Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2021) 29: 1929 – 1943
© TÜBİTAK
doi:10.3906/elk-2005-1

A novel approach for intrusion detection systems: V-IDS

Kenan İNCE*
Department of Computer Engineering, Engineering Faculty, İnönü University, Malatya, Turkey

Received: 01.05.2020 • **Accepted/Published Online:** 10.08.2020 • **Final Version:** 26.07.2021

Abstract: An intrusion detection system (IDS) is a security mechanism that detects abnormal activities in a network. An ideal IDS must detect intrusion attempts and maybe categorize them for further research and keep false-positive analysis at a very low level. IDSs are used in the analysis of network traffic data at all sizes. Studies on this subject focused on machine learning techniques. Even though the performance rates are high, it is seen that processes such as data understanding, preprocessing, and consistency tests are time-consuming and laborious. For this reason, the use of deep learning (DL) models that automatically perform the mentioned steps has become very popular. In this study, a high-performance approach that can be applied in real-time systems is proposed: visual IDS (V-IDS). NSL-KDD dataset, one of the large-scale datasets, is used. Data visualization techniques were applied in order to determine geometric relationships between records, and the data were classified by using the DL model. The model achieved 98% accuracy in total and even higher in some intrusion categories.

Key words: Data visualization, deep learning, intrusion detection, network security

1. Introduction

Today the internet has entered every stage of our lives. As a result, many new fields of study have emerged. Many of these are solutions that make life easier for people. However, some have become initiatives that threaten all the values of humanity.

Actors threatening information systems are called hackers. In fact, the word hacking means using a product or procedure for another purpose. The use of a straw as a cable organizer is an example of hacking in our daily lives. Information attacks are uses outside the purpose of the internet, which was developed for the purpose of communication. For this reason, it is the main goal of intrusion detection is to distinguish between information attacks as normal and abnormal. Intrusion detection studies can be summarized as preventing abnormal activities without obstructing normal activities.

We can divide abnormal activities into two main categories: active and passive. Passive attacks aim to collect information without damaging the system and leaving any trace, but they are the types of attacks that have the potential to turn into active attacks one day. For example, probe attack is in this category. On the other hand, active attacks are somehow damaging attacks by performing manipulations on the systems it accesses. Moreover, active attacks can be examined in two subcategories: those that make the systems unusable and those that aim to gain any kind of benefit by gaining access.

Unauthorized access is a serious problem in network security. A single intrusion may cause server and network systems to crash within seconds, and they bring about huge problems such as data inconsistency,

*Correspondence: kenanince@gmail.com

manipulation, and deletion. These attacks may not be limited to software or information damage, they may also damage hardware or even cause huge financial losses to organizations. One of the most recent cybercrimes is ransomware which is designed for encrypting valuable files and make them unusable by owner unless they pay the requested ransom [1]. IDSs are tools that can help prevent these types of attacks at lower levels. Attacks can be classified into five phases: exploration, exploitation, reinforcement, consolidation, and looting [2]. An attack must be detected in the first three phases, otherwise it will probably be too late.

IDS development studies can be examined under two main titles, which are host-based (HIDS) and network-based (NIDS) [3]. HIDS is capable of monitoring and analyzing the network traffic on its network interfaces. On the other hand, NIDS analyzes the passing traffic on the entire subnet for known attacks. In fact, HIDS should catch anything that passes the NIDS. NSL-KDD dataset is one of the NIDS datasets. It contains four types of attacks, which are DoS, Probe, U2R, and R2L. Moreover, it is still the most commonly used dataset in IDS studies as shown in [4].

In the literature, machine learning (ML) techniques are generally used for intrusion detection systems and network analysis operations. Artificial neural networks [5], genetic algorithms [6], Bayesian networks [7], and decision trees [8] are examples of the machine learning methods used. In addition, many review studies were conducted comparing multiple methods [9, 10].

The biggest disadvantage of ML methods is wrong classification. The development of machine learning models without an understandable approach to analyze the causes of poor performance is often based on the trial-and-error process. Deep learning (DL) has made important stride on challenging issues with ML techniques such as image classification, voice and handwriting recognition, text-to-speech conversion, digital assistants, and infrastructure for autonomous vehicles.

DL became superior over ML by image processing capabilities. Data visualization (DV) is focused on creating images from the dataset according to point of interest [11]. DV plays a crucial role especially in large and complex datasets. DV is used in many research areas like data mining [12], traffic data [13], forest file visualization [14]. However, no study combining these two disciplines for the IDS subject has been found.

In this study, it is aimed to understand the data and reveal the geometric relationship between the data by applying a visualization process to monitor the network traffic, and the attack is attempted to be detected in real time using DL techniques. The major contribution of this paper is using DL method with visualized NSL-KDD dataset.

1.1. Related work

ML can be classified as a subset of artificial intelligence which is study of algorithms and scientific models used by computer systems. Basically, ML is the operation of building a mathematical model based on training data and making predictions or take decisions [15]. Decision trees, K-means clustering, support vector machines (SVM), K-nearest neighbors, convolutional neural network (CNN), Naïve Bayes, and regression are some of the major algorithms in ML.

Machine learning methods have been used in IDS studies frequently since the most basic feature of IDS is the classification of new data by modeling past data. However, with the development of DL methods, it started to be used intensively in IDS studies.

Liu et al. proposed a visual analysis system that makes it easier to understand, diagnose, and develop CNN, i.e. deep convolutional neural networks. They compared classical machine learning methods and deep learning methods and categorized literature studies. While performing these studies, DARPA98, KDD99, UNSW-NB15 datasets, which are used intensively in intrusion detection, were used [16].

Rauber et al. conducted studies to visualize the relationships between the learned representations of observations and to visualize the relationships between the artificial neuron to give network designers an idea of how their systems work. While doing this process, the method of size reduction was proposed and it was written to visualize the relationships between the learned impressions of the observations. As a result of the experiments, it was written that the visualization can provide highly useful feedback for the network designers. The classification of the data by signal preprocessing is emphasized [17].

Fiore et al.'s study is an important publication in the literature where large network traffic is used in cyber security studies. They adopted a semicontrolled anomaly detection system where the classifier is trained with the normal traffic data in the system they call Boltzmann machine. Thus, the way of having dynamic information about abnormal behaviors has been opened. They extracted strong models from the limited Boltzmann model and developed it in order to combine the impressive power of the model with the accuracy capabilities of classification [18].

Gao et al. suggested a method they call SOEKS for in-car safety systems with DL and experience knowledge structure. The safety of vehicle information systems gained extra importance as the need for security increased due to the widespread use of smart vehicles. With this method, an in-vehicle intrusion detection system was developed, and information entropy was used to increase intrusion detection for different vehicles. In practice, a definite model is proposed for all vehicles by training a large specific vehicle dataset with DL method. The results showed that the system had a sensitivity of 98% and could detect a wide range of in-vehicle attacks [19].

Chakravarthi et al. proposed to review the role of DL techniques for IDS and an effective deep autocoder (AE)-based intrusion detection technique. Intrusion detection was carried out in two stages: binary and multiclass classification. In addition, comparisons were made with existing parallel methods. The restructured error of the AE model is compared with the PCA method. The compressed representation of the AE model was classified by SVM and dense NN, respectively. An attempt was made to reduce the false alarm rate [20].

Despite the improvements made, Liu and Lang proposed an IDS taxonomy, which takes objects, as the main dimension of data, to classify and summarize the IDS literature to eliminate the disadvantages of existing IDSs to improve detection accuracy and attack detection. In the study, ML algorithms used in IDS, metrics, and comparison datasets are introduced. Then, the proposed method is explained. Finally, representative studies and future developments are discussed [21].

Alessa et al. analyzed the research environment according to a consistent taxonomy for IDS based on DL techniques. It is an article about DL and attack keywords and their four main database variants such as Web Science, Science Direct, Scopus IEEE Xplore. The dataset consists of 68 articles. It discusses models to run and adopt IDS. Then the result analysis match for new searches is discussed. This study is a comparison for those who are interested in IDS and reveals the advantages and disadvantages of the methods [22].

Lee and Park proposed a high-performance NIDS based on DL for their normal and abnormal classification. They used AE and GAN models to reveal data imbalance and perform high performance analysis. The automatic encoder conditional GAN, which they call AE-CGAN, was claimed to improve the performance of intrusion detection. Canadian Institute CICS-2017 database was used while developing the proposed method. Random forest was used to evaluate the classification performance of the AE-CGAN model. As a result of the classification, the highest performance was achieved with AE-CGAN and a 95% success rate in attack detection was achieved [23].

In their work, Wang et al. explored the concept of cyber security and exposed big data features and diversity in network traffic and attacks. One of the clustering methods, k-means clustering algorithm was used, and KDD-Cup99 and MAWILab were used as dataset [24].

Martin et al. proposed a new application of several deep reinforcement learning (DRL) algorithms in interference detection using tagged datasets to improve the performance of IDS. This proposed algorithm was implemented in NSL-KDD and AWID datasets. The application results of DRL models were obtained using deep Q network (DQN), double deep Q Network (DDQN), policy gradient (PG), and actor critical (AC), and DDQN yielded the best result [25].

2. Theoretical background

Since the advent of artificial intelligence, researchers have aimed for more cleaver algorithms. Although artificial intelligence has continuously improved from past to present, the concept of deep learning has emerged as a new perspective with high performance results in the Large-Scale Visual Recognition (ImageNet) competition in 2012 [26]. This improvement was revolutionary in ML because the performance result obtained was far better than those obtained with traditional techniques. Russakovsky et al. achieved 37.5% in top-1 and 17.0% in top-5, which was better than previous studies [27]. The main advantages of DL are the use of as much data as possible in training stage and hardware resource improvement in recent years, especially for GPUs.

DL has facilitated the solution of problems, as it has fully automated the basic phase of ML, called feature extraction. The convolution is the operation of combining two signals to form a third signal. The first signal is the data itself and the second signal is the filter.

Another point that draws attention about DL is its simplicity. The basis of this simplicity is that derivation in artificial neural networks (NN) does not exist in DL. However, the hard part of deep learning is that it is trained with too many samples. By advances in multicore PCs and especially in GPUs, DL gained velocity because training time with too many samples reduced dramatically. In this manner NVIDIA GPUs are the best choice by means of platform support and processor core counts.

Convolutional neural networks (CNN) are one of the most widely used methods for image recognition and classification [28]. CNN basically extracts the features of the images from input layer and makes a fully connected classification [29]. A CNN architecture basically consists of convolution, pooling, activation, and fully connected classification layers. The images become abstracted to feature map with predefined sizes in the convolution layer and the result is passed to the next layer. The pooling layer reduces the size. In the activation layer, nonlinear properties of the network are reduced by applying some functions. Most popular activation layers are rectified linear unit (ReLu) and dropout. Finally, in the fully connected layer, all the data are combined with their relative weights.

DL can be a CNN or a multilayer perceptron. In this section, the DL layers are briefly presented from the CNN perspective, as follows.

2.1. Layers of DL

In this section, as a summary point of view, seven of major DL layers will be presented.

2.1.1. Input layer

The first layer of DL is the input layer which inputs the raw data. The most important factor in this layer is the size of the data. Big data size requires more hardware resources as well as increasing the required processing time. Small data size, on the other hand, may yield low performance.

2.1.2. Convolution layer

In mathematics, convolution means a mathematical operation on two functions for producing a third function which expresses how the shape of one modified by other. The modifying functions serve the purpose of transformation. In DL, this function is named as kernel and different kernels may be selected depending on the study. The general equation of convolution process is shown in equation 1.

$$f(x) * g(x) = (f * g)(x) = \int_{-\infty}^{\infty} f(\tau)g(x-\tau)d\tau \tag{1}$$

Convolution layer forms the basis of DL. This layer, also known as the transformation layer, moves the convolution window, named kernel, over the entire image and creates a new matrix. By changing the factor of convolution windows in every step, the distinctive features of the images are extracted. In Figure 1, this operation is presented symbolically.

In summary, in this layer, unlike conventional artificial neural networks, feature extraction is performed. The convolution process pulls smaller parts, that is, smaller than the attribute map, and creates the output attribute map by applying the same transformation to all of them. In other words, it can be said that the process of obtaining an output attribute map from the input attribute map is the convolution process. Filters encode high level objects or objects. For example, a filter can encode whether or not an image has face information.

2.1.3. Rectified linear unit (ReLu) layer

The activation function is responsible for transforming an input signal to a different output signal. The output signal is the input of next layer. Simply, ReLu is linear for all positive inputs and zero for all negative inputs as shown in Figure 2. ReLu is the most commonly used activation function especially in CNNs. Simple definition of ReLu is presented in equation 2

$$f(x) = max(0, x) \tag{2}$$

2.1.4. Pooling layer

In the CNN architecture, the purpose of the pooling layer is reducing the size. Generally, it is placed between convolution layers for this purpose.

By reducing the size, pooling layers help network avoid overfitting. Moreover, it reduces calculation cost off course. An example of max pooling is presented in Figure 3.

2.1.5. Dropout layer

The function of the dropout layer is to ignore some neurons during training face with a probability. At each training stage, every node will be kept or dropout according to the predefined probability. This operation prevents network overfitting problem.

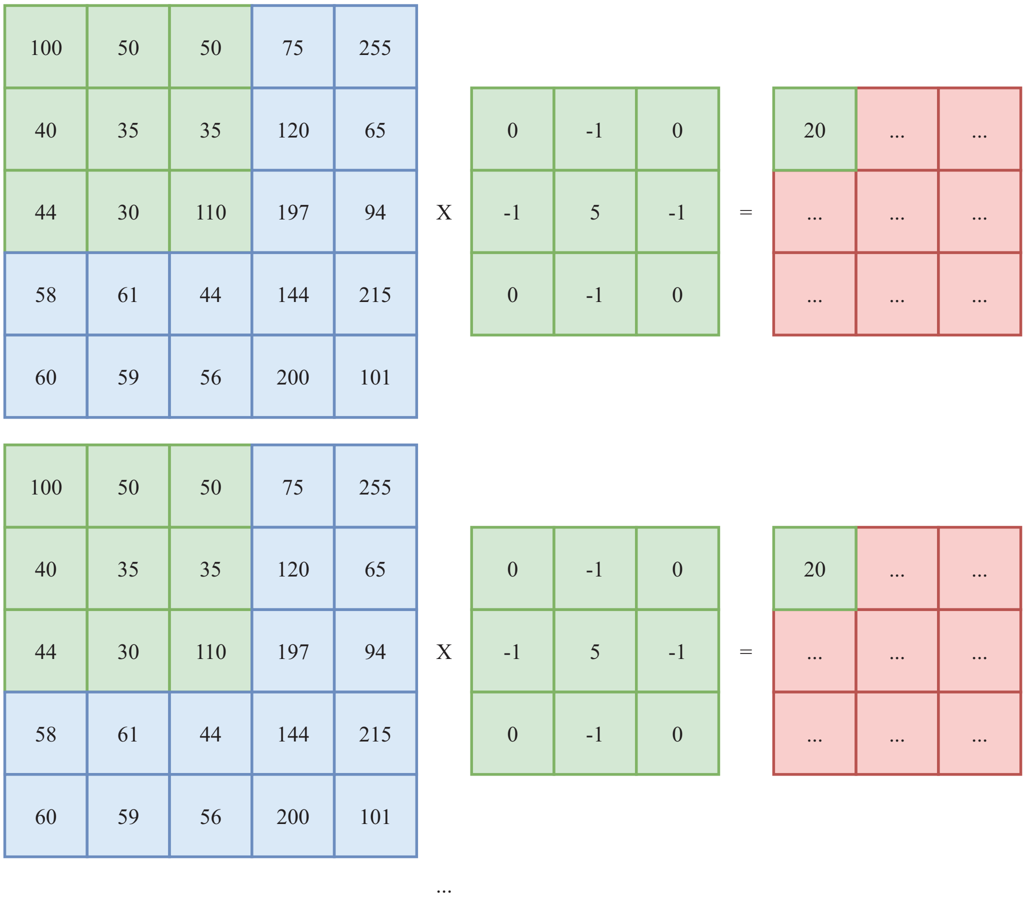


Figure 1. A symbolic convolution operation with 2 steps.

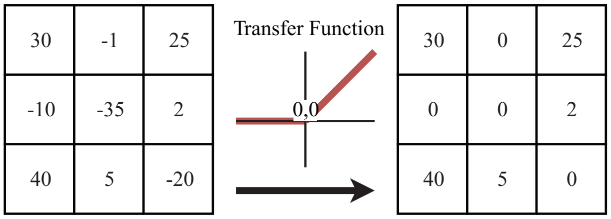


Figure 2. Rectified linear unit function.

2.1.6. Fully connected layer

The fully connected layer is the actual phase of DL architecture that distinguishes the classes. It has connection to all the neurons in the previous layer, as seen in regular neural network. As a result, the weights of these neurons can be computed with a matrix multiplication. It is used to optimize objective like classification scores.

2.1.7. Classification layer

This layer comes after the fully connected layer. The output of this layer is equal to the number of input classes. Since we have two objects in the attack detection system as abnormal and normal, we have two classes. Different

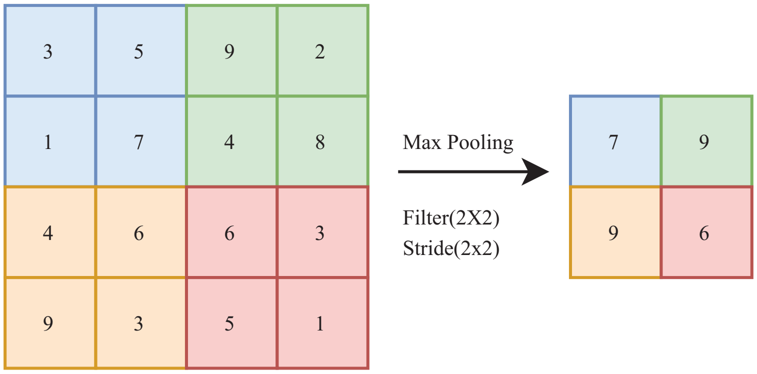


Figure 3. Max pooling with 2 x 2 filter and stride.

classifiers can be used in this layer; however, the most commonly used classifier is Softmax. Also sigmoid and tangent nonlinear functions are used. The mathematical formulas for these functions are given below.

$$sigmoid = \frac{1}{1+e^{-x}} \tag{3}$$

$$tangent = \frac{e^{2x}-1}{e^{2x}+1} \tag{4}$$

$$softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{5}$$

3. Application

3.1. Dataset

Numerous IDS datasets are used in the literature. However, maybe the most widely used datasets are DARPA, KDD'99, and NSL-KDD. NSL-KDD dataset is a new version suggested to solve some problems in KDD dataset. The NSL-KDD dataset is actually a more up-to-date dataset obtained from the KDD99 dataset by clearing the same records and reducing data using the feature selection method [30].

NSL-KDD dataset classes and record numbers are presented in Table 1. NSL-KDD dataset consists of five classes which are Normal, DoS, Probe, R2L, and U2R. NSL-KDD dataset contains four types of features; categorical, binary, discrete, and continuous. There are 4 categorical, 6 binary, 23 discrete, and 10 continuous features.

Table 1. NSL-KDD dataset record counts.

Dataset partition	Number of records					
	Normal	DoS	Probe	U2R	R2L	Total
KDDTest-21	2154	4342	2402	67	2885	11,850
KDDTest	9713	7458	2421	67	2885	22,544
KDDTrain	67,343	45,927	11,656	52	995	125,973
KDDTrain-20	13,449	9234	2289	11	209	25,192

Denial of service (DoS) type of attacks aim to prevent targeted system from serving. Neptune, mailbomb, pod, smurf, teardrop etc. are some of the DoS attacks in the dataset.

Probe attacks aim to obtain the necessary information to access valid IP addresses, active ports, or systems of a server or any machine. Satan, ipsweep, portsweep, nmap, and mscan are some probe attacks that take place in the dataset.

Remote to local (R2L) is the unauthorized access of the user to the computer system, such as a guest or another user. Phf, Spy, Imap, ftp_write, multihop, Guess_passwd, Warezmaster, and Warezclient are some of the R2L attack labels in the dataset.

User to root (U2R)) type of attack is when an unauthorized user tries to perform operations that require administrative permission even though they do not have administrative user privileges. Rootkit, Perl, loadmodule, and Buffer-overflow are some of the U2R attack types in the dataset.

3.2. Data visualization

The algorithm used in this study is presented in a simplified manner in Algorithm 1. The normalization formula is shown in Equation 6.

$$Norm(X_i) = \frac{x_i}{x_{max}} * 255 \tag{6}$$

The NSL-KDD contains four text files which are named according to dataset partitions that are presented in Table 1. Algorithm 1 called for each file and generated image files placed in a directory named according to dataset partition.

Algorithm 1 Algorithm

```
procedure ParseDataFile($a$)                          ▷ a is the path of file
    System Initialization
    Read the file $a$
    while $a$ has next line do
        Normalize all values                          ▷ 0-255
        Create 6x7 matrix
        Save matrix as gray scale image
    Classify generated images
```

Figure 4 shows some samples of visualized record images. Matrices have two dimensions which are rows and columns. 3D tensors are acquired by combining matrices as arrays. Tensors have three main features which are degree, shape, and data type. For example, as a result of visualizing, records of KDDtest-21 will form (11850,6,7) tupple which means total number of 11850 6 x 7 matrix records. The major differences between DL and ML first appeared in image classification with the idea of it being very close to human eye recognition capabilities. Thus, the motivation of this study can be seen clearly in Figure 4.

3.3. Statistical measures

There exist different metric calculations for presenting performance of an algorithm or techniques. In classification problems, there are five important metrics which are accuracy (AC), recall (RC), precision (PR), false-positive rate (FPR), and F1-Score (FS). Accuracy is the proportion of truly classified records to total number of records. The formulation of AC presented in Equation 7. Accuracy is a good comparing metric

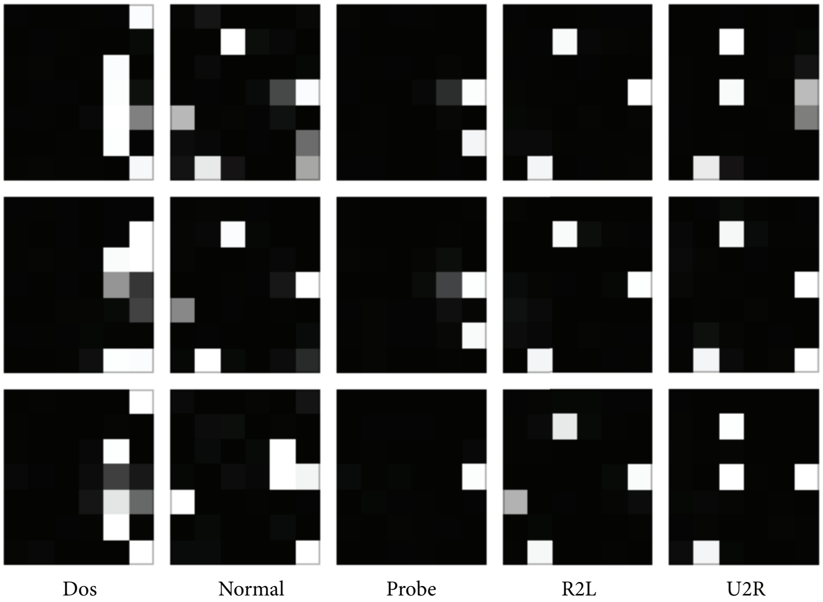


Figure 4. Some sample results of the data visualization process.

because the main goal of the classification problem is to achieve true classification as much as possible. This is directly proportional to results of classification algorithm which are true-possitive (TP), true-negative (TN), false-possitive (FP), and false-negative (FN). AC metric is the combination of these four results. AC metric is generally represented in percentage format. However, some studies normalize it and present it in decimal number between 0 and 1.

$$AC = \frac{TP+TN}{TP+FN+FP+TN} \tag{7}$$

Classifications problems like IDS aim to detect attack requests. RC metric is the value of caught attacks to total attacks. Thus, this metric is the proportion of true-positive records to the total of true-positive and false-negative. The formula of RC is presented in Equation 8. RC is also known as true-positive rate (TPR) or decision rate (DR).

$$RC = \frac{TP}{TP+FN} \tag{8}$$

PR metric can be said to be the inverse of RC which means how many of the records we classified as positive is really positive. This metric is an important one because in IDS, also the FP rate of an algorithm means the rate of some legitimate requests that are rejected, which is not a wanted situation. The calculation of PR is presented in Equation 9.

$$RR = \frac{TP}{TP+FP} \tag{9}$$

FPR estimates the ratio of the intrusions to total number of records. Lower FPR is better. FPR is defined in Equation 10.

$$FPR = \frac{FP}{FP+TN} \tag{10}$$

FS is a metric that contains information from PR and RC. After the calculation of these metrics, the FS metric can be calculated. The formula of FS is presented in Equation 11.

$$FS = 2 * \frac{PR * RC}{PR + RC} \tag{11}$$

3.4. Experiment environment

All the experiments were implemented on Windows 10 operating system (OS) using MATLAB with GPU enabled. The hardware specifications of environment are i9-9900K @3.6 GHz CPU, 32 GB DDR4 3200 Mhz RAM and Nvidia RTX-2080Ti GPU.

3.5. Experimental results

In this study two DL architectures were used: AlexNet CNN [31] and a custom 5-layer DNN architecture which was inspired by [2]. By these architectures, it is intended to examine how V-IDS approach will behave with approved models.

3.5.1. Multiclass Alexnet CNN results

Alexnet is one of the most popular DL models after Large Scale Visual Recognition Challenge 2012 [26]. It contains 8 learned (5 convolution and three fully connected), 7 ReLu, 2 normalization, 2 dropout, 3 pooling, 1 softmax, and 1 classification layers. With image input layer, AlexNet contains 25 layers in total. However, there are studies that modified AlexNet CNN and proposed these modified architectures [32, 33]. In this study, AlexNet CNN was used without any modification.

Because AlexNet CNN contains more layers and in every learned layer it contains more neurons, its running time is longer than custom architecture. Training times in minutes are presented in Table 2.

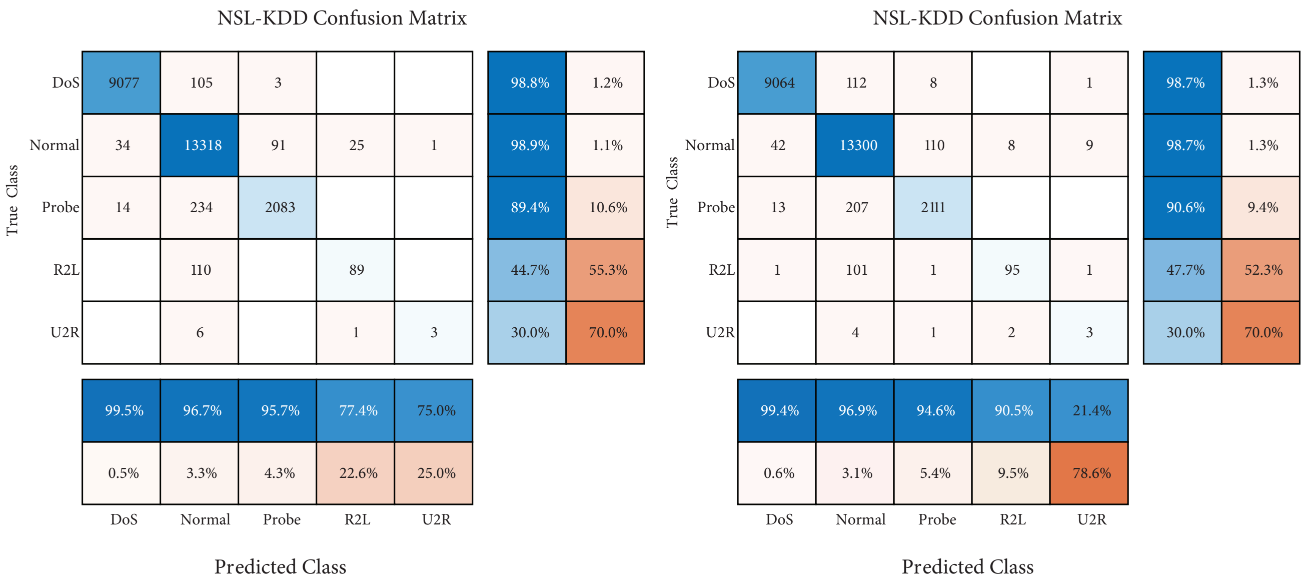
The experiments applied all dataset partitions separately. Table 3 presents all AlexNet CNN training results and Figure 5a shows the training confusion matrix.

Table 2. AlexNet CNN and custom CNN training times in minutes.

Dataset partition	AlexNet CNN	Custom CNN
KDDTest-21	28 min	9 min
KDDTest	80 min	29 min
KDDTrain	10461 min	709 min
KDDTrain-20	124 min	36 min

Table 3. AlexNet CNN NSL-KDD dataset partitions performances.

Dataset partition	Normal	DoS	Probe	R2L	U2R	Total
KDDTest-21	85.7	94.7	95.8	85.3	66.7	91.01
KDDTest	91.3	98.1	90.7	79.1	71.4	91.68
KDDTrain	96.7	99.5	95.7	77.4	75.0	97.52
KDDTrain-20	96.0	98.6	88.0	66.7	-	96.13



(a) AlexNet CNN Training Confusion Matrix

(b) Custom DNN Training Confusion Matrix

Figure 5. Confusion matrices of DL architectures used in training.

In [4], it was concluded that the overall performance was not promising for any of the tested configurations, as they did not exceed 80% accuracy in the best scenario with the NSL-KDD dataset. However, with V-IDS approach, AlexNet CNN without any modification yielded 97.52% accuracy.

3.5.2. Multiclass custom DNN model results

In [2], it was shown that deep neural network (DNN) performs better than classical ML techniques. It was also mentioned that the proposed framework is capable of detecting cyberattacks in real time. This framework was utilized only for arranging the output shape, which was diminished by halving from 4096 to 256. Training times are presented in Table 2 and performance values are presented in Table 4. Furthermore, the confusion matrix is shown in Figure 5b.

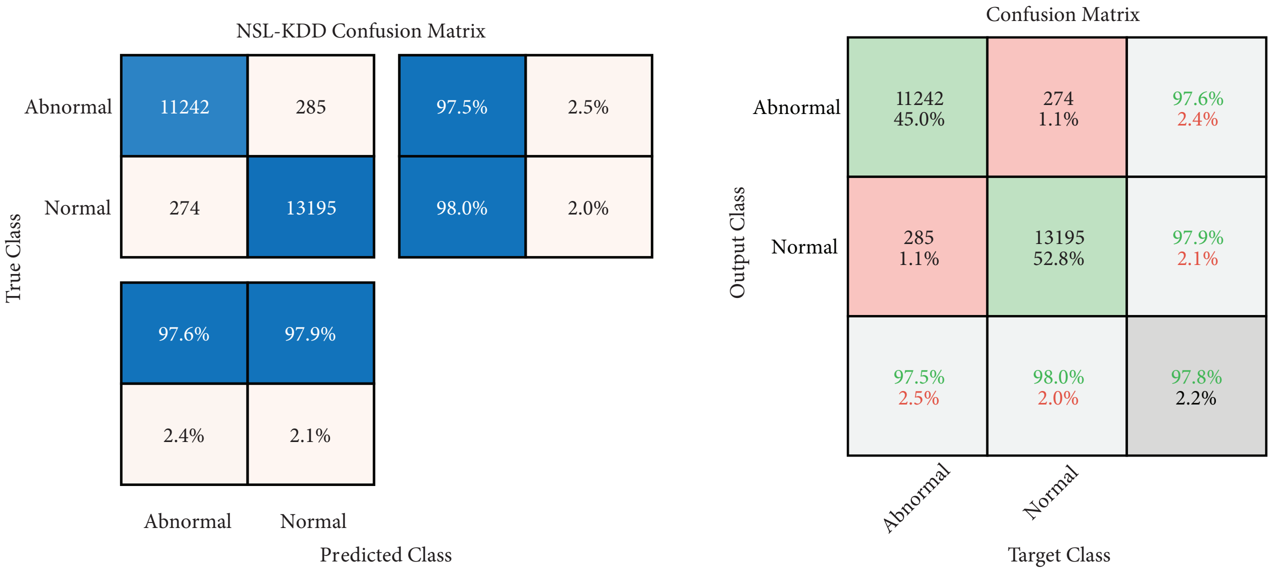
Table 4. Custom DNN NSL-KDD dataset partitions performances.

Dataset	Normal	DoS	Probe	R2L	U2R	Total
KDDTest-21	84.1	96.2	95.8	88.5	50.0	92.02
KDDTest	91.8	97.4	92.3	79.1	55.6	91.84
KDDTrain	96.9	99.4	94.6	90.5	21.4	97.54
KDDTrain-20	96.9	99.1	92.7	82.1	50.0	97.22

Custom CNN model shows a little better performance overall. Also, the training time is far shorter than AlexNet. This study is not intended to compare these two models anyway. The main goal in here is to show how data visualization will affect the overall performance. By comparing other DL studies on IDS with the same dataset, it is shown that V-IDS performs better with either architecture.

3.5.3. Binary training results

Another important criterion is performance of binary classification of a model with dataset like NSL-KDD because it is important to distinguish whether a request is attack or not beyond the classification of attacks. As expected, the success of the binary classification was slightly higher than the multiclass classification. Figures 6a and 6b show the confusion matrix of custom CNN architecture binary classification.



(a) Custom DNN Binary Classification Confusion Matrix

(b) Custom DNN Binary Classification Confusion Matrix

Figure 6. Custom DNN binary classification confusion matrix.

3.5.4. Comparisons with related studies

Table 5 is the comparison table of this study and the related studies. AC metric is used for comparison. AC metric, as mentioned before, is a value of how successfully we separate the intrusions. Comparisons are made in multiclass classification. Even though binary and multiclass classification were done in this study, the studies are compared according to multiclass accuracies, since not all studies contained both classifications.

4. Conclusion

In this study, the DL technique was applied to visualized NSL-KDD dataset which was named V-IDS and it was observed to be a noteworthy approach for the literature. Two DL architectures were used to prove that V-IDS provides significant increase in accuracy compared to other DL techniques used. One percent higher performance value was observed compared to its closest competitor. In addition, a higher success was achieved in binary classification. Also, it was shown that, from a different viewpoint, any technique can perform well with the CNN architectures which are said to yield lower accuracy with the NSL-KDD dataset.

In conclusion, this study showed that deep learning techniques, which are very close to human image perception ability in terms of accuracy, can be applied to classification tasks of any kind of data with the help of data visualization techniques.

Table 5. Comparison table of the related studies which used the NSL-KDD dataset.

Study	Technique	AC (%)
Ahsan et al. [34]	MCD-KDE	91.71
Ieracitano et al. [35]	AE50	87
Garg et al. [36]	SVM	87.56
Dong et al. [37]	MCA-LSTM	82.15
Dey et al.[38]	GRU-LSTM	87.91
Li et al. [39]	GINI-GBDT-PSO	86.10
Wu et al. [40]	CNN	79.48
Le et al.[41]	LSTM	92
Gogoi et al. [42]	TUIDS	96.55
Tang et al. [43]	DNN	91.7
Yin et al. [44]	RNN	83.28
Tang et al. [45]	RNN	89
Proposed technique	V-IDS	97.52

Acknowledgments

This work was supported by the projects of the **İnönü University Scientific Research Projects Department (SRPD) numbered FBG-2018-1107 and FBG-2020-2143**. The author would like to thank İnönü University SRPD for their valuable feedback.

References

[1] Anjana Tk. Discussion on Ransomware, Wannacry Ransomware and Cloud Storage Services against Ransom Malware attacks. International Journal for Research Trends and Innovation 2017; 2(6): 310.

[2] Vinayakumar R, Alazab M, Soman KP, Poornachandran P, Al-Nemrat A et al. Deep learning approach for intelligent intrusion detection system. IEEE Access 2019; 7: 41525-41550.

[3] Okhravi H, Meiners C, Streilein WW, Hobson T. A Study of Gaps in Attack Analysis. Technical report, Massachusetts Institute of Technology, Massachusetts, 2016.

[4] Aldweesh A, Derhab A, Emam AZ. Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. Knowledge-Based Systems 2020; 189: 105124.

[5] Mukkamala S, Janoski G, Sung A. Intrusion detection using neural networks and support vector machines. In: 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290); Hawaii, USA; 2002. pp. 1702-1707.

[6] Li W. Using genetic algorithm for network intrusion detection. In United States Department of Energy Cyber Security Group Training Conference 2004; Kansas City, Kansas; 2004. 24-27.

[7] Altwaijry H, Algarny S. Bayesian based intrusion detection system. Journal of King Saud University - Computer and Information Sciences 2012; 24(1): 1-6.

[8] Rai K, Devi MS, Guleria A. Decision tree based Algorithm for Intrusion Detection. International Journal of Advanced Networking and Applications 2016; 07(04):2828-2834.

[9] Devi TR, Badugu S. A review on network intrusion detection system using machine learning. Advances in Decision Sciences. Image Processing, Security and Computer Vision 2020;1: 598-607.

[10] Liao HJ, Lin CHR, Lin YC, Tung KY. Intrusion detection system: A comprehensive review. Journal of Network and Computer Applications, 2013; 36(1):16-24.

[11] Hansen C, Johnson C. The Visualization Handbook, USA: Elsevier, 2004.

[12] Keim DA. Information visualization and visual data mining. IEEE Transactions on Visualization and Computer Graphics 2002; 8(1):1-8.

[13] Chen W, Guo F, Wang F. A survey of traffic data visualization. IEEE Transactions on Intelligent Transportation Systems 2015; 16(6):2970-2984.

[14] Reynoso M, Diván M. Applying data visualization guideline on forest fires in argentina, 2020.

[15] Christopher Bishop. Pattern Recognition and Machine Learning, volume 27. Springer New York LLC, New York, 1 edition, 2006.

[16] Liu S, Wang X, Liu M, Zhu J. Towards better analysis of machine learning models: A visual analytics perspective. Vis. Informatics 2017; 1: 48-56.

[17] Rauber PE, Fadel SG, Falcão AX, Telea AC. Visualizing the hidden activity of artificial neural networks. IEEE Transactions on Visualization and Computer Graphics, 2017; 23(1):101-110.

[18] Fiore U, Palmieri F, Castiglione A, De Santis A. Network anomaly detection with the restricted boltzmann machine. Neurocomputing, 2013; 122: 13-23.

[19] Gao L, Li F, Xu X, Liu Y. Intrusion detection system using soeks and deep learning for in-vehicle security. Cluster Computing, 2018; 1: 1-9.

[20] Chakravarthi SS, Kannan RJ. Non-linear dimensionality reduction-based intrusion detection using deep autoencoder. International Journal of Advanced Computer Science and Applications, 2019; 10(8): 1-25.

[21] Liu H, Lang B. Machine learning and deep learning methods for intrusion detection systems: A survey. Applied Sciences (Switzerland) 2019; 9(20): 1-25.

[22] Aleesa AM, Zaidan BB, Zaidan AA, Sahar NM. Review of intrusion detection systems based on deep learning techniques: coherent taxonomy, challenges, motivations, recommendations, substantial analysis and future directions. Neural Computing and Applications 2019; 3: 20-44.

[23] Lee JH, Park KH. AE-CGAN model based high performance network intrusion detection system. Applied Sciences (Switzerland) 2019; 9(20): 1-20.

[24] Wang L, Jones R. Big data analytics of network traffic and attacks. NAECON 2018 - IEEE National Aerospace and Electronics Conference; Dayton, USA 2018. pp. 117-123.

[25] Lopez-Martin M, Carro B, Sanchez-Esguevillas A. Application of deep reinforcement learning to intrusion detection for supervised problems. Expert Systems with Applications 2020; 141:112963.

[26] Russakovsky O, Deng J, Su H, Krause J, Satheesh S et al ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) 2015; 115(3):211-252.

[27] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Communication ACM 2017; 60(6):84–90.

[28] Gatys LA, Ecker AS, Bethge M. Texture and art with deep neural networks. Current Opinion in Neurobiology, Computational Neuroscience 2017; 46: 178-186.

[29] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. CoRR 2015; 1: 1-20. abs/1409.1556.

[30] Dhanabal L, Shantharajah SP. A study on nsl-kdd dataset for intrusion detection system based on classification algorithms. 2015.

[31] Krizhevsky A, Sutskever I, Hinton G. Imagenet classification with deep convolutional neural networks. Neural Information Processing Systems 2012; 25: 01.

[32] Almisreb AA, Jamil N, Din NM. Utilizing alexnet deep transfer learning for ear recognition. Fourth International Conference on Information Retrieval and Knowledge Management (CAMP); Kota Kinabalu, Malaysia; 2018. pp. 1-5.

[33] Shin H, Roth HR, Gao M, Lu L, Xu Z et al. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. IEEE Transactions on Medical Imaging 2016; 35(5): 1285-1298.

[34] Ahsan M, Mashuri M, Lee MH, Kuswanto H, Prastyo DD. Robust adaptive multivariate hotelling's t2 control chart based on kernel density estimation for intrusion detection system. Expert Systems with Applications 2020; 145: 113105.

[35] Ieracitano C, Adeel A, Morabito FC, Hussain A. A novel statistical analysis and autoencoder driven intelligent intrusion detection approach. Neurocomputing, 2020; 387: 51-62.

[36] Garg S, Singh R, Obaidat MS, Bhalla VK, Sharma B. Statistical vertical reduction-based data abridging technique for big network traffic dataset. International Journal of Communication Systems 2020; 33(4): e4249.

[37] Dong R, Li X, Zhang Q, Yuan H. Network intrusion detection model based on multivariate correlation analysis – long short-time memory network. IET Information Security 2020; 14(2): 166-174.

[38] Dey SK, Rahman MM. Effects of machine learning approach in flow-based anomaly detection on software-defined networking. Symmetry 2020; 12(1): 1.

[39] Li L, Yu Y, Bai S, Cheng J, Chen X. Towards effective network intrusion detection: A hybrid model integrating gini index and GBDT with PSO. Journal of Sensors 2018; 1: 1-20. doi: 10.1155/2018/1578314.

[40] Wu K, Chen Z, Li W. A novel intrusion detection model for a massive network using convolutional neural networks. IEEE Access 2018; 6: 50850-50859.

[41] Le TTH, Kim Y, Kim H. Network intrusion detection based on novel feature selection model and various recurrent neural networks. Applied Sciences (Switzerland) 2019; 9(7): 4.

[42] Gogoi P, Bhuyan MH, Bhattacharyya DK, Kalita JK. Packet and Flow Based Network Intrusion Dataset. Contemporary Computing, Berlin, Heidelberg: Springer, 2012.

[43] Tang TA, Mhamdi L, McLernon D, Zaidi SAR, Ghogho M. Deep learning approach for network intrusion detection in software defined networking. International Conference on Wireless Networks and Mobile Communications (WINCOM); Morocco; 2016. pp.258-263.

[44] Yin C, Zhu Y, Fei J, He X. A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access 2017; 5: 21954-21961.

[45] Tang TA, Mhamdi L, McLernon D, Zaidi SAR, Ghogho M. Deep recurrent neural network for intrusion detection in sdn-based networks. 4th IEEE Conference on Network Softwarization and Workshops (NetSoft); Montreal, Canada; 2018. pp. 202-206.