A Novel Intrusion Detection Method Based on Lightweight Neural Network for Internet of Things

Ruijie Zhao, *Graduate Student Member, IEEE*, Guan Gui, *Senior Member, IEEE*, Zhi Xue, Jie Yin, Tomoaki Ohtsuki, *Senior Member, IEEE*, Bamidele Adebisi, *Senior Member, IEEE*, and Haris Gacanin, *Fellow, IEEE*

***Abstract*—The purpose of a network intrusion detection (NID) is to detect intrusions in the network, which plays a critical role in ensuring the security of the Internet of Things (IoT). Recently, deep learning (DL) has achieved a great success in the field of intrusion detection. However, the limited computing capabilities and storage of IoT devices hinder the actual deployment of DL-based high-complexity models. In this paper, we propose a novel NID method for IoT based on lightweight deep neural network (LNN). In the data preprocessing stage, to avoid high-dimensional raw traffic features leading to high model complexity, we use the PCA algorithm to achieve feature dimensionality reduction. Besides, our classifier uses the expansion and compression structure, the inverse residual structure, and the channel shuffle operation to achieve effective feature extraction with low computational cost. For the multi-classification task, we adopt NID Loss that acts as a better loss function to replace standard cross-entropy loss for dealing with the problem of uneven distribution of samples. The results of experiments on two real-world NID datasets demonstrate that our method has excellent classification performance with low model complexity and small model size, and it is suitable for classifying the IoT traffic of normal and attack scenarios.**

***Index Terms*—Internet of things, intrusion detection, deep learning, lightweight neural network.**

I. Introduction

This work was supported in part by the Cyber Security from the National Key Research and Development Program of Shanghai Jiao Tong University under Grant 2019QY0703, the JSPS KAKENHI under Grant JP19H02142, the Summit of the Six Top Talents Program of Jiangsu under Grant XYDXX-010, the Program for High-Level Entrepreneurial and Innovative Team under Grant CZ002SC19001, Science and Technology Commission of Shanghai Municipality Research Program under Grant 20511102002, the project of the Key Laboratory of Universal Wireless Communications (BUPT) of Ministry of Education of China under Grant KFKT-2020106. (*Corresponding authors: Zhi Xue, Guan Gui*)

R. Zhao and Z. Xue are with the School of Electronic Information and Electrical Engineering, Shanghai Jiaotong University, Shanghai, China (e-mail: ruijiezhao@sjtu.edu.cn, zxue@sjtu.edu.cn).

G. Gui is with the College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China (e-mail: guiguan@njupt.edu.cn).

J. Yin is with Department of Network Security Corps, Jiangsu Electronic Data Forensics and Analysis Engineering Research Center, Jiangsu Provincial Public Security Department Key Lab of Digital Forensics, Jiangsu Police Institute, Nanjing 210031, China. He is also with State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China (e-mail: yinjie@jspi.cn).

T. Ohtsuki is with the Department of Information and Computer Science, Keio University, Yokohama, Japan (e-mail: ohtsuki@keio.jp)

B. Adebisi is with the Department of Engineering, Faculty of Science and Engineering, Manchester Metropolitan University, Manchester M1 5GD, United Kingdom (e-mail: b.adebisi@mmu.ac.uk).

H. Gacanin is with the Institute for Communication Technologies and Embedded Systems, RWTH Aachen University, Aachen, Germany (e-mail: harisg@ice.rwth-aachen.de).

WITH the era of intelligent things, the Internet of Things (IoT) has been widely developed. The application of the IoT involves many fields such as industry, agriculture, and transportation, and effectively promotes the intelligent development of these fields [1]–[6]. The IoT makes the limited resources more reasonable and convenient to use, and improves the efficiency and profit of the industry. However, because IoT devices exchange massive data with external resources, intruders may invade IoT devices that lack appropriate security protection measures. Once the invasion is successful, it may bring huge property losses to the invaded. Hence, designing an appropriate intrusion detection system (IDS) to ensure the security of IoT has become the focus of our attention.

As an essential countermeasure, IDS is commonly utilized to monitor network traffic and gives a prompt alarming message. In general, intrusion detection can be classified into two main categories, namely: host intrusion detection (HID) and network intrusion detection (NID) [7]. Both of them have advantages and their respective limitations. HID uses the internal log of the operating system for audit and judgment. This detection method is not sensitive to network traffic. The system can accurately locate and define the specific operations of intrusions. However, it occupies a lot of resources on the host itself and depends on the reliability of the host. NID analyzes network traffic data, finds suspicious intrusions hidden in the traffic data, and performs corresponding alarms and intercepts on the detected intrusions. Network traffic data has high dimensionality and complex features. In fact, NID is a classification problem. Specifically, it can automatically identify possible attacks and threats hidden in network traffic in time and determine their specific types.

In recent years, numerous research have been conducted on the detection performance of deep learning (DL) in intrusion detection systems [8]–[12]. However, implementation challenges are rarely discussed. Specifically, the challenges of DL-based methods usually include high model complexity and large model size. Due to the limited storage and computing capabilities of IoT devices [13]–[15], it is impractical to deploy highly complex intrusion detection models in IoT devices. Thus, we argue that the NID method for IoT should adhere to the following restrictions:

- **High Classification Performance.** In the real-world IoT network, any intrusion that is not detected may bring huge losses. Therefore, the primary goal of NID methods is to accurately classify the traffic of normal and specific attack scenarios.

- **Low Model Complexity.** IoT devices usually have limited computing capabilities, so it is necessary to adopt the NID method with low model complexity to reduce the computing burden of the device.
- **Small Model Size.** The NID method for IoT must take into account the limited storage space of IoT devices. In other words, the smaller the model size of the classifier, the better.

To address the aforementioned challenges, we propose a lightweight deep neural network (LNN) model for NID, which achieves more efficient feature extraction through lightweight units. The lightweight unit has three special designs for traffic feature extraction, including expansion and compression structure, inverse residual structure and channel shuffle operation. Besides, The computational complexity of the DL-based classifier grows exponentially with number of neurons [16]. This means that the dimensionality of the input of the classifier must be as low as possible. Thus, the PCA algorithm is adopted for feature dimensionality reduction, which can obtain low-dimensional and high-quality traffic data for the classifier. The major contributions of the proposed work are four-fold:

- We propose a novel NID method based on LNN for IoT, where lightweight units are key components for our method. Our method can efficiently extract traffic features while reducing model complexity by expansion and compression structure. At the same time, we also use inverse residual structure and channel shuffle operation to achieve more effective feature extraction.
- To deal with class imbalance in the multi-classification task, we design NID Loss that acts as a better loss function to replace standard cross-entropy loss. Specifically, NID Loss can down weight the contribution of easy samples and pay more attention to hard samples during training.
- The PCA algorithm is introduced to achieve feature dimensionality reduction, which avoids the high complexity of DL-based model caused by high-dimensional features as input.
- We evaluate our method on two real-world NID datasets. Experimental results show that our method has excellent classification performance with low model complexity and small model size, and it is suitable for classifying the IoT traffic of normal and attack scenarios.

The rest of the paper is organized as follows. In Section II, we present the review of related work in the field of intrusion detection. In Section III, we introduce some backgrounds of our proposed lightweight NID method, including feature dimensionality reduction algorithm and lightweight feature extraction network. In Section IV, we describe dataset preprocessing method, and then propose the lightweight NID model for IoT. In Section V, we analyze the training process and the performance of the proposed method on UNSW-NB15 and Bot-IoT datasets. Finally, we conclude this paper in Section VI. The abbreviations used in this paper are summarized in Table I.

II. Related Work

A variety of methods have been developed and implemented in NID to differentiate attacks or threats from normal traffic. Traditional machine learning methods were utilized at the beginning of the research, I. Thaseen *et al*. [17] proposed a method based on the fusion of principal component analysis (PCA) and optimized support vector machine (SVM). Through suggested automatic parameter selection, SVM parameters and kernels were optimized, thereby reducing training time and providing better accuracy for a few attacks such as U2R and R2L. Other popular machine learning approaches such as multilayer perceptron (MLP), Random Forest (RF), and Naive Bayes (NB), have also been used to detect attacks within modern network [18]–[24]. However, the performances of these traditional machine learning methods were limited due to shallow learning, thus cannot provide an effective solution for a considerable number of traffic data. With the development of DL, more and more DL-based models have achieved excellent performance [25]–[31]. C. Yin *et al.* [32] proposed an IDS based on recurrent neural network (RNN). Compared with traditional classification methods, the design method achieved higher accuracy and detection rate in both binary and multiclass classification. H. He *et al.* [33] proposed an intrusion detection model based on long short-term memory (LSTM) and multimodal deep autoencoder to achieve more accurate detection by grouping traffic characteristics. S. Garg *et al.* [34] proposed an IoT intrusion detection model using leverages grey wolf optimization (GWO) and CNN. The experimental results show that this method can obtain minimized features and better detection performance on DARPA98, KDD99 and synthetic datasets. X. Xu *et al.* [35] used log-cosh conditional variational autoencoder (CVAE) to capture the complex distribution of observed data and generate new data with pre-specified classes, which proves to be more effective to generate diverse intrusion data for the imbalanced classes.

Although these DL-based methods have made some progress in the accuracy of intrusion detection, they lack the consideration of computational complexity when the method is actually deployed. Due to the limitation of storage space and computing power, the storage and calculation of neural network models on devices without GPUs is still a huge challenge [39], [40]. Especially the deployment of intrusion detection systems based on federated learning in IoT devices, which will put forward higher requirements on the lightweight performance of the model [41]. Recently, some researchers have paid attention to this problem and proposed some lightweight NID methods. S. I. Popoola *et al.* [36] proposed a method of LSTM autoencoder (LAE) combined with DL algorithm for IoT intrusion detection, which reduces the dimension of traffic data through AE and then input the low-dimensional data to the classifier for classification. Y. Mirsky *et al.* [37] adopts clustering algorithm to divide the traffic features into multiple groups, and independently calculates the reconstruction loss of different characteristic groups through the AE network to reduce the complexity of the model. G. Bovenzi *et al.* [**?**] proposed a two-stage hierarchical NID

TABLE II
COMPARISON WITH DL-BASED RELATED WORK.

Ref.	Feature extractor	Classification scenarios	IoT traffic	Feature reconstruction	Lightweight FE network	Generality evaluation
[32]	RNN	Binary, multi-class	✗	✗	✗	✗
[33]	AE+LSTM	Binary, multi-class	✗	✔	✗	✔
[34]	GWO+CNN	Binary, multi-class	✗	✔	✗	✔
[35]	Log-cosh CVAE+CNN	Binary, multi-class	✗	✔	✗	✗
[36]	LAE+LSTM	Binary, multi-class	✔	✔	✗	✗
[37]	Clustering Algorithm+AE	Binary	✔	✔	✗	✗
[?]	AE+RF	Binary, multi-class	✔	✔	✗	✗
Ours	**PCA+LNN**	**Binary, multi-class**	✔	✔	✔	✔

TABLE I
SUMMARY OF ABBREVIATIONS.

Abbreviations	Notations
IoT	Internet of Things
IDS	Intrusion Detection System
NID	Network Intrusion Detection
HID	Host Intrusion Detection
DL	Deep Learning
ML	Machine Learning
FE	Feature Extraction
LNN	Lightweight Deep Neural Network
CNN	Convolutional Neural Network
SNN	Separable Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
AE	Autoencoder
LAE	Long Short-Term Memory Autoencoder
CVAE	Conditional Variational Autoencoder
GWO	Grey Wolf Optimization
NB	Naive Bayes
SVM	Support Vector Machine
MLP	Multilayer Perceptron
RF	Random Forest
PCA	Principal Component Analysis
GAP	Global Average Pooling
FLOPs	Floating Point Operations

approach, which achieves a lightweight model by inputting each feature of traffic (i.e. a single modality) independently into AE network and finally weighting sum of the per-modality losses.

The aforementioned lightweight NID methods all reduce the complexity of the model by reconstructing traffic features (i.e., reducing dimensionality or grouping). However, in the structural design of the feature extraction network, they still extract features through some conventional and complex DL algorithms. Different from the previous work, we design a novel lightweight deep neural network to extract traffic features, and prove that LNN is a general network for NID through the performance on the UNSW-NB15 and Bot-IoT datasets. Table II summarises the review of DL-based related work for NID.

III. PRELIMINARIES

In this section, we introduce some backgrounds of our proposed lightweight NID method, including feature dimensionality reduction algorithm and lightweight feature extraction network.

A. PCA Algorithm

The feature dimensionality (i.e., the total number of traffic features) is an important factor that directly affects the computational complexity of the DL-based classifier, which grows exponentially with number of neurons [16]. Thus, it is necessary to use feature dimensionality reduction methods on the original traffic data to achieve better lightweight performance. Feature dimensionality reduction is mainly achieved by using linear or non-linear transformation methods for high-dimensional features. AE-based methods employ nonlinear transformation techniques. However, since the reconstruction error is calculated by backpropagation, these methods require multiple rounds of training, which brings more computational cost. As a low-complexity linear transformation technique, the PCA algorithm is more suitable to be adopted in our method. The basic principle of PCA is to eliminate the relevance of the original data variables and recombine a set of uncorrelated variables. The principal components are obtained by linear combination, and the amount of information contained in each principal component can be measured by variance. The greater the variance, the more information it contains. The main steps of the calculation are as follows:

The first step is to calculate the normalized result y_{ij} of each data. The original dataset contains q instances with its corresponding p features. The value of the i^{th} row and j^{th} column of the input matrix is x_{ij}, μ_j is the mean value of the j^{th} dimension vector, and σ_j is the standard deviation of the j^{th} dimension vector.

$$y_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j} \tag{1}$$

The second step is to calculate the covariance matrix C and the eigenvalues $(\lambda_1, \lambda_2, \cdots, \lambda_p)$ and eigenvectors $(\alpha_{i1}, \alpha_{i2}, \cdots, \alpha_{ip})$ of the covariance matrix.

$$C = \frac{\sum_{i=1}^{p} (Y_i, Y_i^T)}{p-1}, (i = 1, 2, \cdots, p) \tag{2}$$

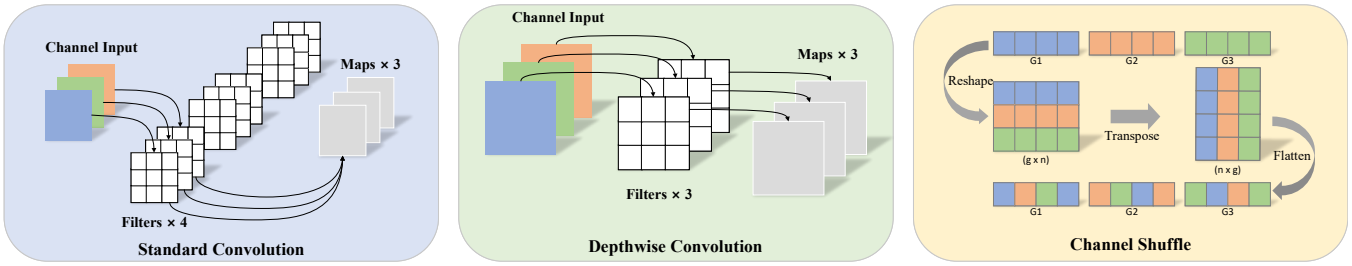


Fig. 1. Key building blocks of the lightweight unit.

The third step is to calculate the contribution rate η according to the eigenvalues.

$$\eta = \frac{\lambda_i}{\sum_{i=1}^{p} \lambda_n} \tag{3}$$

The greater the contribution rate corresponding to the principal component, the more information it contains. According to the contribution rate, take the eigenvectors $(\alpha_{i1}, \alpha_{i2}, \cdots, \alpha_{ik})$ corresponding to the first k largest eigenvalues to form a transformation matrix U with p rows and k columns. The matrix T obtained after dimensionality reduction is:

$$T = XU \tag{4}$$

Through the above steps, we can obtain lower-dimensional data for traffic classification.

B. Lightweight Feature Extraction Network

The lightweight feature extraction network is designed to achieve more efficient feature extraction for NID through a special structure design.

We know from principle of convolution operation that if the tensor dimension is lower, the multiplication calculation amount of the convolutional layer is smaller. Therefore, if the entire neural network is a low-dimensional tensor, the overall calculation speed will be very fast. However, if we just use low-dimensional tensor, the effect will not be satisfactory. The filters of the convolutional layer all use low-dimensional tensors to extract features, so it is difficult to extract enough information. It is necessary to use a method that can have high-dimensional tensors for feature extraction without a lot of computational cost. Many successful lightweight networks use separable convolutions, depthwise convolutions, and special network structural designs (e.g., expansion and compression structure, inverse residual structure, channel shuffle operation, etc.) to achieve more effective feature extraction [45]–[47]. Inspired by these excellent work, our classifier also adopts these lightweight convolutions and structures. Separable convolution splits convolution into two separate layers, namely depthwise convolution and pointwise convolution. The feature map after the depthwise convolution has the same number of channels as the input layer and cannot be expanded. The size of the convolution kernel of pointwise convolution is $1\times1\times M$, and M is the depth of the previous layer. Therefore, the convolution operation here will weight the map of the previous step in the depth direction to generate a new feature map.

Standard convolution takes an $h_i \times w_i \times d_i$ input and an $h_i \times w_i \times d_j$ output. The computational cost of the standard convolutional layer is as follows:

$$h_i \cdot w_i \cdot d_i \cdot d_j \cdot k \cdot k. \tag{5}$$

The computational cost of Separable Convolution is as follows:

$$h_i \cdot w_i \cdot d_i(k^2 + d_j) \tag{6}$$

It can be seen that using depthwise convolution ($k = 3$) for feature extraction, the computational cost is almost 9 times lower than that of standard convolution.

In our lightweight feature extraction network, the lightweight unit is the most important component, which uses a compression and expansion structure to extract traffic features. Key blocks of the lightweight unit are shown in Fig. 1. Before the tensor is input to the depthwise convolution, we first use the standard convolution to expand the number of feature map channels. After extracting features through the depthwise convolution, standard convolution is used to compress the feature map. This structure can greatly reduce the computational cost and effectively extract features. In addition, due to the influence of the distance between the features, it is difficult for convolution network to capture the long-distance feature. Thus, channel shuffle operation is adopted to strengthen the information interaction between different features. In following introduction of the proposed method, we will describe the lightweight unit more specifically.

IV. Proposed LNN-based NID Method for IoT

In this section, we present our LNN-based NID method for IoT. The framework of proposed model is shown in Fig. 2. Our method adopts PCA method to reduce traffic feature dimensionality and lightweight feature extraction network to achieve high classification performance with low computational cost. The process of our approach is summarized in **Algorithm 1**. The details of each processing stage in the proposed method are presented as follows.

A. Dataset

Before introducing the data preprocessing method, we first describe the two intrusion detection datasets used by the LNN-based NID model.

1) The UNSW-NB15 Dataset: The raw network packets of the UNSW-NB15 dataset was created by the IXIA PerfectStorm tool in the Cyber Range Lab of the Australian

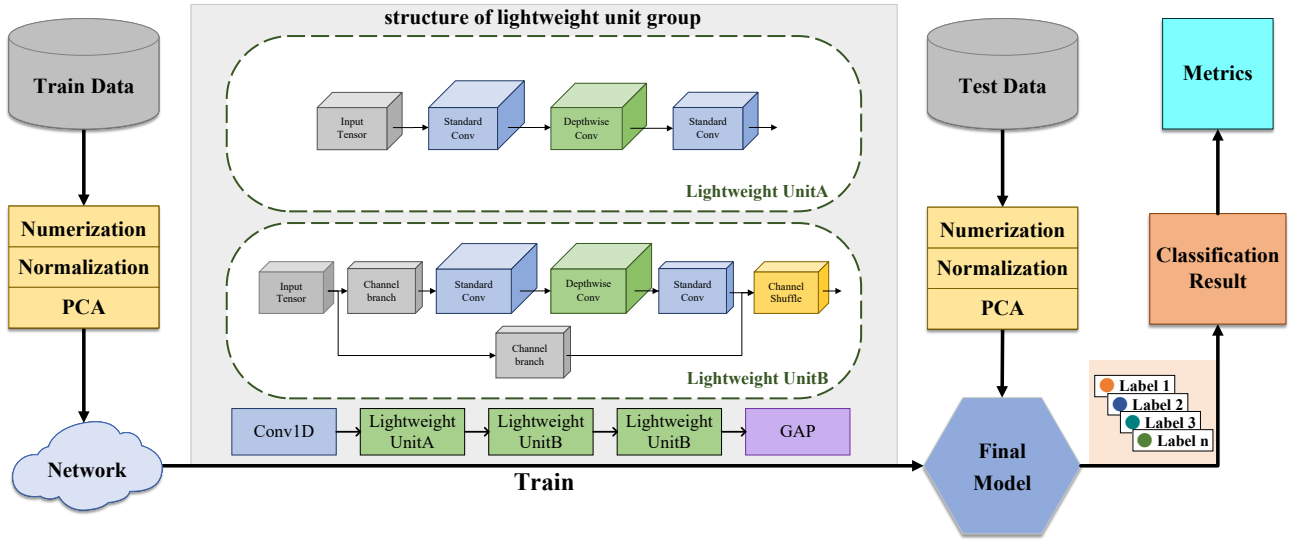


Fig. 2. Proposed framework for LNN-based NID method for IoT.

TABLE III
SUMMARY OF THE DATASETS USED FOR EVALUATION.

Dataset	Category	Training dataset	Testing dataset
UNSW-NB15	Normal	56,000	37,000
	Fuzzers	18,184	6,062
	Analysis	2,000	677
	Backdoors	1,746	583
	DoS	12,264	4,089
	Exploits	33,393	11,132
	Generic	40,000	18,871
	Recon.	10,491	3,496
	Shell	1,133	378
	Worms	130	44
	Total	175,341	82,332
Bot-IoT	Normal	286	191
	DoS	146,293	97,529
	DDos	163,287	108,858
	Recon.	54,649	36,433
	Theft	47	32
	Total	364,562	243,043

Centre for Cyber Security for generating a hybrid of real modern normal activities and synthetic contemporary attack behaviours [42], [43]. It overcomes shortcomings of the KDD99 dataset (e.g., no modern attacks, etc.) and gradually becomes the most widely used dataset in the field of NID in recent years. The number of records in the training dataset is 175,341 and in the testing dataset is 82,332. The UNSW-NB15 dataset has nine types of attacks, namely, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms.

2) The Bot-IoT Dataset: The Bot-IoT dataset is the latest NID dataset for IoT [44]. The network environment incorporated a combination of normal and botnet traffic. It contains normal IoT network traffic and four attack scenarios including DoS, DDoS, Reconnaissance, and Theft. There are many IoT scenarios in the testbed, including a weather station, a smart fridge, motion activated lights, a remote-controlled garage door, and a smart thermostat. Since there are too many traffic records in the raw CSV file of this dataset, we select part of the traffic records for experimental evaluation. The number of traffic records in the training dataset is 364,562 and in the testing dataset is 243,043.

The detailed numbers of different scenarios in the two datasets are shown in Table III.

B. Data Preprocessing

Before the traffic data is applied to the NID model, the data needs to be preprocessed. This involves four steps.

1) **Symbolic feature numerization***:* Convert the symbolic features in dataset such as *protocol*, *service*, and *state* into one-hot encoding.

2) **Label numerization***:* In the multi-classification NID model for IoT, the category label of the traffic data needs to be converted into a numerical representation (e.g., 0 for Analysis, 1 for Backdoor, 2 for DoS, and so on.), and then converted into a one-hot encoding before training.

3) **Normalization***:* Since the traffic data has features of different dimensions, the level of each dimension of the dataset varies greatly. If the original value is directly used for analysis, it will highlight the role of indicators with higher values in the comprehensive analysis, and relatively weaken the role of indicators with low value levels. Therefore, it is necessary to use normalization methods to ensure the reliability of data. The Min-Max normalization is adopted to normalize the data as follows:

$$x^* = \frac{x - MIN}{MAX - MIN} \tag{7}$$

Algorithm 1 Pseudocode of the proposed LNN-based NID method for IoT.

Input:
Training dataset D_a
Testing dataset D_b
Total epoch times n
Output:
LNN model for intrusion detection

procedure Data_Preprocessing (D_a, D_b)
  Convert the symbolic features in dataset
  Transform the category label into a numerical representation
  Compute the Min-Max normalization result of the dataset (7)
  $D_c \longleftarrow$ Reduce the training dataset features by PCA
  $D_d \longleftarrow$ Reduce the testing dataset features by PCA
return D_c, $D_d \Rightarrow$ the new k-dimensional feature space
procedure LNN_Model (D_c, D_d)
while $i \leq n$ **do**
  Load lightweight network
  Input D_c into the lightweight network for training
  if the task is binary classification detection, **then**
    Use binary cross-entropy as loss function
  else
    Use NID Loss as loss function
  end if
  Save NID_i model
end while
Save NID_n model as LNN model
Test and save the performance of LNN model on D_d
return LNN model

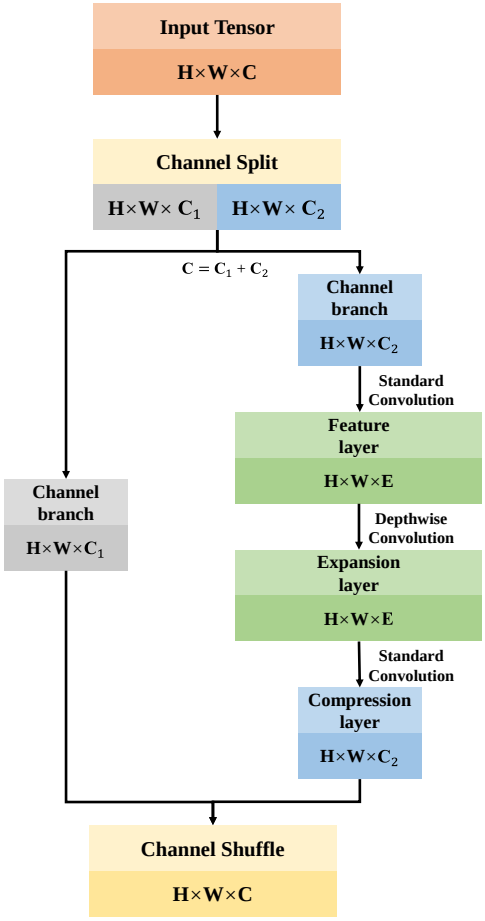


Fig. 3. The structure of lightweight UnitB.

Min-Max normalization technology makes the result fall into the interval $[0, 1]$ through linear transformation of the original data. x is the attribute value, MIN is the minimum of the attribute, and MAX is the maximum of the attribute.

4) **Feature dimensionality reduction***:* The original high-dimensional traffic features will increase the complexity of the classifier (i.e. DL-based model). As a lightweight intrusion detection method, it is necessary to use the dimensionality reduction method to achieve low-dimensional and high-quality traffic data. As mentioned in Section III-A, PCA algorithm can quickly and effectively reduce traffic feature dimensionality. Thus, we use the PCA method to transform the original high-dimensional traffic features into new low-dimensional features through linear transformation.

C. Lightweight NID Model for IoT

In order to deploy intrusion detection systems in IoT devices, some intrusion detection methods have to use traditional machine learning algorithms with low computational complexity and low detection accuracy [14]. However, even if only one intrusion into the IoT system is not detected, it can lead to huge property losses. In our method, we use DL-based neural networks to build our lightweight NID model to accurately detect intrusions.

As shown in Fig. 2, the lightweight NID model for IoT is mainly implemented by lightweight unit. We will first give a specific introduction to the two lightweight units (i.e. Lightweight UnitA and Lightweight UnitB).

Lightweight UnitA is a lightweight network without residual structure, which mainly implements the functions of downsampling and changing the output shape of the tensor. The commonly used downsampling method is to use the maxpooling layer, which is to take the maximum value of the signal in the window and perform translation and non-deformation processing to reduce the signal dimension. Lightweight UnitA can realize the downsampling function when setting the $stride = 2$. The parameter $stride$ is the step size, which represents the step size of the filter for each convolution operation, and determines whether the window of the filter needs to be overlapped. Using this structure to replace maxpooling can effectively reduce computational cost while extracting features.

Lightweight UnitB contains an inverse residual structure, which mainly implements the feature extraction function. The unit structure is shown in Fig. 3. The advantage of using the inverse residual structure is mainly to avoid the problem of model over-fitting and the disappearance of gradient, so we first split the input into two branches of the same size at the beginning of each unit. If the ordinary residual structure

is adopted, the feature map is compressed first, and then the convolution is used to extract the features, which can extract very limited features. Therefore, the input tensor after the channel separation will not be compressed, but will be expanded through the expansion layer. Expansion layer uses 1×1 network structure, the purpose of which is to map low-dimensional space to high-dimensional space. The expansion multiple can be set to achieve a balance between the size of the feature map and the amount of model parameters. After using the depthwise convolution to extract the features, the feature map is compressed through the compression layer using 1×1 network structure. Since the output of the activation function $relu$ for negative inputs is all zero, the conversion from high-dimensional to low-dimensional, using the activation function $relu$ may cause information loss or destruction, so the $relu$ activation function is no longer used in the compression layer but the $linear$ activation function. Finally, the two branches are connected and channel shuffle is used to realize the information exchange between the two branches.

In order to avoid a large number of parameters brought by the fully connected layer, the LNN model uses the global average pooling (GAP) layer instead of the fully connected layer to directly achieve dimensionality reduction. Besides, the dropout approach is used to avert over fitting. As shown in Table III, the number of different categories in the dataset is unevenly distributed. In order to solve this problem, we designed NID loss as a loss function for training in the multi-classification task. In recent years, many successful loss functions have been used to solve the problem of unbalanced distribution of datasets for classification tasks [48], [49]. We refer to the design of the Focal Loss to set different weights α_n for the examples according to the amount of data. Focal Loss is to solve the problem of sample imbalance in binary classification task. In this paper, we extend it to our multi-classification tasks. The calculation formula of the NID Loss function is as follows:

$$NID(p_n) = -\alpha_n(1-p_n)^{\beta}\log(p_n) \tag{8}$$

where p_n is the model estimated probability for the class n. β is used to set the degree of loss value attenuation. The higher the accuracy of the example, the more the loss value attenuates. When $\beta = 0$ and $\alpha_n = 1$, NID loss is equivalent to the standard cross entropy loss. Using NID loss function for training can better learn those examples that are difficult to train.

V. Experimental Results

In this section, several stage experiments are designed to evaluate the performance of the LNN-based NID method for IoT on the UNSW-NB15 and Bot-IoT datasets. We use the other two convolutional neural networks (i.e. CNN and SNN) as comparison DL-based models, both of which have the same number of layers as LNN model. The CNN model uses standard convolutions (i.e. Conv1D) and the SNN model uses some separable convolutions (i.e. S-Conv1D) for feature extraction. The details of three DL models are shown in Table IV. In order to conduct effective and accurate experimental evaluation, each model needs to be run 3 times on both datasets to avoid randomness, and the average performance of the three runs is considered the final result. The following three stage experiments are conducted to verify the performance of the LNN model.

TABLE IV
The Details of Three DL-based Models.

Layer	**CNN**	**SNN**	**LNN**
Input	Input	Input	Input
Layer1	Conv1D (16,3,1)	Conv1D (16,3,1)	Conv1D (16,3,1)
Layer2	Conv1D (64,3,2)	S-Conv1D (64,3,2)	L-UnitA (64,3,2)
Layer3	Conv1D (64,3,1)	S-Conv1D (64,3,1)	L-UnitB (64,3,1)
Layer4	Conv1D (64,3,1)	S-Conv1D (64,3,1)	L-UnitB (64,3,1)
GlobalPool	GAP	GAP	GAP
Output	Output	Output	Output

We set similar filters and kernel size in the convolution of the neural network, which is expressed as (filters, kernel size, stride). The value of the filters of the lightweight unit is the number of input channels for depthwise convolution.

- We evaluate the impact of different time lengths and PCA methods on the performance of the model, and determine the hyperparameter settings of the model.
- We compare the detection performance of various convolutional neural networks and machine learning methods (e.g., SVM, RF and MLP) on the binary classification task. Moreover, to achieve better performance on multi-classification tasks, we use NID Loss as the loss function for training, and analyze the confusion matrices of the classification results.
- To evaluate the lightweight performance of the LNN model, we compare the parameters, model complexity, and model size of different DL models.

To make the experimental evaluation clearer, we introduce the experimental environment and evaluation metrics as follows.

1) Experimental Environment: All the evaluations are conducted in Python 3.6 with the TensorFlow framework of version 2.0.1 and running on the PC with Intel Core i5-8259U@2.30 GHz, 16 GB RAM, and a 512G SSD. The specific configuration and virtual machine implementation environment are shown in Table V.

2) 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

TABLE V
VIRTUAL MACHINE ENVIRONMENT.

Configuration	Description
Ubuntu 18.04	Operating System
Python 3.6	Compiler Environment
Numpy 1.16.3	Extension Library
TensorFlow 2.0.1	Machine Learning Library
Scikit-learn	Machine Learning Library
Keras 2.3.1	Machine Learning Library

F_1 can be obtained:

$$Accuracy = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \tag{9}$$

$$Recall = \frac{T_p}{T_p + F_n} \tag{10}$$

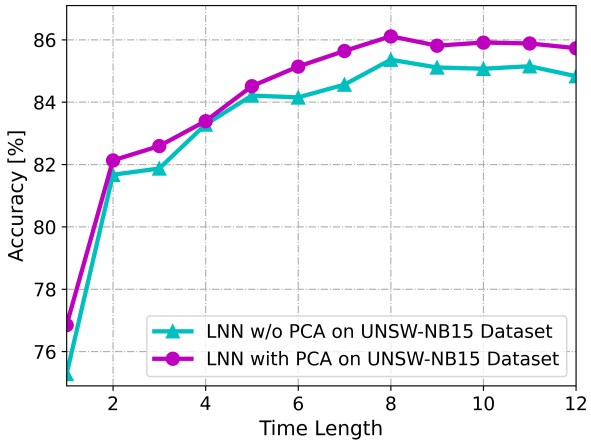
$$Precision = \frac{T_p}{T_p + F_p} \tag{11}$$

$$F_1 = \frac{2Precision \cdot Recall}{(Precision + Recall)} = \frac{2T_p}{2T_p + F_p + F_n} \tag{12}$$

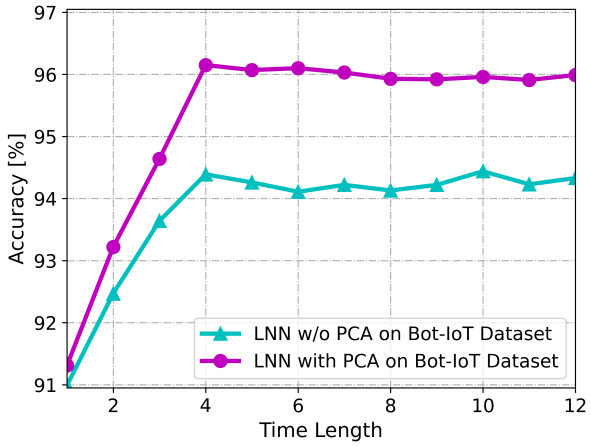
A. Training Process of LNN Model

Before training the classifier, the raw traffic data is preprocessed according to the steps in Section IV-B. We use the binary cross-entropy as the loss function to train the binary classification model and the NID Loss as the loss function to train the multi-classification model. Since the traffic flow is a sequence of packets from a source computer to a destination, multiple continuous flows have more information to reflect the characteristics of the flow at this stage. Thus, we take multiple continuous flows as input features for intrusion detection. The time length represents the number of multiple continuous flows. In addition, the PCA method can reduce the feature dimension and improve the performance of the classifier. In the first stage experiment, we evaluate the impact of time length and PCA method on the classification performance of the multi-classification model.

We vary the value of time length in the set {1,2,3,4,5,6,7,8,9,10,11,12} and check the corresponding multi-classification accuracy on UNSW-NB15 and Bot-IoT datasets. The results are illustrated in Fig. 4. The LNN model achieves the best performance on the UNSW-NB15 dataset when time length is 8 and on the Bot-IoT dataset when time length is 4. Moreover, a dropping trend on both datasets is present as time length increases. For larger time length, LNN basically has no improvement in classification performance and becomes more difficult to train due to large amount of parameters. In addition, it can be seen that the PCA method significantly improves the classification performance of the model. The PCA method maps the raw traffic features from high-dimensional to low-dimensional not only greatly reduces the feature dimensionality, but also enables the classifier to extract features more effectively due to the optimized feature



(a) Multi-classification accuracy on the UNSW-NB15 dataset.



(b) Multi-classification accuracy on the Bot-IoT dataset.

Fig. 4. Impact of time length and PCA method.

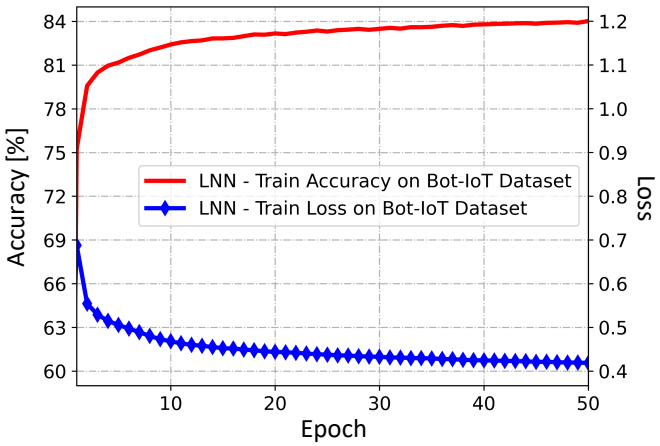
representation. We will use the optimum hyperparameters of the LNN model in this stage for subsequent experiments.

The training accuracy and loss of the LNN model on the UNSW-NB15 and Bot-IoT datasets are shown in Fig. 5. It can be seen that the loss value of each epoch gradually decreases, and finally converges to close to zero. The UNSW-NB15 dataset has more attack scenarios, so the accuracy of the multi-classification model on the UNSW-NB15 dataset is lower than that of the Bot-IoT dataset. However, as a multi-classification model, its accuracy has been very satisfactory. Thus it could be concluded that due to the excellent network structure design, the LNN model can be effectively trained on both datasets and learn the characteristics of traffic in different scenarios well.

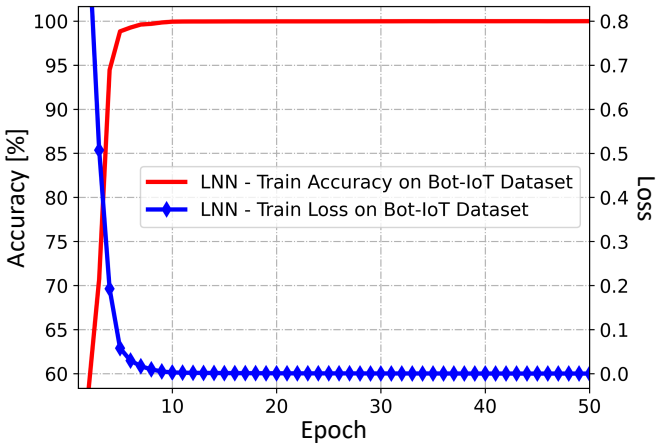
B. Classification Performance of LNN Model

As a misuse detection method for NID in IoT network, the LNN model is trained on benign and malicious traffic samples. Specifically, on the binary classification task, our model can detect normal and malicious traffic, and on the multi-classification task, our model can classify the traffic of normal and specific attack scenarios.

The binary classification task is the basic task of the NID model, which is to distinguish between normal traffic and malicious traffic. It is important to timely and accurately detect

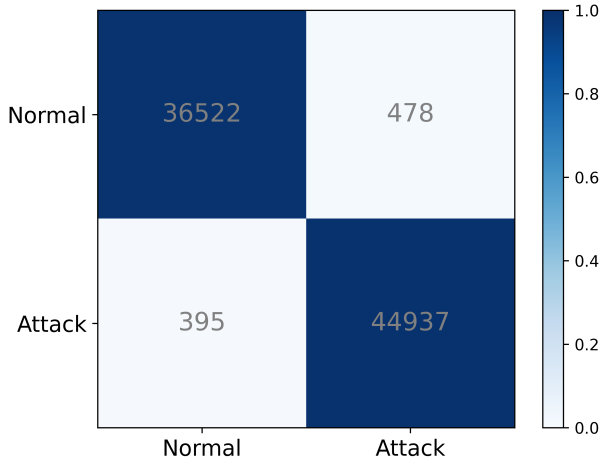


(a) Training process on the UNSW-NB15 dataset.

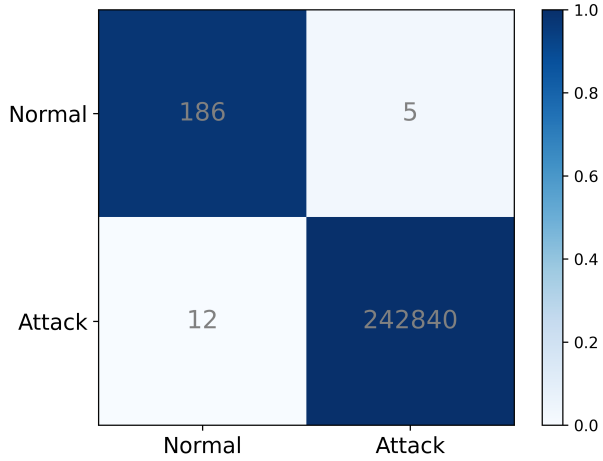


(b) Training process on the Bot-IoT dataset.

Fig. 5. Train accuracy and loss of LNN model on two datasets.



(a) The confusion matrix on the UNSW-NB15 dataset.



(b) The confusion matrix on the Bot-IoT dataset.

Fig. 6. The confusion matrices of LNN model in the binary classification.

malicious traffic in the IoT network. In this stage experiment, we first analyze the binary classification results of the LNN model, and then compare the performance of different models in the binary classification task. The confusion matrices of the binary classification results of the LNN model are shown in Fig. 6. Observe that our approach produces perfect results in the binary classification task, even where the normal samples in the BoT-IoT training dataset is very few, the LNN model can still correctly classify the IoT network traffic.

In addition, we also compare the detailed performance of various deep learning and machine learning methods such as SVM, RF and MLP. For the binary classification task, besides accuracy, we also evaluate the F_1 of the classification result. Since there are very few normal samples in the Bot-IoT dataset, we use macro-F_1 (i.e., the F_1 of all categories is directly averaged regardless of the number of samples) for evaluation. Table VI shows the detailed performance of each binary classification model. Experimental results show that the three DL models (i.e. CNN, SNN and LNN) achieve high binary classification accuracy and macro-F_1. However, machine learning models have not achieved satisfactory detection performance due to insufficient feature extraction capabilities. The F_1 of the proposed LNN model is higher than other DL models on both datasets, and is more than 10% higher than traditional ML models. The excellent detection performance of the LNN model shows that the lightweight unit of LNN can extract traffic features more effectively than standard convolution and separable convolution. In addition, it can be seen that the PCA method can also improve the detection performance of the classifier. Since traditional ML algorithms have not achieved satisfactory detection performance, we will only evaluate neural network models in subsequent experiments.

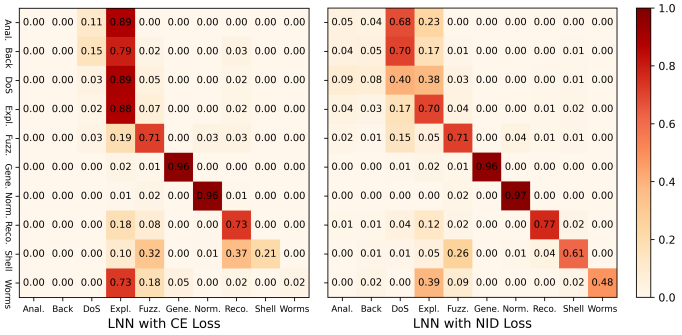
Previous experimental results show that DL-based methods can obtain satisfactory detection performance in binary classification tasks. However, due to the similarity of some attacks and the uneven distribution of samples in the training dataset, accurate classification of different attacks has always been a challenge in intrusion detection research. To comprehensively evaluate the performance of different multi-classification DL models, we compare the classification accuracy, macro-F_1 and weighted-F_1 of CNN, SNN and LNN methods on the UNSW-NB15 and Bot-IoT datasets. In the calculation process, weighted-F_1 performs a weighted average according to the number of samples of each traffic type, and macro-F_1 directly averages the F_1 of each traffic type. In the evaluation experiment of multi-classification performance,

TABLE VII
COMPARISON OF LNN WITH DEEP LEARNING MODELS IN THE MULTI-CLASSIFICATION.

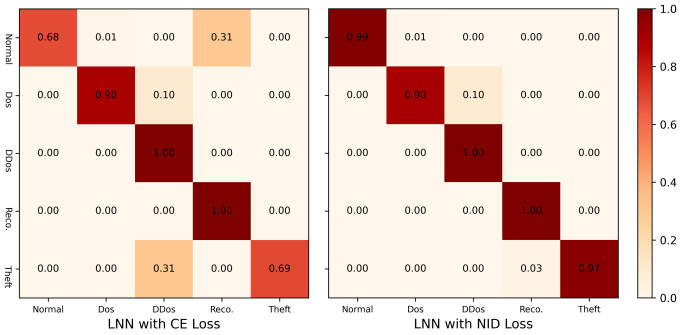
Model	UNSW-NB15			Bot-IoT		
	Acc.	Weighted-F_1	Macro-F_1	Acc.	Weighted-F_1	Macro-F_1
CNN	85.19%	83.86%	43.70%	94.14%	94.09%	90.61%
SNN	82.55%	82.17%	39.34%	91.40%	91.26%	86.26%
LNN	85.97%	84.37%	44.60%	95.83%	95.81%	90.41%
LNN with NID Loss	**86.11%**	**87.02%**	**54.66%**	**96.15%**	**96.14%**	**96.68%**

TABLE VI
COMPARISON OF LNN WITH MACHINE LEARNING AND DEEP LEARNING MODELS IN THE BINARY CLASSIFICATION.

Model	UNSW-NB15		Bot-IoT	
	Acc.	F_1	Acc.	F_1
SVM	80.97%	79.40%	99.66%	61.69%
RF	85.41%	84.66%	99.95%	84.02%
MLP	82.87%	81.68%	99.97%	87.34%
CNN w/o PCA	96.93%	96.88%	99.98%	92.51%
SNN w/o PCA	95.36%	95.26%	99.97%	90.11%
LNN w/o PCA	97.56%	97.52%	99.99%	95.30%
CNN	98.28%	98.26%	99.98%	94.05%
SNN	96.34%	96.28%	99.98%	90.56%
LNN	**98.94%**	**98.93%**	**99.99%**	**98.81%**



(a) The confusion matrices on the UNSW-NB15 dataset.



(b) The confusion matrices on the Bot-IoT dataset.

Fig. 7. The confusion matrices of LNN with the standard cross entropy loss and NID loss in the multi-classification.

we run the model on the testing dataset and evaluate the performance of the model in classifying the traffic of normal and specific attack scenarios. The evaluation results are reported in Table VII. It can be seen that LNN with NID Loss achieves the best performance in both two datasets in terms of all evaluation metrics. We can observe that DL models with standard cross-entropy loss cannot pay attention to categories with a small number of samples, which leads to the classification results of the model with high accuracy but low macro-F_1. In addition, due to the structural limitation of separable convolution, the classification performance of SNN in all tasks is not as good as other DL models. Benefiting from the sophisticated feature extraction unit structure, the LNN model can achieve classification performance close to that of the CNN model, and even outperforms CNN in many evaluation metrics. Furthermore, since NID Loss pays more attention to categories with the samples difficult for classification, the macro-F_1 of the LNN with NID Loss improves significantly.

To further investigate the specific classification of different types of data by the model, we analyze the confusion matrices of the LNN model with the standard cross entropy loss function and the NID loss function. As shown in Fig. 7, LNN model with NID Loss pays more attention to training the hard samples and achieves better performance. On the UNSW-NB15 dataset, it can be seen from Fig. 7(a) that the LNN model with cross entropy loss predicts many attack categories as 'Exploits' attacks. These misclassifications are due to the similarity of these attacks and the lack of attention to difficult samples by the classifier. Compared with LNN using standard cross entropy loss, we observe that the accuracies in 'Worms', 'Shellcode' and 'Dos' are improved 46%, 40% and 37% by NID Loss, for the other classes are improved 1% up to 5%. On the Bot-IoT dataset, NID Loss significantly improves the classification performance of 'Normal' and 'Theft' scenes. However, Dos and DDos are both a type of denial of service attack, so even the LNN model with NID Loss has some confusion in the traffic of these two attack scenarios. Obviously, our NID Loss can enhance the performance of the LNN model in multi-class intrusion detection. Experimental results show that our proposed model is more suitable for deployment in IoT devices due to its lightweight performance, and it has high accuracy in detecting various attacks to ensure system security.

TABLE VIII
COMPUTATIONAL COST AND MODEL SIZE.

Model	UNSW-NB15			Bot-IoT		
	Parameters	FLOPs	Model Size	Parameters	FLOPs	Model Size
CNN w/o PCA	37,914	374,204	496kB	29,669	123,742	401kB
SNN w/o PCA	19,914	230,204 (38.48%↓)	292kB (41.13%↓)	11,669	51,742 (58.19%↓)	195kB (51.37%↓)
LNN w/o PCA	13,050	186,940 (50.04%↓)	283kB (42.94%↓)	5,045	30,110 (75.67%↓)	190kB (52.62%↓)
CNN	29,946	246,716 (34.07%↓)	402kB (18.95%↓)	28,901	117,598 (4.97%↓)	392kB (2.24%↓)
SNN	11,946	102,716 (72.55%↓)	199kB (59.88%↓)	10,901	45,598 (63.15%↓)	186kB (53.61%↓)
LNN	**5,082**	**59,452 (84.11%↓)**	**190kB (61.69%↓)**	**4,277**	**23,966 (80.63%↓)**	**181kB (54.86%↓)**

C. Lightweight Performance of LNN Model

Our method aims not only to achieve excellent classification performance, but also focuses on lightweight performance for better deployment in IoT devices. Since the IoT devices usually have limited computing capabilities and communication resources, we conduct experiments to compare the model complexity and model size of different DL-based models. The model complexity is evaluated by floating point operations (FLOPs), which counts the amount of calculation of the model. In addition, due to the different number of features in data of UNSW-NB15 and Bot-IoT datasets, the lightweight performance of the model is evaluated on the two datasets.

We can see from Table VIII that our proposed model has the minimal FLOPs on both datasets. Compared with the CNN w/o PCA model, the complexity of the LNN model is reduced by 84.11% on the UNSW-NB15 dataset and 80.63% on the Bot-IoT dataset. Our proposed method also has the smallest model size, which can effectively reduce the occupancy of the limited storage space of the IoT device. Moreover, benefiting from the use of the PCA algorithm in the data preprocessing stage, model complexity and model size are further reduced. Especially for the LNN model on the UNSW-NB15 dataset, compared with the LNN w/o PCA model, the model complexity and model size are reduced by 68.20% and 32.86% respectively. Thus, it could be concluded that both lightweight feature extraction network and PCA contribute to the lightweight performance of LNN to a considerable extent, and our method provides a lightweight NID solution for IoT.

VI. CONCLUSION

In this paper, we adopt PCA algorithm to reduce the dimensionality of traffic features, and propose a lightweight feature extraction network for NID. Lightweight units are key building blocks for our model, as it has capacity to fully extract data features while reducing computational cost by expanding and compressing feature maps. At the same time, we also use inverse residual structure and channel shuffle operation to achieve more effective feature extraction. For the multi-classification task, we train our model through NID Loss, which can pay more attention to difficult samples and deal with the problem of uneven distribution of samples. The experimental results show that our proposed model not only has excellent performance in intrusion detection, but also can substantially reduce computational cost and model size. Our method can therefore provide a feasible solution for intrusion detection in IoT devices.

In future research, we will focus on the actual deployment of intrusion detection methods and carry out real world experiments in IoT devices. In addition, we will conduct research on intrusion detection datasets. We plan to build an intrusion detection dataset that is close to the actual situation in terms of attack types and methods.

REFERENCES

[1] J. Wang, C. Jiang, H. Zhang, Y. Ren, K. -C. Chen, and L. Hanzo, "Thirty years of machine learning: the road to pareto-optimal wireless networks," *IEEE Commun. Surv. Tutor.*, vol. 22, no. 3, pp. 1472–1514, 2020.

[2] Z. Wang, Y. Liu, Z. Ma, X. Liu, and J. Ma, "LiPSG: Lightweight privacy-preserving Q-learning-based energy management for the IoT-enabled smart grid," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 3935–3947, May 2020.

[3] Z. Cai and T. Shi, "Distributed query processing in the edge-assisted IoT data monitoring system," *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12679–12693, Aug. 2021.

[4] M. R. Palattella, M. Dohler, A. Grieco, G. Rizzo, J. Torsner, T. Engel, and L. Ladid, "Internet of things in the 5G era: enablers, architecture, and business models," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 510–527, Mar. 2016.

[5] N. Ahmed, D. De, and I. Hussain, "Internet of things (IoT) for smart precision agriculture and farming in rural areas," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4890–4899, Dec. 2018.

[6] S. Chavhan, D. Gupta, B. N. Chandana, A. Khanna, and J. Rodrigues, "IoT-based context-aware intelligent public transport system in a metropolitan area," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6023–6034, Jul. 2020.

[7] M. Liu, Z. Xue, X. Xu, C. Zhong, and J. Chen, "Host-based intrusion detection system with system calls: review and future trends," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–36, Nov. 2018.

[8] Z. Wu, J. Wang, L. Hu, Z. Zhang, and H. Wu, "A network intrusion detection method based on semantic re-encoding and deep learning," *J. Netw. Comput. Appl.*, vol. 164, artcile number: 102688, Oct. 2020.

[9] F. Tang, Y. Kawamoto, N. Kato, and J. Liu, "Future intelligent and secure vehicular network towards 6G: machine-learning approaches," *Proc. IEEE*, vol. 108, no. 2, pp. 292–307, Feb. 2020.
[10] J. Li, M. Liu, Z. Xue, X. Fan, and X. He, "RTVD: A real-time volumetric detection scheme for DDoS in the internet of things," *IEEE Access*, vol. 8, no. 1, pp. 36191–36201, Feb. 2020.
[11] C. Xu, J. Shen, and X. Du, "A method of few-Shot network intrusion detection based on meta-learning framework," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 3540–3552, May 2020.
[12] G. Muhammad, M. S. Hossain, and S. Garg, "Stacked autoencoder-based intrusion detection system to combat financial fraudulent," *IEEE Internet Things J.*, early access, doi: 10.1109/JIOT.2020.3041184.
[13] Z. Cai and Q. Chen, "Latency-and-coverage aware data aggregation scheduling for multihop battery-free wireless networks," *IEEE Trans. Wireless Comm.*, vol. 20, no. 3, pp. 1770–1784, Mar. 2021.
[14] 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, May 2019.
[15] A. Ikpehai, B. Adebisi, K. M. Rabie, K. Anoh, R. E. Ande, M. Hammoudeh, H. Gacanin, and U. M. Mbanaso, "Low-power wide area network technologies for internet-of-things: a comparative review," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2225–2240, Apr. 2019.
[16] R. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," *Advances in neural information processing systems (NIPS)*, pp. 2377–2385, 2015.
[17] I. Thaseen, C. Kumar, "Intrusion detection model using fusion of PCA and optimized SVM," in *International Conference on Contemporary Computing and Informatics*, Mysore, India, Nov. 27–29, 2014, pp. 879–884.
[18] C. Tsai, Y. Hsu, C. Lin, and W. Lin, "Intrusion detection by machine learning: A review," *Expert Syst. Appl.*, vol. 36, no. 10, pp. 11994–12000, Dec. 2009.
[19] N. Moustafa, B. Turnbull, and K. R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4815–4830, Jun. 2019.
[20] K. Johnson, K. Thongam, T. De, "Entropy-based application layer DDoS attack detection using artificial neural networks," *Entropy*, vol. 18, no. 10, Oct. 2016.
[21] S. Lee, D. Kim, J. Park, "A hybrid approach for real-time network intrusion detection systems," in *International Conference on Computational Intelligence and Security*, Harbin, China, Dec. 15-19, 2007, pp. 712–715.
[22] M. Usha, P. Kavitha, "Anomaly based intrusion detection for 802.11 networks with optimal features using SVM classifier," *Wireless Netw.*, vol. 23, no. 8, pp. 2431–2446, Aug. 2017.
[23] H. Wang, J. Gu, S. Wang, "An effective intrusion detection framework based on SVM with feature augmentation," *Knowl. Based Syst.*, vol. 139, pp. 130–139, Nov. 2017.
[24] H. H. Pajouh, R. Javidan, R. Khayami, A. Dehghantanha, and K. R. Choo, "A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks," *IEEE Trans. Emerg. Topics Comput.*, vol. 7, no. 2, pp. 314–323, Feb. 2019.
[25] 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," in *GLOBECOM*, Waikoloa, USA, Dec. 9–13, 2019, pp. 1–6.
[26] G. Gui, M. Liu, F. Tang, N. Kato, and F. Adachi, "6G: Opening new horizons for integration of comfort, security and intelligence," *IEEE Wireless Comm. Mag.*, vol. 27, no. 5, pp. 126–132, May 2020.
[27] Z. Xiong, H. Xu, W. Li, and Z. Cai, "Multi-source adversarial sample attack on autonomous vehicles," *IEEE Trans. Veh. Technol.*, vol. 70, no. 3, pp. 2822–2835, Mar. 2021.
[28] M. E. Aminanto, R. Choi, H. C. Tanuwidjaja, P. D. Yoo, and K. Kim, "Deep abstraction and weighted feature selection for Wi-Fi impersonation detection," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 3, pp. 621–636, Mar. 2018.
[29] Y. Wang, G. Gui, H. Gacanin, T. Ohtsuki, O. A. Dobre, and H. V. Poor, "An efficient specific emitter identification method based on complex-valued neural networks and network compression," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2305–2317, Aug. 2021.
[30] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, "Deep learning based inference of private information using embedded sensors in smart devices," *IEEE Network*, vol. 32, no. 4, pp. 8–14, Jul. 2018.
[31] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y. Pan, "Generative adversarial networks: a survey towards private and secure applications," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1–38, Jul. 2021.
[32] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, no. 1, pp.21954–21961, Dec. 2017.
[33] 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, no. 1, pp. 183207–183221, Dec. 2019.
[34] S. Garg, K. Kaur, N. Kumar, G. Kaddoum, A. Y. Zomaya, and R. Ranjan, "A hybrid deep learning-based model for anomaly detection in cloud datacenter networks," *IEEE Trans. Netw. Serv. Manage.*, vol. 16, no. 3, pp. 924–935, Sept. 2019.
[35] X. Xu, J. Li, Y. Yang and F. Shen, "Towards effective intrusion detection using log-cosh conditional variational autoencoder," *IEEE Internet Things J.*, vol. 8. n0. 8, pp. 6187–6196, Apr. 2021.
[36] S. I. Popoola, B. Adebisi, M. Hammoudeh, G. Gui, and H. Gacanin, "Hybrid deep learning for botnet attack detection in the internet of things networks," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4944–4956, Mar. 2021.
[37] Y. Mirsky, T. Doitshman, Y. Elovici and, A. Shabtai, "Kitsune: an ensemble of autoencoders for online network intrusion detection," in *NDSS*, San Diego, USA, Feb. 18–21, 2018, pp. 1–15.
[38] G. Bovenzi, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescape, "A hierarchical hybrid intrusion detection approach in IoT scenarios," in *GLOBECOM*, Taipei, Taiwan, Dec. 7–11, 2020, pp. 1–7.
[39] Y. Wang, J. Yang, M. Liu, and G. Gui, "LightAMC: lightweight automatic modulation classification via deep learning and compressive sensing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3491–3495, Mar. 2020.
[40] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," in *ICCV*, Venice, Italy, Oct. 22–29, 2017, pp. 2755–2763.
[41] S. A. Rahman, H. Tout, C. Talhi, and A. Mourad, "Internet of things intrusion detection: centralized, on-device, or federated learning?" *IEEE Network*, vol. 34, no. 6, pp. 310–317, Jun. 2020.
[42] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference*, Canberra, Australia, Nov. 10–12, 2015, pp. 1–6.
[43] N. Moustafa, J. Slay, and G. Creech, "Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks," *IEEE Trans. Big Data*, vol. 5, no. 4, pp. 481–494, Dec. 2019.
[44] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset" *Future Gener. Comp. Syst.*, vol. 100, pp. 779–796, 2019.
[45] F. Chollet, "Xception: deep learning with depthwise separable convolutions," in *CVPR*, Honolulu, USA, July 21–26, 2017, pp. 1800–1807.
[46] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: inverted residuals and linear bottlenecks," in *CVPR*, Salt Lake City, USA, Jun. 18–23, 2018, pp. 4510–4520.
[47] N. Ma, X. Zhang, H. Zheng, and J. Sun, "ShuffleNet V2: practical guidelines for efficient CNN architecture design," in *ECCV*, Munich, Germany, Sept. 8–14, 2018, pp. 122–138.
[48] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Trans. Pattern. Anal. Mach. Intel.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.
[49] B. Li, Y. Liu, and X. Wang, "Gradient harmonized single-stage detector," in *AAAI*, Hawaii, USA, Jan. 27–Feb. 1, 2019.

Rujie Zhao (Graduate Student Member, IEEE) received the M.S. degree from the Shanghai Jiao Tong University, Shanghai, China, in 2021. He is currently pursuing the Ph.D. degree in the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China. His research interests include deep learning, network security, internet of things, etc.

Guan Gui (Senior Member, IEEE) received the Ph.D. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2012. From 2009 to 2014, he joined the Tohoku University as a research assistant as well as a post-doctoral research fellow, respectively. From 2014 to 2015, he was an Assistant Professor in the Akita Prefectural University, Akita, Japan. Since 2015, he has been a professor with Nanjing University of Posts and Telecommunications, Nanjing, China.

His recent research interests include artificial intelligence, deep learning, non-orthogonal multiple access, wireless power transfer, and physical layer security. Dr. Gui has published more than 200 IEEE Journal/Conference papers and won several best paper awards, e.g., ICC 2017, ICC 2014 and VTC 2014-Spring. He received the IEEE Communications Society Heinrich Hertz Award in 2021, the Member and Global Activities Contributions Award in 2018, the Top Editor Award of IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY in 2019, the Outstanding Journal Service Award of KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEM in 2020, the Exemplary Reviewer Award of IEEE COMMUNICATIONS LETTERS in 2017, the 2012 Japan Society for Promotion of Science (JSPS) Postdoctoral Fellowships for Foreign Researchers, and the 2018 Japan Society for Promotion of Science (JSPS) International Fellowships for Overseas Researchers. He was also selected as for the Jiangsu Specially-Appointed Professor in 2016, the Jiangsu High-level Innovation and Entrepreneurial Talent in 2016, the Jiangsu Six Top Talent in 2018, the Nanjing Youth Award in 2018. Dr. Gui was recognized as one of the 2020 Highly Cited Chinese Researchers in wireless communications. He is serving or served on the editorial boards of several journals, including IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEICE Transactions on Communications, Physical Communication, Wireless Networks, IEEE ACCESS, Journal of Circuits Systems and Computers, Security and Communication Networks, IEICE Communications Express, and KSII Transactions on Internet and Information Systems, Journal on Communications. In addition, he served as the IEEE VTS Ad Hoc Committee Member in AI Wireless, Executive Chair of VTC 2021-Fall, Vice Chair of WCNC 2021, TPC Chair of PHM 2021, Symposium Chair of WCSP 2021, General Co-Chair of Mobimedia 2020, TPC Chair of WiMob 2020, Track Chairs of EuCNC 2021, VTC 2020 Spring, ISNCC 2020 and ICCC 2020, Award Chair of PIMRC 2019, and TPC member of many IEEE international conferences, including GLOBECOM, ICC, WCNC, PIRMC, VTC, and SPAWC.

Zhi Xue received his B.S. and Ph.D. degrees from Shanghai Jiao Tong University, Shanghai, China, in 1992 and 2001, respectively. He is currently a Professor with the School of Electronic Informationand and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China. His research interests include machine learning, deep learning, network security, data science, etc.

Jie Yin received his B.S. degree from Jiangsu Police Institute and M.S. degree in software engineering from Nanjing University of Technology, Nanjing, China, in 1997 and 2008, respectively. He is currently a senior engineer with the Department of Network Security Corps, Jiangsu Electronic Data Forensics and Analysis Engineering Research Center, Jiangsu Provincial Public Security Department Key Lab of Digital Forensics, Jiangsu Police Institute, Nanjing, China. He is also with State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China. His research interests include deep learning, and its applications in network security and internet of things.

Tomoaki Ohtsuki (Senior Member, IEEE) received the B.E., M.E., and Ph. D. degrees in Electrical Engineering from Keio University, Yokohama, Japan in 1990, 1992, and 1994, respectively. From 1994 to 1995 he was a Post Doctoral Fellow and a Visiting Researcher in Electrical Engineering at Keio University. From 1993 to 1995 he was a Special Researcher of Fellowships of the Japan Society for the Promotion of Science for Japanese Junior Scientists. From 1995 to 2005 he was with Science University of Tokyo. In 2005 he joined Keio University. He is now a Professor at Keio University. From 1998 to 1999 he was with the department of electrical engineering and computer sciences, University of California, Berkeley.

He is engaged in research on wireless communications, optical communications, signal processing, and information theory. Dr. Ohtsuki is a recipient of the 1997 Inoue Research Award for Young Scientist, the 1997 Hiroshi Ando Memorial Young Engineering Award, Ericsson Young Scientist Award 2000, 2002 Funai Information and Science Award for Young Scientist, IEEE the 1st Asia-Pacific Young Researcher Award 2001, the 5th International Communication Foundation (ICF) Research Award, 2011 IEEE SPCE Outstanding Service Award, the 27th TELECOM System Technology Award, ETRI Journal's 2012 Best Reviewer Award, and 9th International Conference on Communications and Networking in China 2014 (CHINACOM '14) Best Paper Award. He has published more than 205 journal papers and 415 international conference papers. He served as a Chair of IEEE Communications Society, Signal Processing for Communications and Electronics Technical Committee. He served as a technical editor of the IEEE Wireless Communications Magazine and an editor of Elsevier Physical Communications. He is now serving as an Area Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and an editor of the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS. He has served as general-co chair, symposium co-chair, and TPC co-chair of many conferences, including IEEE GLOBECOM 2008, SPC, IEEE ICC 2011, CTS, IEEE GLOBECOM 2012, SPC, IEEE ICC 2020, SPC, IEEE APWCS, IEEE SPAWC, and IEEE VTC. He gave tutorials and keynote speeches at many international conferences including IEEE VTC, IEEE PIMRC, IEEE WCNC, and so on. He was Vice President and President of the Communications Society of the IEICE. He is a senior member and a distinguished lecturer of the IEEE, a fellow of the IEICE, and a member of the Engineering Academy of Japan.

Bamidele Adebisi (Senior Member, IEEE) received his Bachelor's degree in electrical engineering from Ahmadu Bello University Zaria, Nigeria, in 1999, and his Masters degree in advanced mobile communication engineering and Ph.D. degree in communication systems from Lancaster University, United Kingdom, in 2003 and 2009, respectively. He was a senior research associate with the School of Computing and Communication, Lancaster University, from 2005 to 2012. He joined Manchester Metropolitan University in 2012, where he is currently a Full Professor (Chair) of intelligent infrastructure systems. He is the current Vice Chair of IEEE TC-PLC; was General Chair, IEEE ISPLC'18, UK; Co-Chair, 6th IEEE Int'l Conference on Smart Grid Communications, 2015, Miami, US, etc. He is a Panel Member of the UK Engineering and Physical Sciences Research Council (EPSRC) Peer Review College, and an EU H2020 Expert Reviewer/ rapporteur. He has been part of multi-partner, multi-country, multi-million pounds projects as PI and Co-I. One of his projects with an SME received the 2020 UK Best Knowledge Transfer Partnership Project of the Year Awards. He has published over 140 peer-review papers and given several talks/panel discussions in the research areas of Internet of Things, smart cities, smart grids, communication systems and cyber physical systems. Bamidele is a Fellow of IET, a Fellow of Higher Education Academy and a Chartered Engineer.

Haris Gacanin (Fellow, IEEE) received his Dipl.-Ing. degree in Electrical engineering from the University of Sarajevo in 2000. In 2005 and 2008, respectively, he received MSc and Ph.D. from Tohoku University in Japan. He was with Tohoku University from 2008 until 2010 first as Japan Society for the Promotion of Science (JSPS) postdoctoral fellow and later, as an Assistant Professor. He joined Alcatel-Lucent Bell (now Nokia Bell) in 2010 as a Physical-layer Expert and later moved to Nokia Bell Labs as Department Head. Since April 2020, he joined RWTH Aachen University. He is a head of the Chair for Distributed Signal Processing and co-director of the Institute for Communication Technologies and Embedded Systems.

His professional interests are related to broad areas of digital signal processing and artificial intelligence with applications in wireless communications. He has 200+ scientific publications (journals, conferences and patents) and invited/tutorial talks. He is a fellow of IEEE. He was a Distinguished Lecturer of IEEE Vehicular Technology Society and an Associate Editor of IEEE COMMUNICATIONS MAGAZINE, while he served as the editor of IEICE Transactions on Communications and IET Communications. He acted as a general chair and technical program committee member of various IEEE conferences. He is a recipient of several Nokia innovation awards, IEICE Communications Society Best Paper Award in 2021, IEICE Communication System Study Group Best Paper Award (joint 2014, 2015, 2017), The 2013 Alcatel-Lucent Award of Excellence, the 2012 KDDI Foundation Research Award, the 2009 KDDI Foundation Research Grant Award, the 2008 JSPS Postdoctoral Fellowships for Foreign Researchers, the 2005 Active Research Award in Radio Communications, 2005 Vehicular Technology Conference (VTC 2005-Fall) Student Paper Award from IEEE VTS Japan Chapter and the 2004 Institute of IEICE Society Young Researcher Award.