2021 IEEE/ACIS 22nd International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD) | 978-1-6654-0403-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/SNPD51163.2021.9704954

An Effective Preprocess for Deep Learning Based Intrusion Detection

Chia-Ju Lin
Department of Computer Science and Information Engineering
National Chinyi University of Technology
Taiping, Taichung, Taiwan
wasjulie0905@gmail.com

*Ruey-Maw Chen
Department of Computer Science and Information Engineering
National Chinyi University of Technology
Taiping, Taichung, Taiwan
raymond@ncut.edu.tw

***Abstract*—The data preprocess directly affects the classification results in various applications. In the field of intrusion detection, less research raised the problems or solutions of unequal metrics in data attributes. This study proposes an effective data preprocessing method for network packets with unequal metrics in packet attributes. A standard deviation standardization was first applied to standardize each attribute of KDDCUP'99 dataset, followed by quantizing it to the range of 0 to 255 interval for afterward use of the image. Meanwhile, the Zigzag arrangement coding and IDCT (Inverse Discrete Cosine Transform) were then used to convert the quantized data into images. Experimental results demonstrate that a more than 94% recall rate of the overall intrusion detection classifier can be yielded by the proposed preprocess method even without a complicated network model. Meanwhile, intrusion detection performance can be guaranteed by using small-size images of packet attributes.**

Keywords—Intrusion Detection, KDDCUP'99, Data Preprocess, Feature Extraction, Standard Deviation Standardization

I. INTRODUCTION

With the prevalence of the Internet in recent years, various kinds of cyber-attacks emerge endlessly. Intrusion detection technologies have been wildly used for solving network security issues.

Intrusion detection technologies can be categorized into two types based on the analyzed technologies, which are anomaly intrusion detection and misuse intrusion detection. Anomaly intrusion detection can detect attacks which are not in the dataset but may accompany a high false positives rate[1]. Misuse intrusion detection is based on the known intrusion signature, the packets are matched with the intrusion signature rules, to judge whether it is an attack.

An intrusion detection system defines as "detecting abnormal behaviors or damage attempts to intrude the system in the database by using intrusion detection technologies to check or compare the behavior, security logs, inspect data or any other information available on the Internet.[2]".

Nowadays, many research methods for intrusion detection are based on machine learning. The classification performance of the intrusion detection system is highly related to the results of the feature extraction and selection [3]. However, still rarely proposed methods import all features for training phases.

Besides, deep learning has been widely used in various research in recent years. Many contributions devout in image recognition and speech recognition, such as Convolutional Neural Network (CNN) has good performance in image classification tasks [4]. Deep learning methods result in good performances in prediction and classification since automatical learning non-linear correlation features from the original data to extract features, no feature selection and extraction process required [5].

In the dataset for intrusion detection, each packet data contains many features, which can be regarded as high-dimensional data. High-dimensional data needs to be preprocessed appropriately to avoid the data being ineffectively used, such as feature selection, data dimension reduction, data standardization or data regularization, etc. [6]. Restated, suitable standardization processes is a way that can be used to effectively improve the accuracy of classification.

Different standardization methods are analyzed and compared in many studies, most of the studies import all features for standardization. However, not all the features have the same metric and ranges. Therefore, standardization using all features may cause data distortion. Thus, standard deviation standardization implemented feature by feature is suggested, then quantize the standardized feature to the range of 0 to 255. Meanwhile, the Zigzag arrangement coding is applied for the quantized data. This indicates that the packet attributes are regarded as the spatial frequency distribution of the associated image. Then the inverse discrete cosine transform (IDCT) is then utilized to transform the spatial frequency spectrum to the spatial domain image. The spatial domain image is the input of the following convolutional neural network (CNN) model. Restated, to take advantage of the automatic learning outstanding in deep learning, this work treats each packet as a gray-scaled image and used it as the input of the CNN model .

II. Proposed Methodology

A. KDDCUP'99 Dataset Description

KDDCUP'99 dataset [7] is modified from DARPA98 dataset. DARPA collected network connection data for nine weeks in a military network environment, including two weeks for testing data, part of the duplicate data was removed by Wenke Lee et al. [8] named DARPA98 dataset.

The KDDCUP'99 dataset includes 4,898,431 packet records. Each packet contains 41 features and one target class feature. It has been divided into four types of attack as Probe, DoS, U2R, R2L, and Normal packets [9]. These four categories of attack packets are categorized as abnormal packets.

B. Preprocess of KDDCUP'99 Dataset

The main purpose of preprocessing is to convert the original data into an appropriate format for subsequent analysis and use [10]. KDDCUP'99 data contains both continuous and discrete attribute characteristics, among the continuous attributes every measurement metric is different. If the import data is without standardization or arrangement before CNN model training, it may undergo a poor feature extracting process.

The flow of the suggested preprocess is shown in Fig. 1. The preprocessing is divided into four steps, which are data numerical, data standardization, data quantization, then Zigzag coding and image conversion.

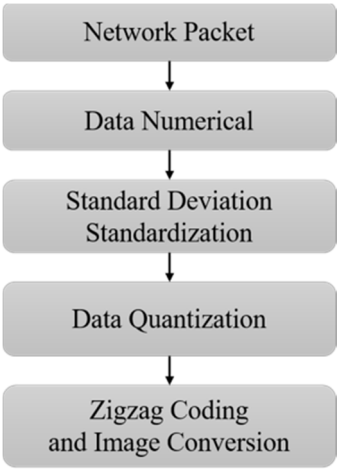

Fig. 1. The flow of preprocessing

The data numerical applies the one-hot encoding method to transform the literal characters into numerical type [11]. In the data standardization process, a standard deviation standardization on each feature attribute is conducted, as shown in Eqs. (1) and (2). The $\overline{X_k}$ represents the average value of the k^{th} attribute, S_k represents the mean square error of the k^{th} attribute, and X_{ik} indicates the k^{th} attribute in the i^{th} packet data record. Then standardize each packet data record, as listed in Eq. (3).

$$\overline{X_k} = \frac{1}{n}\sum_{i=1}^{n} X_{ik} \quad (1)$$

$$S_k = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(X_{ik} - \overline{X_k}\right)^2} \quad (2)$$

$$Z_{ik} = \frac{X_{ik} - \overline{X_k}}{S_k} \quad (3)$$

The data quantization process is shown in Eq. (4). X and X^* are the values to be mapped and the mapped attribute values respectively, min and max are the minima and maximum values in each packet respectively. Each network packet will be quantized in the range of the value 0 to 255.

$$X^* = \left(\frac{x - min}{max - min}\right) \times 255 \quad (4)$$

After the data is standardized and quantized, the data need to be padded and coded then converted to images. Since the image size for the using CNN model are 16×16 and 32×32 pixels respectively, which exceeds the number of the 41 attribute values of the packet, hence the data padding process is applied to fulfill the size of the selected image. Restated, in addition to the 41 characteristic attributes of each packet, the rest of the data elements are padded with zero indicating no data. Here in, the Zigzag arrangement method is used as the data encoding scheme. Figure 2 gives an example of the coding data with the form of Zigzag, the direction of the arrow shows the direction of data filling. Therefore, the Zigzag arrangement operation concentrates data information in the upper left corner that is the lower spatial frequency part of the frequency domain. The Zigzag coding example is shown in Fig. 3.

The Zigzag coding process is considered as a frequency domain map of an image after discrete cosine transform (DCT). Hence, converting the frequency domain map back to a gray-scaled image utilizing inverse discrete cosine transform(IDCT) is required, the image is as shown in Fig. 4, which is the input for Convolutional Neural Network (CNN) model.

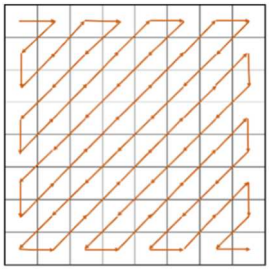

Fig. 2. Zigzag Schematic diagram of data arrangement

Fig. 3. An example of the Zigzag arrangement

Fig. 4. An example of packet attributes image

Identify applicable funding agency here. If none, delete this text box.

III. Model Overview

The used convolutional neural network CNN model for feature extraction and model training includes two convolutional layers and a three-layer fully connected neural network. The results of each convolutional layer pass through a max-pooling layer. Finally, the output of the fully connected layer, which contains a softmax activation function to classify intrusions for detection. The output of the used model classifies the types of packet including normal and abnormal packets. The architecture of the studied model is shown in Fig. 5.

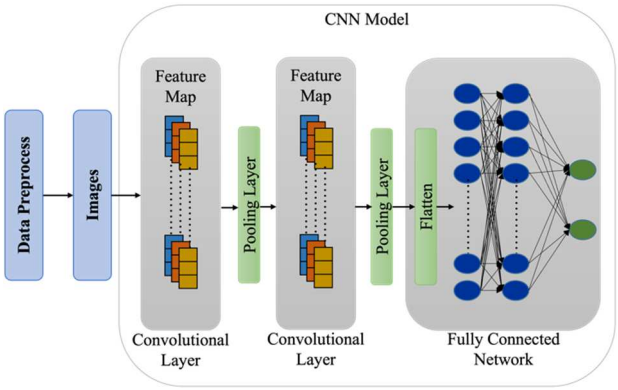

Fig. 5. The architecture of the studied model

IV. Evaluation and Experimental Results

A. Dataset

The kddcup.data_10_percent.gz file was used as the experimental network packet dataset. This file contains 10% of the data volume of KDDCup'99 Data, having 494,021 connection records in total. TABLE I. displays the distribution of training and testing datasets in this study.

TABLE I. Distribution of Training and Testing Sets

Label	Name	Training set (70%)	Testing set (30%)
Normal	Normal	68,094	29,184
Abnormal	DOS	274,021	117,437
	Probing	2,875	1,232
	R2L	788	338
	U2R	36	16

B. Evaluation Metrics

This study aims to predict whether packets are associated with abnormal activities. As shown in TABLE II, the true positive is defined as when a network packet is predicted by the classifier as abnormal when it's an abnormal network packet.

TABLE II. Evaluation Metrics of Intrusion Detection

		Predict	
		Abnormal	Normal
Actual	Abnormal	**T**rue **P**ositive	**F**alse **N**egative
	Normal	**F**alse **P**ositive	**T**rue **N**egative

In this investigation, precision rate, recall rate, and accuracy are used to evaluate the performance of the proposed preprocessing scheme. Precision rate is used to show the detection ability of the abnormal network packets. Recall rate represents the ability to identify abnormal packets of the model. Accuracy is used to evaluate the performance of the whole classifier. Their definitions are presented below. Among these three performance metrics, the recall rate is crucial to the network security issue. A high recall rate indicates abnormal packets can be blocked to prevent the system from being intruding.

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

$$Recall = \frac{TP}{TP+FN} \quad (6)$$

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (7)$$

C. Experimental Results

In the training phase, the number of epochs is set to 2000, the value of the learning rate is set to 0.0006, the batch size in training was set to 256, and the image input size of the neural network is set to 16×16 and 32×32 for comparison. Accordingly, the fully connected layers give 256 and 1024 input neurons followed by 128 and 32 hidden neurons. Figs. 6 and 7 demonstrate the loss value and accuracy rate of 16×16 image size during the training phase respectively, Figs. 8 and 9 indicate the training course of 32×32 images.

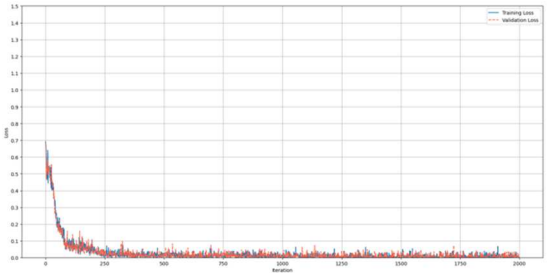

Fig. 6. Loss values of 16×16 image size

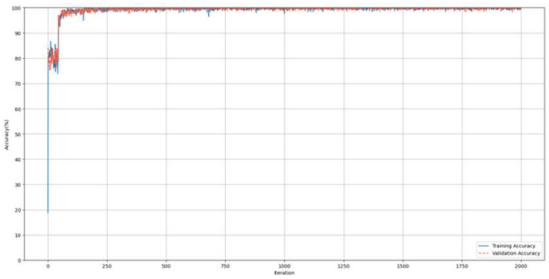

Fig. 7. Accuracy values of 16×16 image size

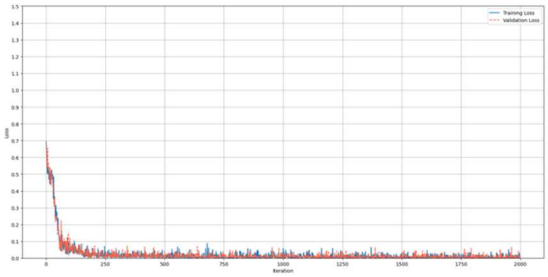

Fig. 8. Loss values of 32×32 image size

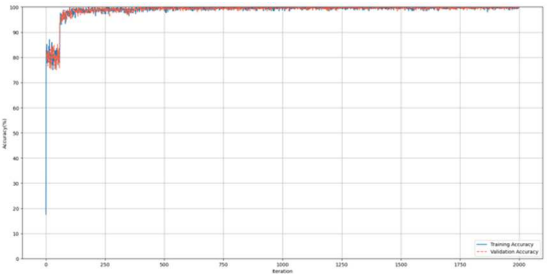

Fig. 9. Accuracy values of 32×32 image size

The training results show that when the image size is 16×16, the classifier model has better convergence. As shown in Figs. 6 and 7.

In the testing phase, the batch size is set to 16000. The precision rate, recall rate, and accuracy of the testing phase are listed in TABLE III. As shown, the Average, Maximum, and Minima values are the results of testing ten times. As indicated in TABLE IV, more than 94% recall rates are obtained, which indicates 94% of abnormal network packets are detected by the proposed classifier. Restated, network security is guaranteed. Meanwhile, packet attributes represented by a 16×16 or a 32×32 image yield almost the same results indicating that abnormal packets detection performance can be guaranteed by 16×16 attributes images.

TABLE III. EXPERIMENTAL RESULTS OF THE TESTING PHASE

Image Size		16×16	32×32
Precision	Maximum	81.140%	81.039%
	Minima	79.901%	79.749%
	Average	**80.325%**	**80.418%**
Recall	Maximum	94.642%	94.526%
	Minima	94.124%	94.183%
	Average	**94.331%**	**94.390%**
Accuracy	Maximum	81.300%	81.327%
	Minima	79.438%	79.695%
	Average	**80.238%**	**80.344%**

V. CONCLUSIONS

Due to the difficulty of feature extraction and selection in intrusion detection fields. This work proposes an effective data preprocessing method combining with a simple CNN model for solving intrusion detection. All packet attributes are imported for data preprocessing including the standard deviation standardization, data quantization, and Zigzag arrangement coding, as well as the image converting to facilitate feature extraction and neural network training.

The experimental results show that the proposed data preprocessing method combined with the simple CNN model provides an effective method in identifying abnormal network packets as indicated in TABLE IV. Meanwhile, it can be seen that if the data undergoes a useful preprocessing procedure, a sound efficiency with more than 94% recall rate of the overall intrusion detection classifier can still be yielded even without a complicated network model. Additionally, packet attributes represented by a 16×16 or a 32×32 image yield almost the same results indicating that applying the proposed preprocessing method enables small size images to achieve the desired abnormal packets detection performance.

REFERENCES

[1] Aminanto, M. E., Kim, H., Kim, K.-M., & Kim, K. (2017). Another fuzzy anomaly detection system based on ant clustering algorithm. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 100*(1), 176-183.

[2] Li, Z., & Qin, Z. (2018). *A semantic parsing based LSTM model for intrusion detection.* Paper presented at the International Conference on Neural Information Processing.

[3] Yang, H., & Wang, F. (2019). Wireless network intrusion detection based on improved convolutional neural network. *Ieee Access, 7*, 64366-64374.

[4] Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review, 53*(8), 5455-5516.

[5] Rezaei, S., & Liu, X. (2019). Deep learning for encrypted traffic classification: An overview. *IEEE communications magazine, 57*(5), 76-81.

[6] Sahu, S. K., Sarangi, S., & Jena, S. K. (2014). *A detail analysis on intrusion detection datasets.* Paper presented at the 2014 ieee international advance computing conference (IACC).

[7] *KDD Cup 1999 dataset*. (1999). Retrieved from: https://kdd.ics.uci.edu/

[8] Lee, W., Stolfo, S. J., & Mok, K. W. (1999). *Mining in a data-flow environment: Experience in network intrusion detection.* Paper presented at the Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining.

[9] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). *A detailed analysis of the KDD CUP 99 data set.* Paper presented at the 2009 IEEE symposium on computational intelligence for security and defense applications.

[10] Alazzam, H., Sharieh, A., & Sabri, K. E. (2020). A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer. *expert systems with applications, 148*, 113249.

[11] Maxfield, C. M. (2008). Chapter 4 - FPGA vs. ASIC Designs. In C. M. Maxfield (Ed.), *FPGAs: Instant Access* (pp. 61-73). Burlington: Newnes.