection,” *International Journal of Advanced Computer Science and Applications (ijacsa)*, vol. 8, no. 6, pp. 389–394, 2017.

[11] H. M. Anwer, M. Farouk, and A. Abdel-Hamid, “A framework for efficient network anomaly intrusion detection with features selection,” in *2018 9th International Conference on Information and Communication Systems (ICICS)*, pp. 157–162, IEEE, 2018.

[12] K. Labib and V. R. Vemuri, “An application of principal component analysis to the detection and visualization of computer network attacks,” in *Annales des télécommunications*, vol. 61, pp. 218–234, Springer, 2006.

[13] W. Huai-bin, Y. Hong-liang, X. Zhi-Jian, and Y. Zheng, “A clustering algorithm use som and k-means in intrusion detection,” in *2010 International Conference on E-Business and E-Government*, pp. 1281–1284, IEEE, 2010.

[14] S. T. Ikram and A. K. Cherukuri, “Improving accuracy of intrusion detection model using pca and optimized svm,” *Journal of computing and information technology*, vol. 24, no. 2, pp. 133–148, 2016.

[15] Q. Tian, J. Li, and H. Liu, “A method for guaranteeing wireless communication based on a combination of deep and shallow learning,” *IEEE Access*, vol. 7, pp. 38688–38695, 2019.

[16] F. A. Khan, A. Gumaei, A. Derhab, and A. Hussain, “A novel two-stage deep learning model for efficient network intrusion detection,” *IEEE Access*, vol. 7, pp. 30373–30385, 2019.

[17] B. Yan and G. Han, “Effective feature extraction via stacked sparse autoencoder to improve intrusion detection system,” *IEEE Access*, vol. 6, pp. 41238–41248, 2018.

[18] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural networks*, vol. 61, pp. 85–117, 2015.

[19] N. Moustafa and J. Slay, “The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set,” *Information Security Journal: A Global Perspective*, vol. 25, no. 1-3, pp. 18–31, 2016.