Intrusion detection using a combination of one-dimensional convolution and GRU

Xiaojuan Wang$^{1,a}$, Bo Xiao$^{2,b}$*

$^1$MOE Research Center for Software/Hardware Co-Design Engineering and Application Software Engineering Institute East China Normal University Shanghai, China
$^2$MOE Research Center for Software/Hardware Co-Design Engineering and Application Software Engineering Institute East China Normal University Shanghai, China

$^a$wxj9182@163.com

$^b$*Corresponding author: bxiao@sei.ecnu.edu.cn

Abstract—Intrusion detection plays an important role in ensuring network information security. Traditional machine learning technology does not work well enough with massive data and various intrusion classes, and detection accuracy turns unsatisfied with unknown intrusions. This paper proposes a new network intrusion detection model (Conv1d-GRU) that combines one-dimensional convolution and GRU for multi-class intrusion detection scenarios, and improves data imbalance by weighting the samples of each category. NSL-KDD as an improved version of the KDD CUP99 dataset is selected for our intrusion detection system. Experimental results on this dataset show that the proposed deep learning method is superior to present intrusion detection methods based on machine learning and deep learning, and has better feature representation learning and classification capabilities.

1. INTRODUCTION

With the development of science and technology, network security has become very important. Network security has many challenges, such as the rapid growth of network data and the diversity of data [1]. Intrusion detection is an important research achievement in the field of information security. It can identify intrusions, which can be ongoing or existing intrusions [2]. The goal of network-based intrusion detection is to classify the network traffic into five types: normal, DoS, probe, U2R and R2L [3]. Different from traditional machine learning, deep learning has a particularly attractive advantage, that is, it overcomes the limitations of shallow learning and can better capture feature information. This article studies how to use deep learning technology for intrusion detection to meet the challenges of network security.

In the early research, the most common way to detect intrusions was the analysis of user activities [6]. Then some machine learning methods emerged, including k nearest neighbor [7], support vector machine [5], naive Bayesian network [9] and random forest [8]. However, traditional machine learning usually emphasizes feature engineering and feature selection, and it is difficult to effectively process large-scale intrusion data [10]. In recent years, researchers have made many attempts to apply deep learning to the field of intrusion detection. Peng et al. [11] proposed a classification method to deal with data imbalance. But it requires high computational cost. Liu et al. [12] proposed an undersampling method to balance the dataset, but reduced the feature selection performance of CNN. Li et al. [4] proposed a deep learning intrusion detection method based on multiple neural network fusion methods. They divide the
characteristic data according to the correlation. According to [13], the author proposed a three-layer RNN architecture, but such a simplified neural network does not have the ability to model high-dimensional features. Le et al. [14] achieved good results using long short-term memory. In addition, Abien Fred M. Agarap [15] introduced a linear support vector machine to replace Softmax in the final output layer of the GRU model.

Based on the inspiration of previous work, this paper proposes a new method of intrusion detection based on one-dimensional convolution and GRU. In addition, the NSL-KDD dataset with separate training and test sets is used to evaluate the performance of the model in network intrusion detection in multi-class classification. In the experiment of this paper, we compare them with the latest machine learning and deep learning methods proposed by previous research.

Inspired by [2] and [4], this paper proposes a novel deep learning method based on the combination of one-dimensional convolution and GRU, and uses experiments to prove the advantages of the proposed method. The main contributions of this paper are summarized as follows.

- Due to our model uses a recurrent neural network and assumes that there is a dependency between data features, we use the method of data clustering [4] to weaken the feature correlation.
- By weighting the category that accounts for a relatively small proportion of the total sample when sampling, and weighting the samples of different categories when calculating the loss function, the data imbalance has been resolved to a certain extent.
- Using the combination of one-dimensional convolution and GRU on the benchmark NSL-KDD dataset provided by [5]. This provides a new idea for intrusion detection.

The rest of the paper is organized as follows. Section II introduces the methods used in the proposed Conv1d-GRU. Detailed experimental results and comparisons with previous studies using the NSL-KDD dataset are given in Section III. Finally, we draw conclusions and future research directions in Section IV.

2. MATERIALS AND METHODS

For the entire intrusion detection process. The first is to extract features from the processed one-dimensional data through one-dimensional convolution, and GRU is used as the next layer to better capture the correlation between features. Then, we use all the hidden states of the GRU to perform a convolution operation on the current result, and finally the fully connected layer predicts the final classification result. The process from model input to output is shown in Figure 1. The steps involved in this chapter include dataset description, data preprocessing, data clustering and model structure.

![Figure 1. Architecture of implemented Conv1d-GRU for intrusion detection.](image)

2.1. Dataset Description

NSL-KDD dataset is offline network data based on KDD CUP99 dataset [16]. Its size is relatively small, and the training set does not include redundant records, thus reducing the complexity [17]. The various advantages of NSL-KDD dataset over the original KDD dataset discussed by [17]. All models in this article are trained using the KDDTrain+ dataset, and are tested using the KDDTest+ and KDDTest-21 datasets respectively, which are all from NSL-KDD dataset. There are 41 features and 1 class label for every traffic record, and the features include basic features (No.1- No.10), content features (No.11 -
No.22), and traffic features (No.23 - No.41) as shown in Table 1 [2]. It is worth noting that KDDTest-21 contains records of attack types that are not in the KDDTrain+ and KDDTest+ data sets. Although this poses a greater challenge to the classification accuracy, it also enables it to provide a more realistic theoretical basis for intrusion detection.

2.2. Data Preprocessing
The NSL-KDD dataset contains 34 numeric features and 7 character features. Since the model requires the input to be numerical, the data needs to be numericalized. For all the discrete features, we choose one-hot coding method, which can make the distance calculation between features more reasonable. Because the function "num_outbound_cmds" contains only a fixed value in the dataset, so it is a useless function. Therefore, digitization can be used to convert features into 121 dimensional digital features.

In addition, data normalization can improve the convergence speed and training effect of the model, because it can eliminate the difference between data of different dimensions. Therefore, we adopt min-max normalization to make the data in the sample fall within the range of [0, 1]. The transformation function is as follows. Where \(x\) represents the initial data of the features, \(x_{\text{max}}\) represents the maximum data of the features, \(x_{\text{min}}\) represents the minimum data of the feature, and \(x'\) represents data after normalization of \(x\).

\[
x' = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}
\]  

(1)

2.3. Data Clustering
This article proposes a deep learning method combining one-dimensional convolution and GRU, but in the process of GRU processing data, it assumes that there is a dependency between features. In order to reduce the impact of feature correlation, we use a method of data clustering [3] to divide the feature data into 3 parts, as shown in Table 1, the first part is the basic feature, and the second part is the content feature. Statistical characteristics of network traffic is the third part. Data clustering can help learn high-level relationships between global features, while other classification algorithms ignore these relationships. By separating the less relevant features, the influence of additional additive correlation can be effectively reduced [3].

| No. | Features |
|-----|----------|
| No.1 - No.10 | duration, protocol_type, service, flag, src_bytes, dst_bytes, land, wrong_fragment, urgent, hot |
| | num_failed_logins, logged_in, num_compromised, root_shell, su_attempted, num_root, num_file_creations, num_shells, num_access_files, num_outbound_cmds, is_host_login, is_guest_login |
| No.11 - No.22 | count, srv_count, servs_rate, srv_serror_rate, srv_error_rate, srv_serror_rate, same_srv_rate, diff_srv_rate, srv_diff_host_rate, dst_host_count, dst_host_srv_count, dst_host_same_srv_port_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst_host_srv_serror_rate, dst_host_srv_error_rate, dst_host_srv_rate, dst_host_srv_diff_host_rate, dst_host_same_srv_rate, dst_host_diff_host_rate |
| No.23 - No.41 |
2.4. Model Structure

For different parts of the dataset, we use the same model structure. Take the basic feature part of the dataset as an example. First, after the input layer receives the data, the one-dimensional convolutional layer uses 16 convolution kernels to process the data. After passing through a recurrent neural network with 64 hidden neurons, all the generated hidden states perform a one-dimensional convolution again, changing the number of output channels to 1, and finally passes through a fully connected layer to produce the final classification result. It is worth noting that the recurrent neural network model uses GRU. GRU can solve the problems of gradient disappearance and gradient explosion in long-term memory and backpropagation, and can greatly improve training efficiency while maintaining the same effect.

The current input $x_t$ and a hidden state $h_{t-1}$ passed down from the previous node. This hidden state contains information about the previous node. Finally, GRU will get the output $y_t$ of the current hidden node and the hidden state $h_t$ passed to the next node. The calculation formula of GRU is as follows.

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$ (2)

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$ (3)

$$\tilde{h}_t = \tanh(W_\tilde{h} \cdot [r_t \ast h_{t-1}, x_t])$$ (4)

$$h_t = (1 - z_t) \ast h_{t-1} + z_t \ast \tilde{h}_t$$ (5)

$$y_t = \sigma(W_o \cdot h_t)$$ (6)

In the above formula, $\ast$ means two vectors are connected, and $\times$ means the product of two matrix. $W_r, W_z, W_\tilde{h}, W_o$ are the parameters to be learned by the model.

3. RESULTS AND DISCUSSION

Similar to most existing deep learning research, the model we propose is implemented using pytorch, all of our evaluations were performed using GPU-enabled pytorch running on a 64-bit Ubuntu 16.04 LTS PC with an Intel Core 3.60GHz processor, 64 GB RAM and an NVIDIA GTX 1080 Ti GPU. This article only conducts experiments on multi-classification scenarios for intrusion detection, and compares them with models proposed by other papers. The experimental results show that the prediction results of the KDDTest+ and KDDTest-21 test sets are better than other existing models.

First, we conduct an experiment on training a Conv1d-GRU model using all features as input. On this basis, we use three types of features separately as input to train another similar model. The output of the two models after the second one-dimensional convolutional layer are concated, and then the fully connected layer produces classification results. In fact, we also conduct experiments on training four similar models. After experiments, we find that when training two Conv1d-GRUs, using all feature data and basic features as input respectively, the model performance reaches the best. So we use this model to compare with other papers. Experiments actually show that when part of the data is used to assist all the data to train the Conv1d-GRU model, it can well weaken the assumption of GRU that there are dependencies between features, effectively extracts the high-level features of the data, and makes the final classification result have a higher accuracy rate and f1 score.

In order to solve the problem of data imbalance, we use the sampler module in pytorch to sample the data. The sampling method used is WeightedRandomSampler, which selects data according to the weight of each sample. At the same time, when we use the cross-entropy loss function, the reciprocal of the proportion of the sample size of each category to the total sample size is used as the weight of the loss function. In the experiment, the optimizer uses sgd and the learning rate is 0.1. The number of epochs is set to 170. Below we will introduce the performance of the Conv1d-GRU model on the two test sets. In addition, we found that the model performs best when no activation function is added between the layers.
of the model. The guess is that the GRU is sufficient to learn the nonlinear relationship between the data. Therefore, we use no activation function when training all models in the experiment.

**TABLE 2 THE ACCURACIES OF DIFFERENT MODELS IN MULTI-CLASS CLASSIFICATION.**

| Model      | Accuracy of KDDTest+ | Accuracy of KDDTest-21 |
|------------|----------------------|------------------------|
| RNN        | 81.29%               | 64.67%                 |
| SAE        | 79.10%               | -                      |
| STL-IDS    | 93.48%               | -                      |
| ANN        | 79.90%               | -                      |
| Multi-CNN  | 81.22%               | 64.81%                 |
| Our model  | 82.65%               | 65.94%                 |

In order to verify that our proposed model can indeed achieve better performance, that is, to achieve higher intrusion detection accuracy, we compare with some previous models with better performance, including RNN [2], SAE [18], STL-IDS [19], ANN [20], Multi-CNN [4], the comparison results are shown in Table 2. As you can see in the table, our model performance on the KDDTest+ test set is slightly better than other latest models, 0.32% higher than the previous optimal Multi-CNN Model, and for the KDDTest-21 test set, our model is 1.13% higher.

Figures 2 and 3 are the accuracy curves and f1 score curves of our model on the KDDTrain+, KDDTest+ and KDDTest-21 datasets. For the accuracy of each category, we also make a more detailed comparison. Compared with the RNN model, our model has a great improvement in the accuracy of R2L and U2R on the KDDTest+ dataset, and is similar to Multi-CNN in performance. The accuracy in the five categories is 94.83%, 82.14%, 82.90%, 37.07% and 22%. On the KDDTest-21 dataset, our model's detection accuracy of normal traffic is nearly 20% higher than

![Figure 2. The accuracy on the KDDTrain+, KDDTest+ and KDDTest-21 datasets in the multi-class classification.](image-url)
Figure 3. The f1 score on the KDDTrain+, KDDTest+ and KDDTest-21 datasets in the multi-class classification.

that of the Multi-CNN model, is 81.41%. The accuracy of the other four categories are 69.35%, 82.64%, 37.07% and 22.50%. In fact, the KDDTest-21 test set contains records of attack types that are not in the KDDTrain+ and KDDTest+ datasets, making it more difficult to classify. The NSL-KDD dataset itself has limitations in classification, so although the performance on the KDDTest+ test set is not so outstanding, the performance on the KDDTest-21 test set proves that our proposed model has good generalization ability on unknown attack types.

4. CONCLUSIONS
This paper proposes a network intrusion detection method that combines one-dimensional convolution and GRU. We use data clustering, and solve the problem of data imbalance through weighted sampling and weighting the loss function, which indeed improve the accuracy of intrusion detection. Experimental results show that compared with the latest machine learning and deep learning algorithms, our proposed Conv1d-GRU intrusion detection model performs better, with 81.65% and 65.94% accuracy on the multi-class classification of the KDDTest+ and KDDTest-21 test sets, respectively. The research in this paper proves that data clustering helps to weaken the hypothesis of RNN, that is, there is correlation between data features, and it proves that the combination of one-dimensional convolution and GRU has a good ability to extract advanced features. In short, Conv1d-GRU is suitable for intrusion detection. Our next job is to consider how to introduce the attention mechanism into the model structure, and continue to consider how to improve the detection accuracy, so as to better detect network attacks.

REFERENCES
[1] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi. (2018) A deep learning approach to network intrusion detection. IEEE Trans. Emerg. Topics Comput. Intell., vol. 2, no. 1, pp. 41–50.
[2] C. Yin, Y. Zhu, J. Fei, X. He. (2017) A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access 5: 21954–21961.
[3] Wu KH, Chen ZG, Li W. (2018) A novel intrusion detection model for a massive network using convolutional neural networks. IEEE Access, 6: 50850–50859.
[4] Li Y, Xu Y, Liu Z, et al. (2020) Robust detection for network intrusion of industrial IoT based on multi-CNN fusion. Measurement. 154: 107450.
[5] M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani. (2009) A Detailed Analysis of the KDD CUP 99 Data Set, in: 2009 IEEE Symposium on Computational Intelligence for Security and Defence Applications. IEEE, pp. 1-6.
[6] Anup K Ghosh, Aaron Schwartzbard, and Michael Schatz. (1999) Learning Program Behavior Profiles for Intrusion Detection. In Workshop on Intrusion Detection and Network Monitoring, Vol. 51462: 1–13.

[7] W. Li, P. Yi, Y. Wu, L. Pan, and J. Li. (2014) A new intrusion detection system based on KNN classification algorithm in wireless sensor network. J. Elect. Comput. Eng., vol. 2014, no. 5, pp. 1–8.

[8] G. Kim, S. Lee, and S. Kim. (2014) A novel hybrid intrusion detection method integrating anomaly detection with misuse detection, Expert Syst. Appl., vol. 41, no. 4, pp. 1690–1700.

[9] L. Xiao, Y. Chen, and C. K. Chang. (2014) Bayesian model averaging of Bayesian network classifiers for intrusion detection. In Proc. Comput. Softw. Appl. Conf. Workshops, pp. 128–133.

[10] J. Pang, D. Liu, Y. Peng, X. Peng. (2017) Anomaly detection based on uncertainty fusion for univariate monitoring series. Measurement 95: 280-292.

[11] L. Peng, H. Zhang, Y. Chen, B. Yang. (2016) Imbalanced traffic identification using an imbalanced data gravitation-based classification model, Comput. Commun., vol. 102, pp. 177–189.

[12] Y. Liu, S. Liu, and X. Zhao. (2017) Intrusion detection algorithm based on convolutional neural network. Beijing Ligong Daxue Xuebao/Trans. Beijing Inst. Technol., vol. 37, no. 4, pp. 1271–1275.

[13] M. Sheikhan, Z. Jadidi, and A. Farrokhi. (2012) Intrusion detection using reduced-size RNN based on feature grouping. Neural Comput. Appl., vol. 21, no. 6, pp. 1185–1190.

[14] T. T. H. Le, J. Kim, and H. Kim. (2017) An effective intrusion detection classifier using long short-term memory with gradient descent optimization. In Proc. Int. Conf. Platform Technol. Service, pp. 1–6.

[15] Abien Fred M. Agarap. (2018) A Neural Network Architecture Combining Gated Recurrent Unit (GRU) and Support Vector Machine (SVM) for Intrusion Detection in Network Traffic Data. In ICMLC 2018: 2018 10th International Conference on Machine Learning and Computing, Macau, China.

[16] P. Bhoria, K. Kanwal Garg. (2013) Determining feature set of DOS attacks. International Journal of Advanced Research in Computer Science and Software Engineering, vol.3 issue 5, May 2013, pp. 875-878.

[17] NSL-KDD Data Set. (2015) Available: http://nsl.cs.unb.ca/NSL-KDD/.

[18] A. Javaid, Q. Niyaz, W. Sun, M. Alam, A deep learning approach for network intrusion detection system. (2016) In: Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS), ICST (Institute for Computer Sciences, Social Informatics and Telecommunications Engineering), pp. 21-26.

[19] M. Al-Qatif, Y. Lasheng, M. Al-Habib, K. Al-Sabahi. (2018) Deep learning approach combining sparse autoencoder with SVM for network intrusion detection, IEEE Access 6: 52843-52856.

[20] B. Ingre, A. Yadav, Performance analysis of nsl-kdd dataset using ann. (2015) In: Signal Processing And Communication Engineering, in: Systems (SPACES), 2015 International Conference on, IEEE, pp. 92-96.