An Efficient and Lightweight Approach for Intrusion Detection based on Knowledge Distillation

Ruijie Zhao[†¶], Yu Chen[†¶], Yijun Wang[†], Yong Shi[†], and Zhi Xue[†*]
[†]School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China
[¶]Contributed Equally [*]Corresponding Author

ICC 2021 - IEEE International Conference on Communications | 978-1-7281-7122-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICC42927.2021.9500574

***Abstract*—Network intrusion detection (NID) is an important cyber security scheme to identify attacks in network traffic. Recent years, a large amount of studies try to improve the accuracy of the NID by kinds of deep learning approaches. However, these models always require a lot of calculation and space, which constitutes a major hurdle to practical implementation of DL-based models. Thus, lightweight model is imperative, but there are very few applications of DL-based lightweight algorithms in NID models. In this paper, we propose a lightweight knowledge distillation (LKD) model for NID using the idea of knowledge distillation and separable convolution. To the best of our knowledge, it is the first system to use the knowledge distillation approach for NID. The experiment results show that the accuracy of the proposed approach reaches 91.46% and 94.30% on the KDD-CUP99 and UNSW-NB15 datasets respectively. The performance of our model is superior to some approaches based on deep neural network or some machine learning methods. Moreover, both the computational cost and model size of our model are reduced by about 99% compared to the original model.**

***Index Terms*—Intrusion detection, deep learning, knowledge distillation, separable convolution.**

I. Introduction

With the rapid development of information and communication technology, cyber security is becoming more and more important due to the increase of network intrusion attacks. Network intrusion detection (NID) is regarded as a very important security countermeasure to identify malicious activities in network traffic. NID is a typical classification problem, which aims to identify attacks or potential threats hidden in network traffic timely and accurately.

At present, the data analysis methods of NID are classified into misuse detection and anomaly detection. Misuse detection can quickly identify attacks by matching predefined patterns. But it requires manual maintenance of the signature library and can only detect well-known attacks [1]. Anomaly detection, on the contrary, has the ability to identify new attacks. It establishes a statistical model to identify attacks by comparing the activities to be detected with normal activities.

Recent years, a large amount of studies have combined classification tasks with artificial intelligence technology to improve the performance of the model [2]–[6]. With the continuous progress of artificial intelligence technology, the accuracy and stability of NID continue to improve. However, it cannot be ignored that the model size and the consumption of calculation increase rapidly at the same time. This will consume a lot of resources and time in practical application, which is not conducive to the detection of large network traffic. However, there are very few applications of DL-based lightweight algorithms in NID models.

In this article, we propose a lightweight neural network model called lightweight knowledge distillation (LKD) model for NID using the idea of knowledge distillation (KD). Our proposed model includes a teacher network and a student network. The student network adopts separable convolutions to achieve more effective feature extraction while reducing computational cost and model size. We implemente the method respectively and demonstrated its excellent stability and accuracy in binary classification based on the UNSW-NB15 (NB15) and KDD-CUP99 (KDD99) datasets. The main contributions of this article are summarized as follows:

- We propose an efficient and lightweight approach for NID based on KD. To the best of our knowledge, it is the first time to use KD for NID.
- We propose a lightweight student network with separable convolutions for NID. Separable convolutions are key building blocks for our model, which can fully extract data features while reducing computational burden.
- We evaluate our NID model on the NB15 and KDD99 datasets. The results show that the accuracy of anomaly detection is 94.30% and 91.46% respectively. Besides, both floating point of operations (FLOPs) and model size reduce to about 1% of the original model.

The rest of this paper is organized as follows. Section II describes some of the related work in the field of NID and lightweight model. Section III includes the main components of the proposed model. In section IV, we compare our work with other works and analyzed the results. Finally, the conclusion is in section V.

II. Related Works

In the early stages of research, scholars tried to introduce traditional machine learning into NID. They used several popular machine learning approaches, such as Support Vector Machine (SVM), Random Forest (RF) and k-means to implement NID system [7]. However, due to the limitations of shallow learning, the accuracy of these methods is not high enough.

Therefore, deep learning has been introduced into NID these years. Feng et al. [8] proposed a method based on multi-channel LSTM and channel voting, which was verified on the

978-1-7281-7122-7/21/$31.00 ©2021 IEEE

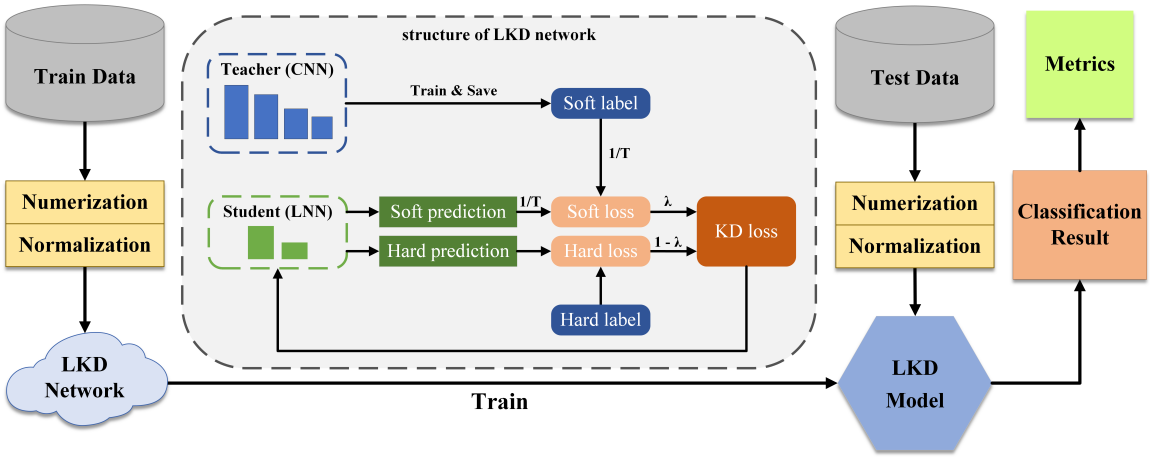


Fig. 1. Proposed framework for NID using KD network.

NSL-KDD dataset with high accuracy. In [9], a multimodal-sequential NID approach based on multimodal deep auto encoder (MDAE) and LSTM was proposed, and got high accuracy in NSL-KDD and NB15 datasets. From the above works, it shows that most researchers only tried to achieve higher detection accuracy by using novel model. However, these models always have high computational complexity. Lightweight model is imperative, but we have not seen relevant research in NID field. Actually, there are several approaches for obtaining small networks, which attract attention. Moblienet uses separable convolutions to build lightweight deep neural networks (DNN) [10]. Besides, Shufflenet also greatly reduces model size and FLOPs by introducing pointwise group convolution, channel split and channel shuffle into DNN [11], [12].

In 2015, Hinton et al. [13] proposed a KD framework. The key idea is to train a large network (teacher), which generates soft labels to provide lots of information for a small network (student), so as to help the student achieve good learning results. Some works focused on improving the quality of KD. Zhang et al. [14] trained a pair of models, distilling knowledge bidirectionally at every epoch. Furlanello et al. [15] suggested to train a teaching assistants network between the teacher and the student. In terms of applications of KD, KD has made achievements in sequence modeling [16], semi supervised learning [17] and domain adaptive [18]. Sanh et al. [19] successfully used KD to make models lighter and faster in the field of natural language processing.

It is believed that we can get a smaller, faster and more accurate model by introducing the idea of separable convolution and KD into NID.

III. The Proposed LKD-based NID Method

In this section, we propose our LKD model using the idea of KD and separable convolution. The framework of proposed model is shown in Fig. 1. Before describing the model, we first introduce the datasets used to evaluate the NID model.

A. Dataset Introduction and Preprocessing

It is not always accurate to evaluate the model with only one dataset, so this article uses two different datasets in experiment. One is the classic KDD99 dataset, the other is NB15 dataset which contains a variety of attacks from the modern real network.

The KDD99 dataset is constructed by processing the original tcpdump data of the 1998 DARPA intrusion detection challenge dataset. It is so classic that many researchers still use it to measure the performance of NID methods. The KDD99 dataset contains 39 kinds of attack types, in which 17 are unknown attack types only appear in the test set to evaluate the generalization performance of the algorithm. In our experiment, we only use 10 percent data of the KDD99, and its composition is detail shown in Table I. Each record in the dataset contains 41 features and a label. The features are divided into 34 continuous features and 7 discrete features. In order to meet the requirements of model input, one-hot encoding is used to deal with discrete variables. As a result, there are 116 features in each record. Then, we normalize all the features to scale them between 0 and 1 by using min-max normalization according to Eq. 1.

$$x^* = \frac{x - MIN}{MAX - MIN} \tag{1}$$

The NB15 dataset is created by Moustafa et al. [20] in 2015 and collected from real network environment. It overcomes some shortcomings existed in KDD99. There are 9 different kinds of attack types in NB15. The specific numbers of normal traffic and attack traffic are also shown in Table I. Each sample in the dataset has 42 features and a label. Similarly, one-hot encoding and min-max normalization are used for data preprocessing, and after that each sample eventually has 196 features to work with.

B. Lightweight KD Model

The LKD model we proposed is based on KD and separable convolution. As Fig. 1 shows, the main idea is to make the

TABLE I
DATASET INFORMATION.

Dataset	Category	Train dataset	Test dataset
KDD-CUP99	Normal	97,278	60,593
	Attacks	396,743	250,436
	Total	494,021	311,029
UNSW-NB15	Normal	56,000	37,000
	Attacks	119,341	45,332
	Total	175,341	82,332

student network get the performance similar to the teacher network through KD. At the same time, the student network is built by separable convolutions to further reduce the model size and FLOPs. The idea behind KD is that soft labels output by the teacher network contain a lot more information than just class labels. Concretely, the teacher network produces a vector of logits: $Z^t(x) = z_1^t(x), z_2^t(x), ..., z_k^t(x)$, for a given input x, and the probability of the vector output through $sigmoid$ layer is: $p_k^t(x) = \frac{1}{1+exp(-z_k^t(x))}$. As the probability is peaky distributions and may carry less information, temperature T is used to "soften" it according to Hinton et al. [13]:

$$p_k^t(x) = \frac{1}{1 + exp(-z_k^t(x)/T)} \tag{2}$$

The whole process of LKD model is shown in Algorithm 1. The student network similarly produces a soft probability $p_k^s(x) = \frac{1}{1+exp(-z_k^t(x)/T)}$ for soft cross entropy loss L_{soft}. Meanwhile, the student network produces a typical cross entropy loss L_{hard} by real labels. Then, the KD loss L_{KD} for the student network is a linear combination of L_{soft} and L_{hard}. By using L_{KD} as loss function, the student network can learn from teacher network.

In order to further compress the model, we construct the student model with separable convolutions. Separable convolution firstly uses 1-channel convolution kernel to do depthwise convolution calculation, and then uses 1×1 convolution to fuse information between different channels. Thus, the computational complexity of separable convolution is much less than that of ordinary convolution. For example, suppose there is a 3×3 convolution layer with 16 input channels and 32 output channels. The parameter of ordinary convolution is $3 \times 3 \times 16 \times 32 = 4608$, while the parameter of separable convolution is $3 \times 3 \times 16 + 1 \times 1 \times 16 \times 32 = 656$, which is much smaller. Therefore, we use separable convolutions to build a lightweight neural network (LNN) model, which are key building blocks for our model.

Finally, we propose our LKD model by combining KD and LNN. Our proposed teacher network is a standard CNN model with batch nomalization and dropout after each convolution layer to prevent over fitting. The student network is LNN model built by separable convolutions.

Algorithm 1 The LKD model for intrusion detection.

Input:
Train dataset D_a
Test dataset D_b
LKD network total training epochs n
Output:
LKD model for intrusion detection

procedure Data_Preprocessing (D_a, D_b)
  Convert the symbolic features in dataset
  D_c, $D_d \longleftarrow$ Compute the Min-Max normalization result of the train dataset and test dataset respectivily (cf. Eq. 1)
return D_c, $D_d \Rightarrow$ the new k–dimensional feature space
procedure Teacher_Model (D_c)
  Load CNN teacher network for intrusion detection
  Input D_c into the teacher network for training
  Use binary cross-entropy as loss function
  Save logits $z^t(x)$ before the final activation layer
  Save $Teacher$ model as the teacher model
return $z^t(x)$, $Teacher$ model
procedure LKD_Model (D_c, $z^t(x)$)
  while $i \leq n$ **do**
    Load LNN student network for intrusion detection
    Input D_c into the lightweight network for training
    Save logits $z^s(x)$ before the final activation layer
    Calculate soft cross entropy loss as L_{soft} based on $z^t(x)/T$ and $z^s(x)/T$
    Calculate hard cross entropy loss as L_{hard} based on Y_{true} and $Y_{predict}$
    $L_{KD} \longleftarrow \lambda \cdot L_{soft} + (1 - \lambda) \cdot L_{hard}$
    Use L_{KD} as loss function
  end while
  Save $Student$ model as the LKD model
return LKD model
Test and save the performance of LKD model on D_d
end

C. Evaluation Metrics

The parameters and related calculation methods used in the evaluation are shown below, where T_p is a true positive example, T_n is a true negative example, F_p is a false positive example, and F_n is a false negative example. Based on the above definition, Accuracy, Recall, Precision and F_1 can be obtained:

$$Accuracy = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \tag{3}$$

$$Recall = \frac{T_p}{T_p + F_n} \tag{4}$$

$$Precision = \frac{T_p}{T_p + F_p} \tag{5}$$

$$F_1 = \frac{2Precision \cdot Recall}{(Precision + Recall)} = \frac{2T_p}{2T_p + F_p + F_n} \quad (6)$$

Besides, to measure the resources consumed by the models, FLOPs, model size and the number of parameters are required. FLOPs is calculated as follows: in the convolution layer, $FLOPs = 2HWC_{out}(C_{in}K^2 + 1)$, where HW is output feature map size, K is kernel size, C_{in} and C_{out} are input channels and output channels. In the full connection layer, $FLOPs = O(2I - 1)$, where I is the number of input neuron and O is the number of output neuron.

IV. Experimental Results

In this section, two stage experiments are designed to evaluate the performance of our proposed LKD model on the KDD99 and UNSW-NB15 datasets. The first stage is to find the suitable hyper-parameter values T and λ to get the best performance of the model. The second stage is to measure performance of LKD model by using the hyper-parameters obtained from the first experiment. All the experiments are performed in Python 3.7 with TensorFlow framework of version 2.1.0 and running on the PC with Intel® Core™ i7-8550U @ 1.80 GHz and 8 GB of RAM.

A. Suitable Hyper-parameters for LKD Model

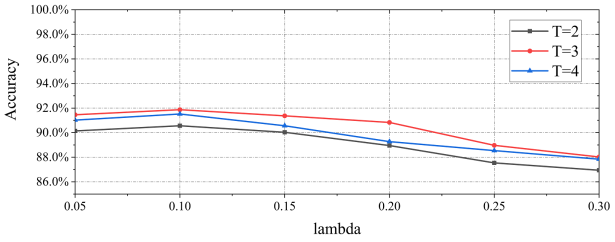
As described in Section III-B, the two hyper-parameters values T and λ determine the effect of KD. Thus, the KD model needs to find the suitable T and λ. We use $sigmoid$ for the output layer and $adam$ for the optimizer. The loss functions of teacher network and student network are binary cross entropy loss and L_{KD} defined in Section III.

The experiment is taken on both KDD99 and NB15 datasets. For KD network a well trained teacher is needed. Firstly, we train a teacher model with good performance. Then, we start training our student model. We change the temperature T from 1 to 3, and the λ from 0.05 to 0.30. As Fig. 2 shows, the results in both datasets are similar. For each hyper-parameters value T, the KD model has the highest accuracy when $\lambda = 0.1$, so this is the best participation of soft labels. Moreover, the curve with $T = 3$ is always above the other curves, which means the soft labels with $T = 3$ carry the largest amount of information. As a result, the best values for T and λ are 3 and 0.1 respectively.

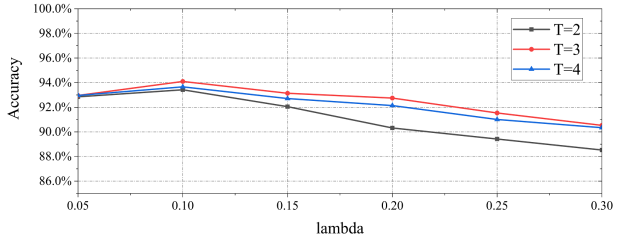
B. Performance of LKD Model for Intrusion Detection

According to the results of the first experiment, we set $\lambda = 0.1$ and $T = 3$ for training the model. To evaluate our proposed LKD model more comprehensively, we perform the binary classification tasks on the teacher network model, LNN model, ordinary-KD (OKD) model, and proposed model respectively. We use the same teacher network in the LKD and OKD model. The student network of the OKD model uses a network structure similar to that of LNN, but only uses standard convolution. The details of the two models' student networks are described in Table II.

The loss values using the two models for KDD99 and NB15 datasets are shown in Fig. 3(a) and 3(b) respectively. We can see that the loss functions of both models decrease rapidly and eventually converge to near zero. Although the loss function of KD is always smaller than that of LKD, they are very close after more than 20 epochs of training.



(a)



(b)

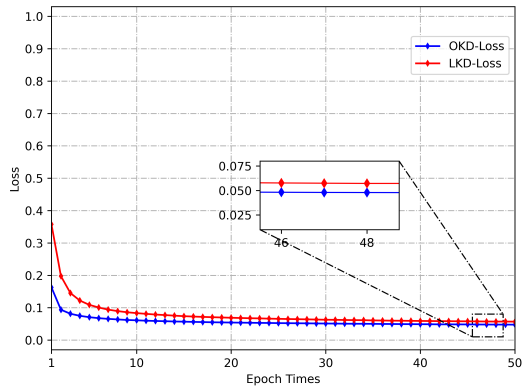
Fig. 2. The test accuracy of LKD models with different T and λ on (a) KDD99 dataset, (b) NB15 dataset.

TABLE II
Structure of Two Models' Student Networks.

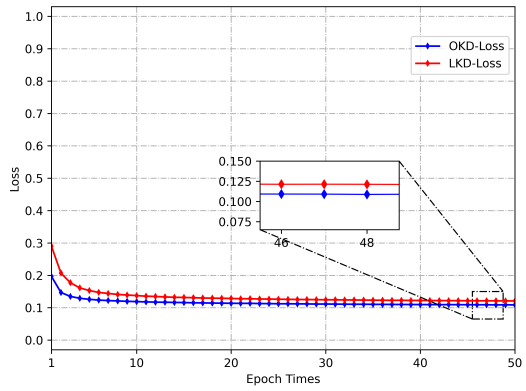
Layer	OKD	LKD
Input	Input	Input
Conv	Conv1D(32, 3)	SeparableConv1D(32, 3)
Downsampling	Maxpool(strides=2)	Maxpool(strides=2)
Conv	Conv1D(64, 3)	SeparableConv1D(64, 3)
Batchnomalization	Batchnomalization	Batchnomalization
Globalpool	Globalpool	Globalpool
Dropout	Dropout(0.3)	Dropout(0.3)
Dense	Dense(1, sigmoid)	Dense(1, sigmoid)

The train accuracy of the two models for both two datasets are shown in Fig. 4. After 5 epochs of training, the train accuracy of the two models is basically convergent. After 40 epochs of training, the accuracy is slightly improved. Similar to the case of loss function, although the train accuracy of OKD is always higher than that of LKD, they are very close.

The above results show that the performance of LKD model is roughly the same as OKD model. In order to further demonstrate the performance of the LKD model. We compare teacher network model, LNN model, OKD model and LKD model in all aspects. According to Table III, in the experiment with KDD99 dataset, the performance of LKD is significantly better than that of LNN, but a little worse than that of teacher. This indicates that LNN can effectively improve its classification

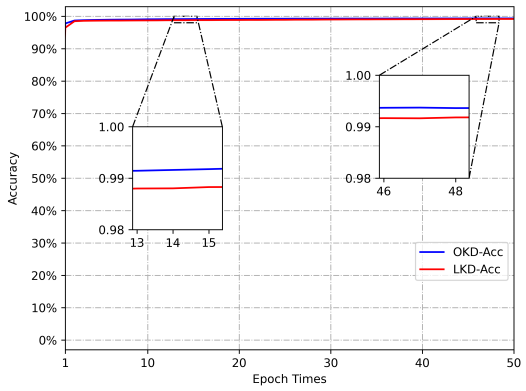


(a)

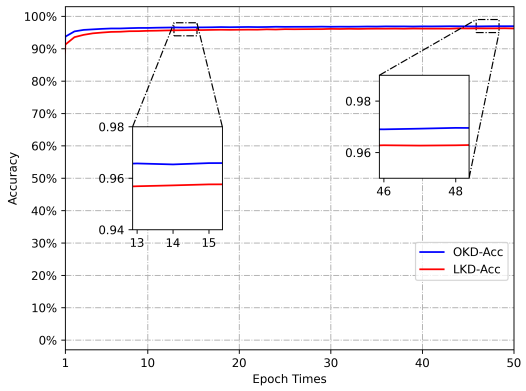


(b)

Fig. 3. Loss values of OKD and LKD on (a) KDD99 dataset, (b) NB15 dataset.



(a)



(b)

Fig. 4. Train accuracy of OKD and LKD on (a) KDD99 dataset, (b) NB15 dataset.

TABLE III
TEST RESULTS OF TEACHER NETWORK, LNN, OKD AND LKD MODEL.

Dataset	Model	Accuracy	Precision	Recall	F_1
KDD99	TeacherNet	92.40%	99.43%	92.40%	95.76%
	LNN	87.82%	99.38%	85.41%	91.87%
	OKD	91.97%	99.29%	90.67%	94.79%
	LKD	**91.46%**	**99.40%**	**89.95%**	**94.44%**
NB15	TeacherNet	94.19%	94.63%	94.19%	94.14%
	LNN	92.95%	97.69%	86.35%	91.67%
	OKD	94.38%	99.04%	86.09%	92.11%
	LKD	**94.30%**	**98.56%**	**88.61%**	**93.32%**

TABLE IV
COMPUTATIONAL COST, MODEL SIZE AND PARAMETERS OF DIFFERENT MODELS.

Dataset	Model	FLOPs	Model Size	Parameters
KDD99	TeacherNet	1,334,803	6,383KB	668,289
	OKD	45,772 (96.57%↓)	110KB (98.28%↓)	22,657 (96.61%↓)
	LKD	**4,725 (99.65%↓)**	**17KB (99.73%↓)**	**2,301 (99.66%↓)**
NB15	TeacherNet	1,396,243	7,479KB	699,009
	OKD	76,492 (94.52%↓)	194KB (97.41%↓)	38,017 (94.56%↓)
	LKD	**27,493 (98.03%↓)**	**73KB (99.02%↓)**	**13,517 (98.07%↓)**

ability by learning knowledge from teacher network. Besides, the four indicators of LKD are similar to those of OKD. For the experiment with NB15 dataset, the situation is similar, except the accuracy of LKD is a little higher than that of teacher network, which means the KD network works very well.

Our proposed model is not only to obtain excellent model performance, but also to reduce the computational cost and model size. Therefore, we compare the FLOPs, model size and parameters of the teacher network, OKD and LKD model as shown in Table IV. Benefiting from the use of the KD network, the computational cost and the model size of the neural network model are significantly reduced. We can see that compared with the teacher network, the computation of OKD

decreases by more than 95%, and the model size decreases by more than 97%. Moreover, the LKD model requires much less computation and model space than OKD. As a result, our proposed LKD model consumes only about 1% of the resources consumed by standard CNN model. According to these two tables, it is proved that our LKD model for NID can effectively reduce the computational cost and the model size, and at the same time achieves high accuracy and detection rates of intrusion detection.

In addition, our approach also compared performance with the conventional approaches in [9], [21], which are the recent literatures providing the detailed performance of various machine and deep learning methods. From the results in Table V, our model achieves the best performance in the metric of accuracy and precision, especially in NB15 dataset where the accuracy is 94.3%, higher than any other models. The results show that our LKD model's performance is similar to the best conventional model or even better, while it has extremely low computational cost and small model size.

TABLE V
THE PERFORMANCE COMPARISON BETWEEN THE OUR PROPOSED MODEL AND OTHER MODELS.

Dataset	Model	Accuracy	Precision	Recall	F_1
KDD99	NB	87.7%	99.4%	85.2%	91.8%
	KNN	92.5%	99.8%	90.9%	95.2%
	SVM	87.7%	99.4%	85.2%	91.8%
	DNN	92.9%	99.9%	91.4%	95.4%
	LSTM	91.5%	97.4%	98.2%	97.8%
	LKD	**91.5%**	**99.4%**	**90.0%**	**94.4%**
NB15	NB	77.3%	85.4%	80.5%	82.9%
	KNN	81.0%	93.2%	77.8%	84.8%
	SVM	65.3%	99.8%	49.2%	65.9%
	DNN	78.4%	94.4%	72.5%	82.0%
	LSTM	94.1%	91.0%	99.1%	94.9%
	LKD	**94.3%**	**98.6%**	**88.6%**	**93.3%**

V. CONCLUSION

In this article, we propose a lightweight NID model called LKD based on KD and separable convolution. We compare our model with the standard CNN model and ordinary KD model on the KDD99 and NB15 datasets. As results show, our model successfully reduces the amount of computational cost and model size to about 1% without reducing the accuracy. Besides, compared with the conventional approaches, the LKD model performs very well, especially in accuracy and precision. As a conclusion, our proposed lightweight NID model is an efficient and lightweight scheme. In the future work, we would like to further improve the performance of the model and reduce the consumption of the model. Besides, we will try to implement a similar scheme on the multi-classification problem for NID.

ACKNOWLEDGMENT

This work was supported by the Foundation Item: Cyber Security from the National Key Research and Development Program of Shanghai Jiao Tong University under Grant 2019QY0703.

REFERENCES

[1] G. Kim, S. Lee, and S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1690–1700, 2014.
[2] H. Yao, P. Gao, P. Zhang, J. Wang, C. Jiang and L. Lu, "Hybrid Intrusion Detection System for Edge-Based IIoT Relying on Machine-Learning-Aided Detection," *IEEE Network*, vol. 33, no. 5, pp. 75–81, 2019.
[3] Y. Wang, L. Guo, Y. Zhao, J. Yang, B. Adebisi, H. Gacanin, G. Gui, "Distributed Learning for Automatic Modulation Classification in Edge Devices," *IEEE Wireless Communications Letters*, doi: 10.1109/LWC.2020.3016822.
[4] M. Liu, Z. Xue, X. Xu, C. Zhong, J. Chen, "Host-Based Intrusion Detection System with System Calls," *ACM Computing Surveys*, vol. 51, no. 5, pp. 1–36, 2018.
[5] G. Gui, M. Liu, F. Tang, N. Kato, and F. Adachi, "6G: Opening new horizons for integration of comfort, security and intelligence," *IEEE Wireless Commun. Mag.*, in press, doi: 10.1109/MWC.001.1900516.
[6] L. Zhang, J. Wu, S. Mumtaz, J. Li, H. Gacanin and J. J. P. C. Rodrigues, "Edge-to-Edge Cooperative Artificial Intelligence in Smart Cities with On-Demand Learning Offloading," *GLOBECOM 2019*, Waikoloa, USA, 2019, pp. 1–6.
[7] C. Tsai, Y. Hsu, C. Lin, and W. Lin, "Intrusion detection by machine learning: A review," *Expert Systems with Applications*, vol. 36, no. 10, pp. 11994–12000, 2009.
[8] J. Feng, et al., "Deep Learning Based Multi-Channel Intelligent Attack Detection for Data Security," *IEEE Transactions on Sustainable Computing*, vol. 5, no. 2, pp. 204–212, 2020.
[9] H. He, X. Sun, H. He, G. Zhao, L. He and J. Ren, "A Novel Multimodal-Sequential Approach Based on Multi-View Features for Network Intrusion Detection," *IEEE Access*, vol. 7, pp. 183207–183221, 2019.
[10] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *CVPR*, Salt Lake City, USA, 2018, pp. 4510–4520.
[11] X. Zhang, X. Zhou, M. Lin and J. Sun, "ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices," *CVPR*, Salt Lake City, USA, 2018, pp. 6848–6856.
[12] N. Ma, X. Zhang, H. Zheng, et al, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," *ECCV*, Munich, Germany, 2018, pp. 116–131.
[13] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *NIPS Deep Learning and Representation Learning Workshop*, Montreal, Canada, 2015.
[14] Y. Zhang, T. Xiang, T. Hospedales and H. Lu, "Deep Mutual Learning," *CVPR*, Salt Lake City, USA, 2018, pp. 4320–4328.
[15] T. Furlanello, Z. Lipton, et al., "Born-again neural networks," *ICML*, Stockholm, Sweden, 2018, pp. 1602-1611.
[16] Yoon Kim and Alexander M. Rush, "Sequence-level knowledge distillation," *EMNLP*, Austin, USA, 2016, pp. 1317–1327.
[17] T. Antti, V. Harri, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *NIPS*, Long Beach, USA, 2017, pp. 1195–1204.
[18] Z. Meng, J. Li, Y. Gong and B. Juang, "Adversarial Teacher-Student Learning for Unsupervised Domain Adaptation," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Canada, 2018, pp. 5949–5953.
[19] V. Sanh, L. Debut, J. Chaumond and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," *NIPS*, Vancouver, Canada, 2019, arXiv:1910.01108.
[20] N. Moustafa, 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 Systems Security*, vol. 25, no. 13, pp. 18–31, 2016.
[21] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.