Arabian Journal for Science and Engineering (2024) 49:3623–3632
https://doi.org/10.1007/s13369-023-08092-1

RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

A Novel Cyber Security Model Using Deep Transfer Learning

Ünal Çavuşoğlu[1] · Devrim Akgun[1] · Selman Hizal[2]

Received: 9 January 2023 / Accepted: 30 April 2023 / Published online: 24 July 2023

Abstract
Preventing attackers from interrupting or totally stopping critical services in cloud systems is a vital and challenging task. Today, machine learning-based algorithms and models are widely used, especially for the intelligent detection of zero-day attacks. Recently, deep learning methods that provide automatic feature extraction are designed to detect attacks automatically. In this study, we constructed a new deep learning model based on transfer learning for detecting and protecting cloud systems from malicious attacks. The developed deep transfer learning-based IDS converts network traffic into 2D preprocessed feature maps.Then the feature maps are processed with the transferred and fine-tuned convolutional layers of the deep learning model before the dense layer for detection and classification of traffic data. The results computed using the NSL-KDD test dataset reveal that the developed models achieve 89.74% multiclass and 92.58% binary classification accuracy. We performed another evaluation using only 20% of the training dataset as test data, and 80% for training. In this case, the model achieved 99.83% and 99.85% multiclass and binary classification accuracy, respectively.

Keywords Network security · Intrusion detection system · Deep learning · Transfer learning · VGG16

1 Introduction

Because most computer networks are vulnerable to security and privacy threats, many intrusion detection systems (IDS) have been developed in recent years. The majority of developed machine learning techniques are ineffective in dealing with various attack types. For this reason, researchers continue to implement IDS systems using deep learning-based models. This section presents state-of-the-art studies on IDS systems that use deep learning architectures. Zhang et al. [1] proposed an efficient end-to-end network intrusion detection multi-layer representation learning model using deep neural networks (CNN) using gcForest and a P-Zigzag data encoding system. Li et al. [2] offered an approach based on a multi-CNN process for intrusion detection. The multi-CNN fusion model has achieved, 86.95%, 76.67% for the binary classification, and 81.33% and 64.81% for the multi-class classifications on the test sets. Roy and Cheung [3] proposed a new IDS for IoT system that is consist of long short-term memory (LSTM) in bidirectional structure and has a 95% accuracy. Lin et al. [4] suggested a CNN-based model where each sample's character is represented as a vector to form the input matrix. Zhang et al. [5] proposed an IDS model that focuses on optimizing imbalanced data. The Synthetic Minority Oversampling Technique (SMOTE) is employed to add artificial samples to minority groups. Naseer et al.[6] suggested various IDS models. LSTM and DCNN models performed 85% and 89% accuracy, respectively on the NSL-KDD test dataset. Xin et al. [7] discuss the ML and DL network security strategies. The authors mainly presents the newest ML and DL applications. The USTC-TRC2016 flow dataset is used in this analysis, and the data preprocessing kit USTCTK2016 is created. Through reviewing the eight experimental findings and the overall accuracy of classifiers is 99.41%, according to experimental data. Manimaran et al. [8] created a novel model of the CNN-NIDS, introducing the latest work on network anomaly detection using various learning approaches. The methodology of deep learning traditionally employing standard NIDS approaches is examined as well. Shone et al. [9] suggested a non-symmetric

✉ Ünal Çavuşoğlu
unalc@sakarya.edu.tr

Devrim Akgun
dakgun@sakarya.edu.tr

Selman Hizal
selmanhizal@subu.edu.tr

[1] Software Engineering Department, Sakarya University, Sakarya, Turkey

[2] Computer Engineering Department, Sakarya University of Applied Sciences, Sakarya, Turkey

deep autoencoder (NDAE) for unsupervised functional learning methods for intrusion detection. The suggested attack detection model was implemented on TensorFlow using the NSL-KDD and KDD Cup'99 datasets, assessing its attack detection efficiency.

Three preprocessing strategies for a CNN are evaluated in [12]. Lopez et al. [13] suggested new deep reinforcement learning (DRL) models for IDS using a labeled dataset. In contrast to current machine learning techniques, this study shows that DRL can boost intrusion detection results. Kasongo and Sun [14] built the Deep Learning methodology that combines feed-in deep neural networks (FFDNN), employing a information gain (IG) filter-based functional selection method. Kasongo and Sun [15] in another study, introduced a wireless IDS classifier which employs several layers of LSTM units to detect intrusions into wireless networks. The average accuracy using validation dataset is 99.51%, the F-Score is 99.43%, and the test data set accuracy is 86.99%. Yin et al. [16] presented IDS model based on RNN and the NSL-KDD dataset was used to train and analyze the model. Zhiqiang et al. [17] developed an IDS using a deep learning model and they used the UNSW-NB15 dataset and achieved good results. Wang [18] tested cutting-edge algorithms using the Deep Learning methodology on the NSL-KDD data set. The possible weaknesses of neural networks used in IDS have been tested in a lab setting. Specific features' functions in producing adversarial examples are investigated. Parampottupadam and Moldovann [19] focused on real-time IDS using deep learning. They explored the potential of deep learning-based binary and multiclass classification to detect in real-time using a cloud-based representation framework. Vinayakumar et al. [20] proposed an IDS and warning system based on a highly scalable architecture. For real-time data analysis, the architecture uses a distributed model with DNNs. Zhang et al. [21] proposed IDS based on deep autoencoder which is used to compress less significant characteristics and extract important features. Zhang et al. [22] propose an IDS model using a redesigned genetic algorithm (GA) and a deep belief network. The NSL-KDD dataset is used to train and assess the model and procedures. Su et al. [23] present the innovative BAT-MC model uses two-phase bidirectional LSTM learning and time series functionality for intrusion detection. The output of the BAT-MC technique is evaluated using the KDD Test datasets. Zhang et al. [24] developed a Baesian CNN deep learning model and increase overall performance. Furthermore, CNN and SVM are used to create baseline models for comparison. Ujjan et al. [25] offered the sampling of sFlow and adaptive polls along with the Snort IDS. The evaluation of the suggested system shows a compromise between the accuracy of attack detection and resource overhead. Hassan et al. [26] presented an IDS based on CNN-WDLSTM model. The experimental findings show that they achieved 97.1% accuracy in the UNSW-NB15 large dataset. Parra et al. [27] proposed an IDS for IoT devices connected to the cloud for identifying and mitigating phishing and Botnet attacks. Gamage and Samarabandu [28] evaluated four basic learning models on CIC-IDS2017 and CIC-IDS2018 datasets for the intrusion classification; deep neural network, LSTM, an autoencoder, and a deep belief network. Priyadarshini et al.[29] developed a deep learning based IDS system against DDoS attacks for protecting fog networks. The models are trained on the Hogzilla dataset before being tested against a real-time DDoS attack. Kasim [30] developed a model based on Autoencoder and SVM and trained the model with CICIDS dataset.Despite the uneven data collection, 99.1% of detection tests are successful. Cavusoglu [31] proposed a hybrid IDS using an integration of some machine learning based IDS and feature selection methods is proposed to detect diverse attacks. The performance tests of the proposed system were carried out according to different criteria and in comparison to the related literature and studies. Ferraq et al. [32] conducted a comparison of deep learning algorithms for IDS. They focus on deep discriminative, generative and unsupervised techniques, and all deep learning approaches on the CSE-CIC-IDS2018 and the Bot-IoT datasets. Elmasry et al. [33] presented a particle swarm-based technique that is utilized prior to training to choose the optimum attributes and hyperparameters of the deep learning model automatically. Kasongo and Sun [34] proposed a deep learning model to identify intruders in a wireless IDS environment. Zhang et al. [35] examined a deep-seated intrusion detection technique (IDs) of a vehicle, which utilizes gradient decrease with momentum and gradient decrease, to improve IDS performance and accuracy. Binbusayyis and Vaiyapuri [36] provided an unsupervised intrusion detection deep learning technique. Contrary to the conventional IDS approach, the classification system is extracted and trained in two different steps. Tian et al. [37] introduced a solution to intrusion detection using an extended deep belief network on the two popular data sets: the UNSW-NB15 and NSL-KDD. Al-Qatf et al. [38] suggest a deep model that combined a Sparse Auto Encoder and SVM. They use Self-Taught Learning (STL) and SVM to improve data representation and classification, respectively. Yang et al. [39] proposed a model that extracts features using a Deep Belief Network and classifies using an SVM. They demonstrated the model's success by comparing it to other baseline approaches based on SVM, DBN, Principal Component Analysis and SVM. Mushtaq et al.[40] suggested a hybrid IDS that makes use of a deep automated encoder and a bidirectional LSTM and tested it on the NSL-KDD dataset. Rani et al. [41] proposed a model for IDS based on deep neural network that offers a solution to the class imbalance at the classifier level. They analyzed the effect of class imbalance and extensively tested this system on NSL-KDD and UNSW-NB15 datasets. Naseri and

Gharehchopogh [42] presented a binary implementation of the farmland fertility algorithm for feature selection in classifying IDS. Ding and Li [43] developed and IDS based on Graph Convolution Networks (GCN) used for feature extraction and LSTM used for modeling the changes. Ahmad et al.[44] suggested an intrusion detection method using an ensemble classifier to test unknown attack instances. The ensemble model was made up of various classifiers such as autoencoder, CNN, and LSTM models. They trained their model on multiple datasets, including NSL-KDD, and produced good results for the evaluated cases.

According to the most recent reviews, machine learning approaches, particularly deep learning, are frequently used in IDS. The performance of the suggested methods has been demonstrated by performing tests on traffic data collected from many different datasets or natural environments. In these datasets, the NSL-KDD, which includes several attack types and has abundant sample space, is widely used for research. The preprocessing and normalization operations increase the performance of the datasets. In some designs, the performance evaluations vary considerably according to the algorithm used. The performance is relatively low due to the incompatibility of the dataset and the method used. In some studies, relatively high-performance values are obtained when a percentage of the training dataset is used for testing. Therefore, performance values still need to be improved for both binary and multiclass classification applications and, we developed an IDS for better classification performance.

2 Background

Developing an intrusion detection system involves various components. First, we explained the general working principle of IDS architectures. Then we presented the basics of the DNN and transfer-learning model structure. The details of the NSL-KDD Dataset were also given, and the types of attacks and their explanations were introduced. Finally, the performance metrics for evaluating the proposed IDS system have been given.

2.1 Network Intrusion Detection Systems

It has become an inevitable need for all beings to transfer data using a communication network. It is essential to take precautions and ensure security and detect malicious attacks for this communication network. IDS such as firewalls, antivirus software, or other measures are used to secure communication systems. However, the detected attack results must be evaluated by an expert person or system. IDS fails to detect an attack-type that has never been encountered before. Anomaly-based IDS is effectively used to protect target systems against new malicious activities. A general example of IDS architecture is shown in Fig. 1. One or more attackers carry out harmful intrusions on servers or data centers over the cloud. The firewall covers the attacks first. It allows the passage of packets that meet the specified rules. Otherwise, the packets are blocked and dropped. If the firewall accepts a packet, it is then checked by an IDS. Then, notifications are made to a management system where control results are automatically evaluated. If there is an attack, this system prevents the attack, or the user is allowed to access the server.

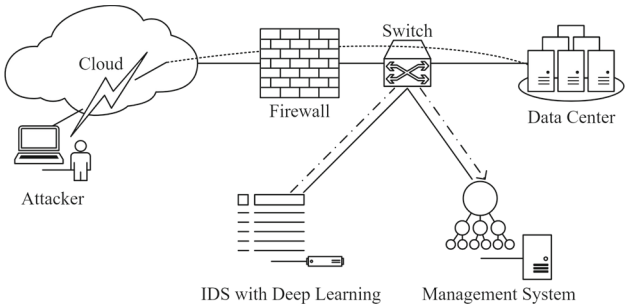

Fig. 1 Anomaly-based Network IDS Architecture

2.2 Deep Neural Networks

The majority of today's machine learning applications depend greatly on neural networks. They can be trained with the help of backpropagation algorithms for a specific problem providing useful input features and labels. Input features should be handcrafted carefully to provide generality for increasing the success of the model. Deep learning models offer an advanced substitute for the problems where features required for the application are hard to figure out. Deep Learning contains feature extraction layers, for example, convolutional neural networks (CNNs) and often demands a large amount of training data. CNN layers can extract desired features according to problem type, and they are trained together with dense layers for classification. Some of the features extracted with CNNs are common in various cases, especially when the datasets are large enough and can be utilized in another problem. Deep transfer learning is the application of using weights from a trained network for implementing another deep learning architecture. For this purpose, there are various popular models for computer vision applications such as VGG16, ResNet50, and Inception. VGG16 is one of the famous examples of deep transfer learning applications. It is made up of 13 convolutional layers and three dense layers. Although it is mainly used for computer vision applications, there are also other applications where input features are expressed with feature matrices

Table 1 Traffic statistics in the NSL-KDD test/train datasets

NSL-KDD Dataset	Normal	DoS	U2R	R2L	Probe	Total
NSL-KDD Train+	67343	45927	52	995	11656	125973
%20 NSL-KDD Train+	13449	9234	11	209	2289	25192
NSL-KDD Test+	9711	7458	200	2654	2421	22544

2.3 NSL-KDD Dataset

The NSL-KDD dataset is suggested for solving various problems with the KDD'99 [11]. The quantity of samples in this dataset has been reduced for improved accuracy, and the traffic has been filtered. Table 1 shows the sample numbers of normal and attack traffic types in the dataset. There are 41 different attributes in total. The NSL-KDD contains five classes according to the activity or targets of the attacker performing the cyber-attack.

DoS (Denial of Service) Attacks: This attack occurs most frequently in the NSL-KDD dataset. It can be defined as cyber-attacks that are used to prevent normal users from receiving service by sending more connection requests to a server than the number of requests it can handle, causing the server to be unable to respond or shut down to protect itself. For example, during the COVID-19 pandemic, where distance education has become more widespread, DoS attacks are being carried out to prevent students from accessing online exams in higher education institutions. Although distributed servers are used in such exam systems, over time, these attacks affect the network infrastructure and make the system completely unresponsive.

R2L (Remote-to-Local) Attacks: A cyberattack that aims to enter a remote computer without authorization as a guest or another user. To obtain access to the victim's computer, a variety of ways have been used. R2L attacks are attempts to achieve local access to a remote system. This victim machine is usually a personal computer or a server with permission to access the server.

U2R (User-to-Root) Attacks: U2R attacks provide unauthorized access to cloud system servers. These types of trench attacks are usually carried out within the network system where the servers are accessed and try to gain the root user authority by taking advantage of a vulnerability or vulnerability in the server. For example, they are attacks where a user who has permission to enter the servers providing online content in the university management system but does not have administrator privileges obtains administrator privileges and performs unauthorized operations.

Probe Attacks: This is a cyber-attack to learn important information about a server or any machine on the network. For example, before hacking a database server in a management information system, some critical information is tried to be obtained. These are the active ports used by the relevant server or important information such as the IP address, operating system, type of IDS used, are investigated by a probing attack.

Table 2 The structure of confusion matrix

		Predicted Intrusions	Normal	Total
Actual	Intrusion	TP	FN	$TP + FN$
	Normal	FP	TN	$FP + TN$
	Total	$TP + FP$	$FN + TN$	N

2.4 Performance Metrics

Metrics that are frequently used in IDS performance evaluation are explained in this section. First, the structure of the confusion matrix is given. Then, we briefly explain how the performance metrics are calculated. Table 2 displays the typical representation of confusion matrix.

The confusion matrix contains the variables TP, TN, FP, and FN, as described below. Once these variables are determined, we can compute those equations ranging from 1 to 5 used for performance evaluations.

- **TP (true-positive):** positive sample quantity positively classified.
- **TN (true-negative):** negative sample quantity negatively classified.
- **FN (false-negative):** positive sample quantity negatively classified.
- **FP (false-positive):** negative sample quantity positively classified.

1. Accuracy: It is the proportion of the sample count accurately categorized to the overall sample count. Equation 1 shows its computation.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

2. Precision: It is the proportion of the true positive sample count to the overall positive sample count. Equation 2 shows how to compute it.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

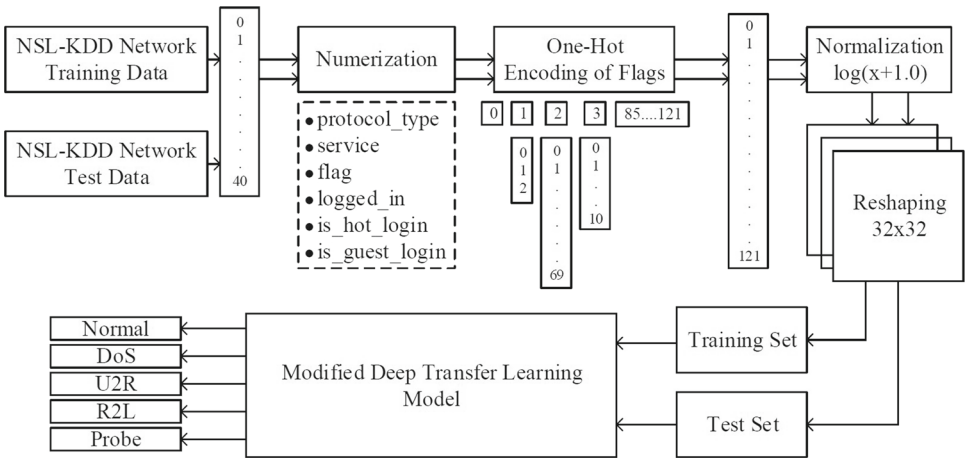

Fig. 2 The proposed Anomaly-based IDS

3. Recall: As shown in Eq. (3), it is the proportion of the true positive sample count to the total of true positive and false negative sample counts.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

4. Specificity: As shown by Eq. (4), it is the proportion of the true negative sample count to the sum of true negative and false positive sample counts.

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{4}$$

5. F-Score: Recall and precision values are utilized to produce a new value in this evaluation criterion. The F-Score is calculated using the harmonic mean formula where the variables are the precision and recall values, as indicated in Eq. (5).

$$\text{F\_Score} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \tag{5}$$

3 The Proposed IDS Based on Deep Transfer Learning

The Fig. 2 shows the subcomponents, connections between them, and input/output parameters of the proposed model. It performs intrusion detection for multiclass classification and the same model also performs binary classification by assuming all attack types to be a single attack type. For the binary classification system, the traffic is categorized as attack or normal. Figure 3 illustrates the main components of the suggested Deep Transfer Learning-based method, which is the most important component of the proposed system. We also used these models to produce a single output for normal and attack traffic for binary classification by replacing the sofmax function with sigmoid.

Numerization: Service, protocol type, and flag features were converted to numeric. Service types include seventy symbolic values, and these are converted to 0, 1, …, 69. Protocol type contains three symbolic values: ICMP, TCP, and UDP were converted to 0, 1, and 2, respectively. Similarly, flag which contains eleven symbolic values are also converted to 0, 1,..,10.

One-Hot Encoding Flags: Integer values belonging to service, protocol type, and flag features were turned into binary representation through one-hot encoding. Therefore 41 features are expanded to 122 features.

Normalization: In machine learning applications, it is a common practice to scale input values to a reasonable range. The ranges in the NSL-KDD dataset exhibit significant differences; hence logarithmic normalization was utilized.

Reshaping: We converted the network traffics into two-dimensional data like images so that our deep transfer learning model can operate on it. The CNN-based model reduces the input data's dimensionality via pooling operations or stride size of the convolution. Hence they are usually trained for some determined image dimensions. VGG16 requires its input size to be at least 32×32 so that the model can process input. In order to satisfy the input dimensions, the values are repeatedly filled to obtain the feature matrices. Figure 4 shows the example visualizations for the generated and reshaped network traffic features.

Train/Test: The NSL-KDD has two parts for train and test. The train part was employed to train the developed model, and the test part was employed to assess its performance. Figure 5 illustrates two test cases for NSL-KDD dataset. For the first case we used NSL-KDD dataset fully and in the other case 20% of train as test data.

Deep Transfer Learning Model: Figure 3a shows the designed model where the weights of the frozen layers were transferred from the VGG16 model which contains various numbers of convolutions in the form of blocks. Note that only the first four block transferred and first and last

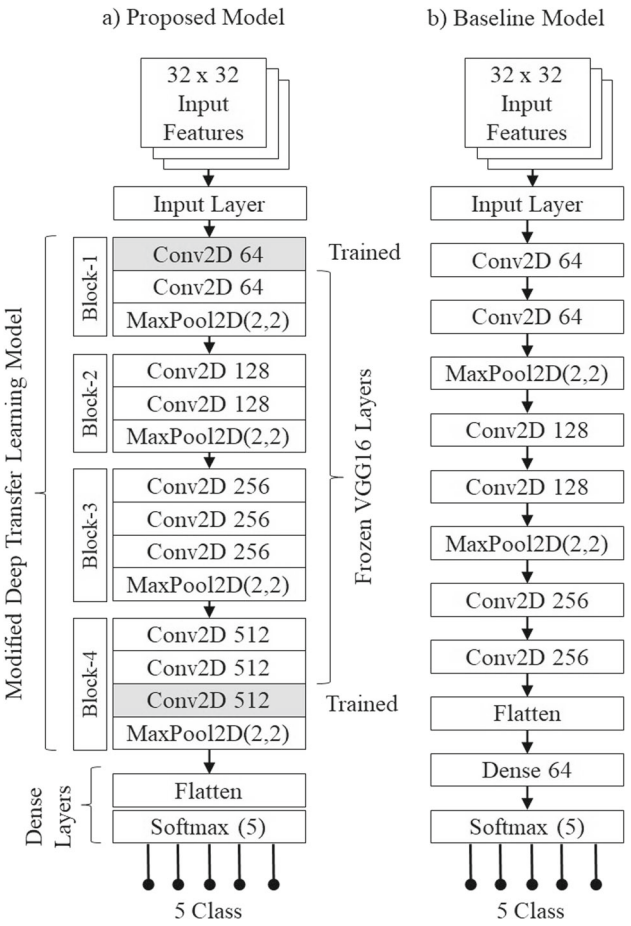

Fig. 3 **a** Deep transfer learning model, and **b** the baseline CNN model for comparison

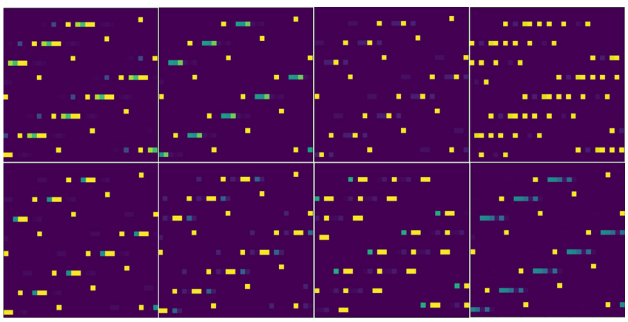

Fig. 4 Example reshaped 32×32 network traffic features

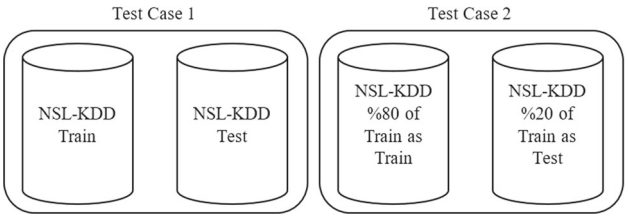

Fig. 5 Testing cases for NSL-KDD

convolution block are fine tuned. There are a total of ten two-dimensional convolutions in the proposed model, and two of them are included in the training process. One of them is selected as the first convolutional layer. The convolutional layer weights can be more compatible with the input feature vectors before transferring them to the following frozen layers. The other is selected as the last convolutional layer for it selects features specific to the IDS model. Finally, a single fully connected layer was used for classification. For multiclass-classification, there are five outputs with softmax activation functions. For binary classification, there is only one output with a sigmoid activation function. Figure 3b shows the baseline CNN model where all the convolutional layers included in training. We replaced the softmax activation function layer with a sigmoid for binary classification evaluations.

4 Experimental Evaluations

4.1 Experimental Setting

The Table 3 summarizes the hardware and software used to develop the new IDS system. All the codes to make experiments were written in Python, and specifically, Keras, a general framework for implementing deep learning applications, was used for training and testing. Performance measurements were done on the Ubuntu operating system, which runs on hardware with a CPU model, AMD Ryzen 2700 eight-core processor with 16 GB memory. The GPU model is NVIDIA GeForce®GTX 1080 with 8 GB memory. The developed models were trained with the RMSprop optimizer for both multiclass and binary class classifications, using the categorical and binary cross-entropy functions, respectively. To deal with class imbalances, proper class weights were determined using the compute_class_weight function from the Scikit-Learn library. During training, the learning rate, the number of training epochs, and the batch size was selected as 10^{-4}, 100, and 64, respectively. $ModelCheckpoint$ class was used to determine the best model based on validation scores.

4.2 Test Results

The proposed multiclass classification and binary classification models were examined with two test datasets. These datasets are: one is 20% of the training dataset and the other is the test dataset of NSL-KDD. The complexity matrix results obtained in the test results are presented in Tables 4 and 5. Figure 6 display evaluation results when the multiclass model was trained with the train part and tested using the test part. The proposed model estimated 20,231 of a total of 25,544 test samples correctly. Other samples in the test data

Table 3 Experimental hardware and software environment

Hardware	Features
Operating system	64 bit, Ubuntu 18.04
TensorFlow/Keras	2.3.1 / 2.4.3
CPU	AMD Ryzen 2700 Eight-CoreProcessor
RAM	16 GB
Video graphics card	NVIDIA GeForce®GTX1080

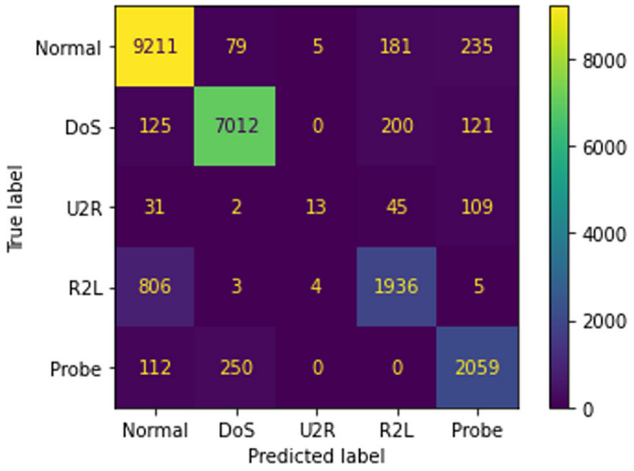

Fig. 6 Confusion matrix results on test data for multiclass classification

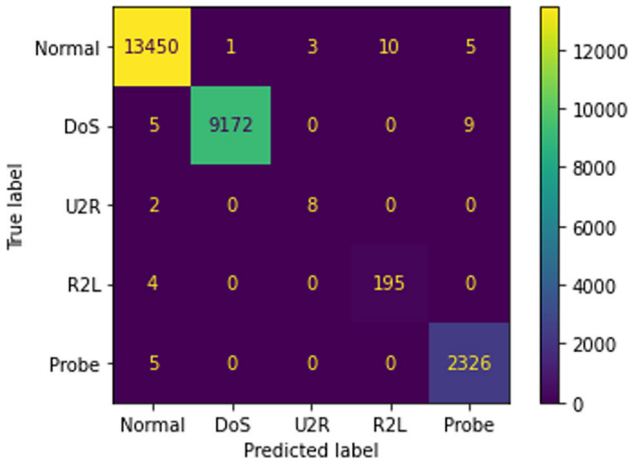

Fig. 7 Confusion matrix result on 20% of train data as test data for multiclass classification

set were classified as incorrect. Figure 7 displays the values of the confusion matrix for the seconds test case where 80% of the training part was allocated to train and the rest was used to test. For this test, the proposed IDS estimated 25,151 of 25,195 samples correctly and the rest of the samples were classified as incorrect. According to this result, a higher performance result was obtained compared to the first test case. Similar test cases were also repeated for the binary classification model.

Table 4 Confusion matrix result on test data for binary classification

	Normal	Attack
Normal	9,094	617
Attack	1,054	11,779

Table 5 Confusion matrix of 20% of train data as test data

	Normal	Attack
Normal	13449	20
Attack	18	11708

Table 6 Evaluation of results for multiclass classification using test data

	Recall, TPR	Specifity, TNR	Precision	F-Score
Normal	0.9485	0.9163	0.8956	0.9213
DoS	0.9402	0.9779	0.9545	0.9473
U2R	0.0650	0.9996	0.5909	0.1171
R2L	0.7030	0.9785	0.8196	0.7568
Probe	0.8505	0.9766	0.8142	0.8319

Table 7 Evaluation of results for multiclass classification using 20% of train data as test data

	Recall, TPR	Specifity, TNR	Precision	F-Score
Normal	0.9986	0.9986	0.9988	0.9987
DoS	0.9985	0.9999	0.9999	0.9992
U2R	0.8000	0.9999	0.7273	0.7619
R2L	0.9799	0.9996	0.9512	0.9653
Probe	0.9979	0.9994	0.9940	0.9959

Table 8 Evaluation of results for binary classification using test data

	Recall, TPR	Specifity, TNR	Precision	F-Score
Normal	0.8961	0.9179	0.8961	0.9159
Attack	0.9502	0.9365	0.9502	0.9338

Table 9 Evaluation of results for binary classification using 20% of train data as test data

	Recall, TPR	Specifity, TNR	Precision	F-Score
Normal	0.9985	0.9985	0.9987	0.9986
Attack	0.9985	0.9985	0.9983	0.9984

Table 10 The multiclass classification comparisons

References	Accuracy for test dataset	References	Accuracy for 20% of train dataset
Masum-2021 [48]	78.39	Xu-2018[49]	99.24
M. Al-Qatf-2018 [38]	80.48	Yin-2017 [16]	81.29
Yin-2017 [16]	81.29	Elmasry-2020 [33]	98.77
Li-2020 [2]	81.33	Shone-2018 [9]	85.42
Su-2020 [23]	84.25	Yan-2018 [50]	99.35
Kasongo-2021 [51]	88.42	Yang-2019 [39]	97.45
Base-line CNN model	82.85	Base-line CNN model	99.59
Our Model	**89.74**	Our Model	**99.83**

Table 11 The binary classification comparisons

References	Accuracy for test dataset	References	Accuracy for 20% of train dataset
Mushtaq-2022 [40]	89.00	Jiang-2019 [52]	98.94
Rani-2022 [41]	85.56	Abeshu-2018 [53]	99.20
M. Al-Qatf-2018 [38]	84.96	M. Al-Qatf-2018 [38]	99.41
Naseri-2022 [42]	83.00	Kasim-2020 [30]	99.50
Li-2020 [2]	86.95	Elmasry-2020 [33]	99.83
Masum-2021 [48]	89.30	Parampottupadam-2018 [19]	**99.91**
Base-line CNN model	84.67	Base-line CNN model	99.67
Our Model	**92.58**	Our Model	99.85

Tables 6, 7, 8 and 9 give an evaluation of the results using Recall, Specificity, Precision and F-score for both proposed models and both dataset cases where the test dataset and 20% of the training dataset are used for evaluation. Table 6 shows the test results obtained as a result of the operations performed on the test dataset using multiclass classification. When the results are examined, it is seen that the highest performance is obtained for TPR, normal, and DOS classes, and the lowest values for U2R. For TNR, it was observed that the highest value was in the U2R class and the lowest value was in the Normal class. While the best value for Precision and F-Score is obtained for the DOS attack class, it is seen that the worst value is for U2R.

Table 7 presents the test results for multiclass classification operations done with 20% of the training dataset. According to the test results, the highest performance was found for TPR, normal, DOS, and Probe classes, and the lowest value for U2R. It was observed that high performance was achieved in all classes for TNR. While the best value for Precision and F-Score is obtained for Normal, DOS, and Probe classes, it is seen that the worst value is for U2R. Tables 8 and 9 present the evaluation results for 20% of the test and training dataset for binary classification. When the results for the test data set in Table 8 are examined, it is seen that the results of the Attack class are higher in all performance tests. In the tests performed with 20% of the training data set, equal values were obtained for TPR and TNR in both classes, Precision and F-Score Normal class were higher in Table 9.

Table 12 Evaluations on Nvidia GTX 1080 GPU

Inference time (ms)	0.1403
Packages per second	7128.4

5 Conclusion

The performance of proposed multiclass and binary classification models have been given in detail Table 10 and Table 11 respectively. When the NSL-KDD test data set is compared with the studies on the multiclass classification model, Table 10 shows that the suggested model yields the best results, with an accuracy rate of 89.74%. Among the studies performed with a 20% training dataset, our model achieved the best work of 99.83%. When compared to the literature, the overall evaluations of the NSL-KDD utilizing the suggested model for the test dataset were better. Also, the results for tests using 20% of the training dataset show that our model performs best for multiclass classification. Table 11 shows the results of the comparison of the binary classification results obtained with the studies in the literature. In the tests performed using the test data set, our model achieved an accuracy value of 92.58%, and when compared with the

studies in the literature, it is seen that this result outperforms the other studies compared. In the tests performed with 20% of the training data set, our model has an accuracy value of 99.85%. When compared with the studies in the literature, it was found to be very close to the best result of 99.91%. Our final evaluation presents the inference time and the number of packages per second on Nvidia GTX 1080 GPU as shown by Table 12. These results can significantly be improved further by using more powerful and up-to-date GPU cards.

IDS is critical in protecting computer networks from harmful intrusions. Nowadays, various researchers study Deep Learning-based IDS to achieve better results. We developed a novel IDS model based on CNN in this work. The packet features are normalized and reshaped into 2D to use transfer learning. Except for the two layers, we transferred the weights of the two-dimensional convolutional layers from a pre-trained network. Using the models trained from scratch usually limits the test dataset results to about 82%. Training data does not provide enough generality because of the content of test dataset. Hence, a better alternative is to transfer the weights from a previously trained model, which usually helps extract some features that may not be learned from limited data. For this purpose, we converted the input into image data and incorporated the VGG16 model, a CNN-based pre-trained network trained on a more extensive data set. Given the experimental results, one of the important reasons for the success of the proposed method among the examined literature studies is the proposed deep learning model that partially contains pre-trained model. In addition, the conversion of the traffic into images, one-hot encoding and logarithmic normalization operations increase the performance. According to the results for multiclass and binary evaluation, the proposed model outperforms current studies in intrusion detection accuracy on the test dataset. Moreover, the results for multiclass model for 20% of the train dataset, which is usually preferred for comparison in the literature, provided good results. We plan to develop better transfer learning models with improved architectures for better detection rates for future studies. In addition, the proposed preprocessing techniques will be improved to increase IDS performance, and different machine learning techniques will be tested by examining different datasets.

Acknowledgements This work was not supported by any institution or organization.

Data Availibility The NSL-KDD dataset is accessible at: https://www.unb.ca/cic/datasets/nsl.html.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethical Approval This article does not contain any studies with animals performed by any of the authors.

Informed Consent Informed Consent was obtained from all individual participants included in the study.

References

1. Zhang, X.; Chen, J.; Zhou, Y.; Han, L.; Lin, J.: A multiple-layer representation learning model for network-based attack detection. IEEE Access **7**, 91992 (2019)
2. Li, Y.; Xu, Y.; Liu, Z.; Hou, H.; Zheng, Y.; Xin, Y.; Zhao, Y.; Cui, L.: Robust detection for network intrusion of industrial IoT based on multi-CNN fusion. Measurement **154**, 107450 (2020)
3. Roy, B.; Cheung, H.: A deep learning approach for intrusion detection in internet of things using bi-directional long short-term memory recurrent neural network. In: *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)* (IEEE, 2018), pp. 1–6
4. Lin, S.Z.; Shi, Y.; Xue, Z.: Character-level intrusion detection based on convolutional neural networks. In: *2018 International Joint Conference on Neural Networks (IJCNN)* (IEEE, 2018), pp. 1–8
5. Zhang, Y.; Zhang, H.; Zhang, X.; Qi, D.: Deep learning intrusion detection model based on optimized imbalanced network data. In: *2018 IEEE 18th International Conference on Communication Technology (ICCT)* IEEE, pp. 1128–1132 (2018)
6. Naseer, S.; Saleem, Y.; Khalid, S.; Bashir, M.K.; Han, J.; Iqbal, M.M.; Han, K.: Enhanced network anomaly detection based on deep neural networks. IEEE Access **6**, 48231 (2018)
7. Xin, Y.; Kong, L.; Liu, Z.; Chen, Y.; Li, Y.; Zhu, H.; Gao, M.; Hou, H.; Wang, C.: Machine learning and deep learning methods for cybersecurity. IEEE Access **6**, 35365 (2018)
8. Manimaran, A.; Chandramohan, D.; Shrinivas, S.; Arulkumar, N.: A comprehensive novel model for network speech anomaly detection system using deep learning approach. Int. J. Speech Technol. **23**(2), 305 (2020)
9. Shone, N.; Ngoc, T.N.; Phai, V.D.; Shi, Q.: A deep learning approach to network intrusion detection. IEEE Trans. Emerg. Topic. Comput. Intell. **2**(1), 41 (2018)
10. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A.: *2009 IEEE Symposium On Computational Intelligence for Security and Defense Applications* IEEE, pp. 1–6 (2009)
11. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A.: Tavallaee, Mahbod and Bagheri, Ebrahim and Lu, Wei and Ghorbani, Ali A. In: *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications* , pp. 1–6 (2009)
12. Jo, W.; Kim, S.; Lee, C.; Shon, T.: Packet preprocessing in CNN-based network intrusion detection system. Electronics **9**(7), 1151 (2020)
13. Lopez-Martin, M.; Carro, B.; Sanchez-Esguevillas, A.: Application of deep reinforcement learning to intrusion detection for supervised problems. Expert Syst. Appl. **141**, 112963 (2020)
14. Kasongo, S.M.; Sun, Y.: A deep learning method with filter based feature engineering for wireless intrusion detection system. IEEE Access **7**, 38597 (2019)
15. Kasongo, S.M.; Sun, Y.: A deep long short-term memory based classifier for wireless intrusion detection system. ICT Express **6**(2), 98 (2020)
16. Yin, C.; Zhu, Y.; Fei, J.; He, X.: A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access **5**, 21954 (2017)
17. Zhiqiang, L.; Mohi-Ud-Din, G.; Bing, L.; Jianchao, L.; Ye, Z.; Zhijun, L.: Modeling network intrusion detection system using feed-forward neural network using UNSW-NB15 Dataset. In: *2019*

IEEE 7th International Conference on Smart Energy Grid Engineering (SEGE) IEEE, pp. 299–303 (2019)
18. Wang, Z.: Deep learning-based intrusion detection with adversaries. IEEE Access **6**, 38367 (2018)
19. Parampottupadam, S.; Moldovann, A.N.: Cloud-based real-time network intrusion detection using deep learning. In: *2018 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)* IEEE, pp. 1–8 (2018)
20. Vinayakumar, R.; Alazab, M.; Soman, K.; Poornachandran, P.; Al-Nemrat, A.; Venkatraman, S.: Deep learning approach for intelligent intrusion detection system. IEEE Access **7**, 41525 (2019)
21. Zhang, C.; Ruan, F.; Yin, L.; Chen, X.; Zhai, L.; Liu, F.: A deep learning approach for network intrusion detection based on NSL-KDD dataset. In: *2019 IEEE 13th International Conference on Anti-counterfeiting, Security, and Identification (ASID)* IEEE, pp. 41–45 (2019)
22. Zhang, Y.; Li, P.; Wang, X.: Intrusion detection for IoT based on improved genetic algorithm and deep belief network. IEEE Access **7**, 31711 (2019)
23. Su, T.; Sun, H.; Zhu, J.; Wang, S.; Li, Y.: BAT: deep learning methods on network intrusion detection using NSL-KDD dataset. IEEE Access **8**, 29575 (2020)
24. Zhang, J.; Li, F.; Ye, F.: An ensemble-based network intrusion detection scheme with bayesian deep learning. In: *ICC 2020-2020 IEEE International Conference on Communications (ICC)* IEEE, pp. 1–6 (2020)
25. Ujjan, R.M.A.; Pervez, Z.; Dahal, K.; Bashir, A.K.; Mumtaz, R.; González, J.: Towards sFlow and adaptive polling sampling for deep learning based DDoS detection in SDN. Futur. Gener. Comput. Syst. **111**, 763 (2020)
26. Hassan, M.M.; Gumaei, A.; Alsanad, A.; Alrubaian, M.; Fortino, G.: A hybrid deep learning model for efficient intrusion detection in big data environment. Inf. Sci. **513**, 386 (2020)
27. Parra, G.D.L.T.; Rad, P.; Choo, K.K.R.; Beebe, N.: Detecting Internet of Things attacks using distributed deep learning. J. Netw. Comput. Appl. **163**, 102662 (2020)
28. Gamage, S.; Samarabandu, J.: Deep learning methods in network intrusion detection: a survey and an objective comparison. J. Netw. Comput. Appl. **169**, 102767 (2020)
29. Priyadarshini, R.; Barik, R.K.: A deep learning based intelligent framework to mitigate DDoS attack in fog environment, J. King Saud Univ.-Comput. Inform. Sci. (2019)
30. Kasim, O.: An efficient and robust deep learning based network anomaly detection against distributed denial of service attacks. Comput. Netw. **180**, 107390 (2020)
31. Çavuşoğlu, Ü.: A new hybrid approach for intrusion detection using machine learning methods. Appl. Intell. **49**(7), 2735 (2019)
32. Ferrag, M.A.; Maglaras, L.; Moschoyiannis, S.; Janicke, H.: Deep learning for cyber security intrusion detection: approaches, datasets, and comparative study. J. Inform. Secur. Appl. **50**, 102419 (2020)
33. Elmasry, W.; Akbulut, A.; Zaim, A.H.: Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic. Comput. Netw. **168**, 107042 (2020)
34. Kasongo, S.M.; Sun, Y.: A deep learning method with wrapper based feature extraction for wireless intrusion detection system. Comput. Secur. **92**, 101752 (2020)
35. Zhang, J.; Li, F.; Zhang, H.; Li, R.; Li, Y.: Intrusion detection system using deep learning for in-vehicle security. Ad Hoc Netw. **95**, 101974 (2019)
36. Binbusayyis, A.; Vaiyapuri, T.: Unsupervised deep learning approach for network intrusion detection combining convolutional autoencoder and one-class SVM, Applied Intelligence pp. 1–15 (2021)
37. Tian, Q.; Han, D.; Li, K.C.; Liu, X.; Duan, L.; Castiglione, A.: An intrusion detection approach based on improved deep belief network. Appl. Intell. **50**, 3162 (2020)
38. Al-Qatf, M.; Lasheng, Y.; Al-Habib, M.; Al-Sabahi, K.: Deep learning approach combining sparse autoencoder with SVM for network intrusion detection. IEEE Access **6**, 52843 (2018)
39. Yang, H.; Qin, G.; Ye, L.: Combined wireless network intrusion detection model based on deep learning. IEEE Access **7**, 82624 (2019)
40. Mushtaq, E.; Zameer, A.; Umer, M.; Abbasi, A.A.: A two-stage intrusion detection system with auto-encoder and LSTMs. Appl. Soft Comput. **121**, 108768 (2022)
41. Rani, M.; et al.: Effective network intrusion detection by addressing class imbalance with deep neural networks multimedia tools and applications. Multimedia Tools Appl. **81**(6), 8499 (2022)
42. Naseri, T.S.; Gharehchopogh, F.S.: A feature selection based on the farmland fertility algorithm for improved intrusion detection systems. J. Netw. Syst. Manage. **30**(3), 1 (2022)
43. Ding, Q.; Li, J.: AnoGLA: an efficient scheme to improve network anomaly detection. J. Inform. Secur. Appl. **66**, 103149 (2022)
44. Ahmad, R.; Alsmadi, I.; Alhamdani, W.; Tawalbeh, L.: A deep learning ensemble approach to detecting unknown network attacks. J. Inform. Secur. Appl **67**, 103196 (2022)
45. Simonyan, K.; Zisserman, A.: arXiv preprint arXiv:1409.1556 (2014)
46. He, K.; Zhang, X.; Ren, S.; Sun, J.: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778 (2016)
47. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z.: *Proceedings of the IEEE conference on computer vision and pattern recognition* , pp. 2818–2826 (2016)
48. Masum, M.; Shahriar, H.; Haddad, H.M.: A transfer learning with deep neural network approach for network intrusion detection. Int. J. Intell. Comput. Res. (2021)
49. Xu, C.; Shen, J.; Du, X.; Zhang, F.: An intrusion detection system using a deep neural network with gated recurrent units. IEEE Access **6**, 48697 (2018)
50. Yan, B.; Han, G.: Effective feature extraction via stacked sparse autoencoder to improve intrusion detection system. IEEE Access **6**, 41238 (2018)
51. Kasongo, S.M.; Sun, Y.: A deep gated recurrent unit based model for wireless intrusion detection system. ICT Express **7**(1), 81 (2021)
52. Jiang, F.; Fu, Y.; Gupta, B.B.; Liang, Y.; Rho, S.; Lou, F.; Meng, F.; Tian, Z.: Deep learning based multi-channel intelligent attack detection for data security. IEEE Trans. Sustain. Comput. **5**(2), 204 (2018)
53. Abeshu, A.; Chilamkurti, N.: Deep learning: the frontier for distributed attack detection in fog-to-things computing. IEEE Commun. Mag. **56**(2), 169 (2018)