Open Access **Baghdad Science Journal** **P-ISSN: 2078-8665**
2020, 17(2 Special Issue)NICST:701-709 **E-ISSN: 2411-7986**

DOI: http://dx.doi.org/10.21123/bsj.2020.17.2(SI).0701

Anomaly Detection Approach Based on Deep Neural Network and Dropout

Zaid Khalaf Hussien* ***Ban N.Dhannoon***

Received 7/9/2019, Accepted 20/2/2020, Published 23/6/2020

Abstract:

Regarding to the computer system security, the intrusion detection systems are fundamental components for discriminating attacks at the early stage. They monitor and analyze network traffics, looking for abnormal behaviors or attack signatures to detect intrusions in early time. However, many challenges arise while developing flexible and efficient network intrusion detection system (NIDS) for unforeseen attacks with high detection rate. In this paper, deep neural network (DNN) approach was proposed for anomaly detection NIDS. Dropout is the regularized technique used with DNN model to reduce the overfitting. The experimental results applied on NSL_KDD dataset. SoftMax output layer has been used with cross entropy loss function to enforce the proposed model in multiple classification, including five labels, one is normal and four others are attacks (Dos, R2L, U2L and Probe). Accuracy metric was used to evaluate the model performance. The proposed model accuracy achieved to 99.45%. Commonly the recognition time is reduced in the NIDS by using feature selection technique. The proposed DNN classifier implemented with feature selection algorithm, and obtained on accuracy reached to 99.27%.

Key words: Deep Learning, Dropout, Feature Selection, Network Security, NIDS.

Introduction:

There is no doubt, online application and the internet are important tools in our daily life. They have been used essentially in more fields such as, education and business. Therefore, network security is required to provide secure data channels(1). The network intrusion detection systems (NIDS) are a critical equipment's in the network system administration for detecting suspicious activities, the NIDS monitors and checks up the packets going into or leaving from the network devices, and logging the traffic and issuing warning notifications if the intrusion is detected. According to strategies of intrusion detection system, there are two kinds of NIDS depending on techniques to distinguish attacks (2). First one is signature-based detection, in spite of being unable to perceive the modern attacks, this way sustains the most prominent methodology within commercial IDSs. Anomaly-based detection is the second IDS kind, compares the new data by the model of normal user behavior in order to identify what is considered normal for the network as an anomaly by utilizing machine learning(3).

Department of Computer, College of Science, AL-Nahrain University, Baghdad, Iraq
*Corresponding author: stcs-zkh18@sc.nahrainuniv.edu.iq
*ORCID ID: https://orcid.org/0000-0002-6406-8499

The NIDSs are developed as classifiers to separate the normal traffic from the anomalous one(4). Deep learning has been emerged as a new method that could be utilized for Big Data in low training time consumption and high accuracy rate with its distinctive learning mechanism (5). Deep learning is non-linear approach within machine learning, it could be used for detection intrusions to develop adaptive IDSs(6, 7). Dropout is a regularized technique used to prevent the deep model from overfitting (8). Because of the large number of features, accommodation of the data in pattern detection becomes restricted sometimes. The feature selection method used with the classifier to provide enhanced estimation and decrease the implementation time (9).

Precisely, the significant contributions of this paper are:

- NIDS was provided by using DNN model. Dropout is technique used to reduce the overfitting.
- The proposed DNN model yields a detection rate of 99.45%, and it is able to classify the data into five class labels (normal, and four attack labels). The test outcomes demonstrate this approach is essentially potential for real time detection .

Related Work

Various studies have been considered for improving the classification problems, precisely in the intrusion detection system. The greater parts of the related works are:

1) *Reyadh Shaker Naoum1, et.al, "An Enhanced Resilient Backpropagation Artificial Neural Network for Intrusion Detection System", 2012(10)*. The authors proposed classifier system for intrusions utilizing an improved resilient backpropagation neural network. This classifier has ability to classify the records into five classes with a sensibly decent identification rate about 94.7% and with a false positive rate of 15.7%. The dataset which utilized in that analysis was NSL_KDD.

2) *Hee-su Chae, et.al "Feature Selection for Intrusion Detection using NSL-KDD" 2013 (11).* The authors in this paper have proposed feature selection method using autoregressive (AR) model and compared it with three feature selectors, correlation-based feature selection, Information Gain and Gain Ratio. The experiment shows that AR achieved the highest accuracy (99.794%) using 22 features.

3) *Ni GAO, Ling GAO, et.al," An Intrusion Detection Model Based on Deep Belief Networks" 2014*(12), this paper respected on intrusion detection classifier based on deep belief networks (DBN), which it is deep neural network classifier that combining from multilayer unsupervised learning networks called restricted Boltzmann machine (RBM), and a supervised learning network called Backpropagation network. The trial results on KDD CUP 1999 dataset and the classification achieved Acc=91.7%.

4) *Quamar Niyaz, et.al, " A Deep Learning Approach for Network Intrusion Detection System" 2016*(4), the authors used Self-taught Learning (STL), a deep learning-based technique. NSL-KDD dataset utilized in fitting and assessing. The soft-max regression (SMR) was utilized to order into 5-class, the classifier accuracy accomplished was79.10%.

5) *Manoj Kumar Putchala, "Deep Learning Approach For Intrusion Detection System (IDS) In The Internet Of Things (IOT) Network Using Gated Recurrent Neural Networks (GRU) 2017* (13). The author proposed the deep learning approach to develop the IDS model called Gated Recurrent Neural Networks, and he performed GRU on the KDD Cup99 data set for evaluating its performance, the experiment resulted that model had on an accuracy reached to 89.91%.

6) *Rana F. Najeeb, et.al, " A Feature Selection Approach Using Binary Firefly Algorithm For Network Intrusion Detection System" 2018 (14),* A wrapper type feature selection method based on binary firefly algorithm (BFA) and NBC was proposed and applied to intrusion detection system. The NSL-KDD dataset were used and it empirically proved that the randomization and movement of the FA were enhanced by calculation the hamming distance. The BFA achieve 92.02% accuracy rate with 14 features.

7) *R. Vinayakumar, Mamoun Alazab2, et.al, "Deep Learning Approach for Intelligent Intrusion Detection System" 2019* (15). The authors proposed IDS system, this system uses Deep Neural Network DNN model for anomaly detection and their experiments were performed on KDDCup 99 dataset, and as well as NSL-KDD. The results were shown as in Table (1).

Table 1. Model detection results on different datasets

Attack category	KDDCup99		NSL_KDD	
	Train	**Test**	**Train**	**Test**
Normal	97278	6593	67343	9710
Dos	391458	229853	45927	7458
Probe	4107	4166	11656	2422
R2L	1126	16189	995	2887
U2R	52	228	52	67
Total	494021	311029	125973	22544

8) *Ahmad HIJAZI, et.al," A Deep Learning Approach for Intrusion Detection System in Industry Network" 2019* (16). Authors proposed a deep approach to secure the industrial control systems ICS network. It is a multi-layer perceptron with binary classification. Simulated dataset from normal network traffic has been used for evaluating the model performance. They capture two types of data (normal and malicious packets) to train the neural network. The model accuracy achieved 99.89%.

Deep Neural Network (DNN)

Deep learning is a powerful gathering of methods used to learn the neural nets. The neural network is a biologically motivated paradigm enables the computer to learn from observational data (17). The expression "deep" usually alludes to the quantity of hidden layers with the neural net, each layer can be viewed as an individual algorithm all alone. DNN is one of the deep learning algorithms, which it is commonly used. The DNNs structure comprises from input layer, number of hidden layers, and output layer. Input data value are

Baghdad Science Journal
P-ISSN: 2078-8665
E-ISSN: 2411-7986

fed to the DNNs, and the output values are calculated progressively along the DNN hidden layers, at each layer, the input vector represented the output of every unit in the last hidden layer multiply with weight vector with each unit in the current layer in order to compute the weighted sum. At that point, the nonlinear function such as (hyperbolic tangent Tanh, sigmoid or rectified linear unit RELU) is utilized after weighted sum to compute the layer output values. The series of computation in layers change the representations into bit more abstract representations (18).

Dropout

A dropout is technique use to cripple the deep neural network by removing hidden units stochastically from it during training cases, to reduce data overfitting, we randomly omit hidden units with dropout rate 0.5. So, we are randomly sampling from collection of 2^n different thinned networks (***n*** is the number of units which can be dropped), and all these thinned networks share weights, this is as extreme as bagging can get. At testing phase, the geometric mean has been taken to all thinned network predictions to produce the mean network prediction. The 'mean network' that has all the outgoing weights halves(8). Dropout makes discourage brittle co-adaptations of hidden unit feature discriminators; it has been done by injecting with special kind of noise to the hidden output values through the forward pass of training phase. The noise zeros drops out to the limited fraction of the output values of the units within current layer, exactly like to the type of noise that added to the de-noising autoencoder input (19).

Feature Selection

The data features that are used to train the machine learning models have a great influence on the model performance. Unfortunately, a considerable lot of these features are either partially or totally irrelevant/redundant to the objective concept (20). Feature selection is a procedure utilizes to choose the best number of features needed to improve the data accuracy. By utilizing pertinent features, the classifier can in general improve its predictive accuracy(21).

The NSL-KDD Dataset

The NSL-KDD Dataset was set up to keep away from some characteristic issues of the KDD Cup 1999 Dataset. Indeed, even so it is generally old and not ideal representative to actual networks, it remained the perfect reference to show the contrast between the NIDS models. It was utilized in the past to assess the NIDS model performance by numerous researchers. This dataset contains 125,973 network traffic points in the KDD Train+ dataset (22). Each NSL-KDD dataset record builds on 41 features. It was readied by utilizing the system traffic captured by 1998 DARPA IDS assessment program the network traffic incorporates normal and various attack types, for example root-to-local (R2L), Probing, user-to-root (U2R), and DoS. It is sure that the vast majority of the recent attacks are possibly derived from the known attacks.

The Proposed Solution for NIDS

Simple deep neural network was constructed. NSL_KDD dataset was utilized to fit and assess the model, this dataset consists of 41 data features, and is categorized into five categories according to their characteristics, one is normal and the four others are attacks. SoftMax output layer with cross entropy loss function were used to enforce the model in multi class classification. Figure 1. represents general block diagram of the proposed system.

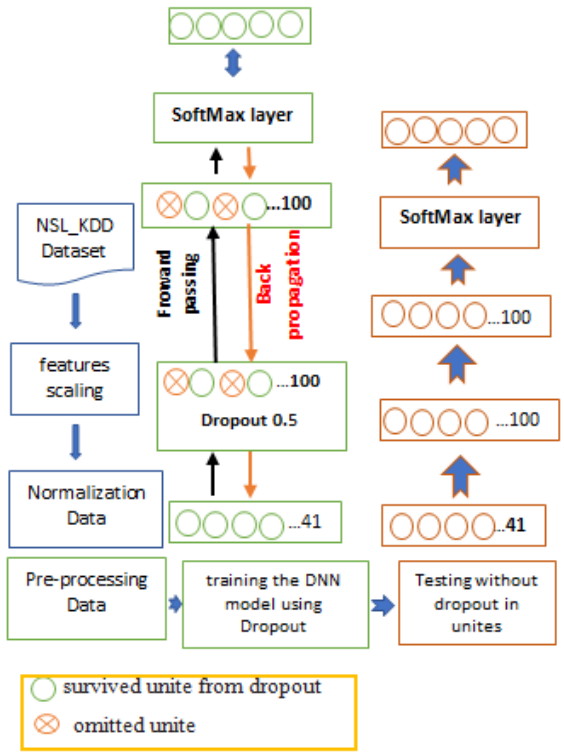

Figure 1. General block diagram of the proposed system

The Proposed Model Description

The proposed deep neural network is constructed with ***two*** hidden layers. $z^{(1)}$ signify the vector of inputs to layer l, and $y^{(1)}$ signify the vector of outputs consequent from layer l, $y^{(0)}$ = x is the input. $b^{(1)}$ and $w^{(1)}$ are the biases and weights at layer 1, where ***f*** is ReLU activation function, f***(x) = max (0, x).***

(**P** is the dropout rate= 0.5).

1. The Feed-Forward Operation Becomes:

$$r_j^{(1)} \sim Bernoulli(1-p), \quad (1)$$
$$y^{\wedge(1)} = r^{(1)} * y^{(1)},$$
$$z_i^{(2)} = \sum \frac{1}{1-p} w_i^{(2)} y^{\wedge 1} + b_i^{(2)},$$
$$y_i^{(2)} = f\left(z_i^{(2)}\right),$$

For more clarification see Fig.2

Here * denotes the element-wise multiplication, $\boldsymbol{r^{(1)}}$ is a vector of independent Bernoulli random variables every one of which has (1- ***p)*** of being 1. That vector is inspected and multiplied with elementwise by the outputs of that layer $y^{(1)}$, to create the thinned outputs $y^{\sim(1)}$. At that point these thinned outputs are utilized as input to the following layer. This procedure is applied at each layer. The factor of $\frac{1}{1-p}$ utilized during training phase to ensure at test time, each input will reach each layer when all units get utilized.

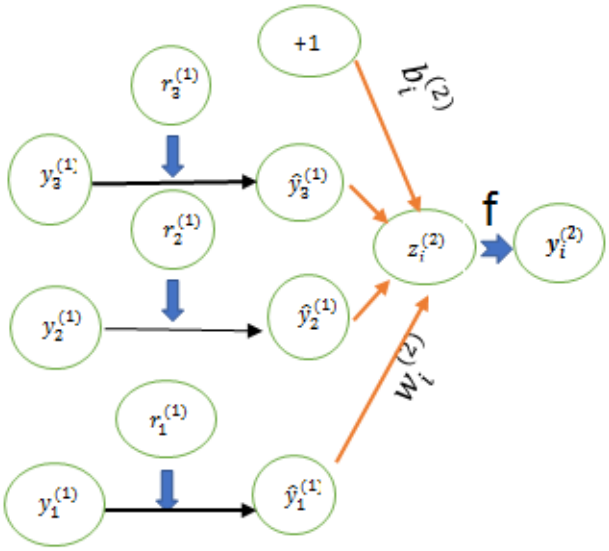

Figure 2. Feed-forward with dropout

2. Cross-entropy Function and SoftMax Output Layer:

The input of the proposed model to the first layer will be 41 nodes and the output will be five nodes. We will utilize the root mean squared-error cross entropy calculation as loss function, to calculate the contrast between two probability distributions. Normally the true distribution (the one that the machine learning calculation is attempting to match), and predict distribution as follows:

$$E = H(p,q) = -\sum_x p(x) \log q(x). \quad (2)$$

Where $\boldsymbol{p}$ is the target distribution, $\boldsymbol{q}$ is the predicted distribution.

SoftMax is to characterize another sort of output layer for the proposed neural networks, it will map the last hidden neurons to output nodes, where the SoftMax activation function is:

$$\boldsymbol{a_k^L = S_{i^{th}\ class}(z_k^L) = \frac{e^{z_k^L}}{\sum_{n\ class} e^{z_k^L}}} \quad (3)$$

a_k^L refers to neuron activation function in last layer. z_k^L refers to the neuron input in last layer as shown in Fig.3. This activation function starts in the same way as with a ReLU layer, by forming the weighted input:

$z_j^L = \sum_k w_{jk}^k \, a_k^{L-1} + b_j^L$ However, the ReLU function to get the output is not applied. Rather, a SoftMax function is applied to the z_j^L. As indicated by this function, the activation a_j^L to j^{th}output neuron.

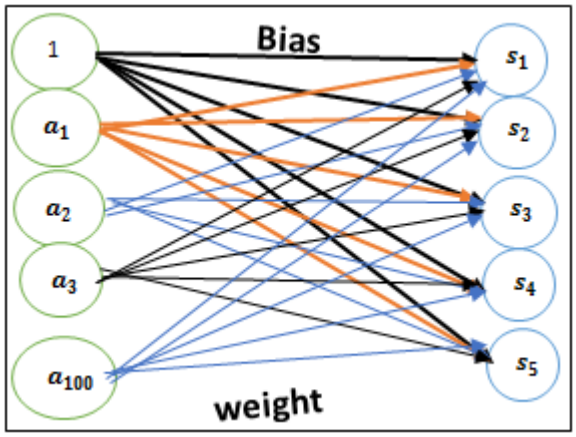

Figure 3. SoftMax output layer

In testing phase, geometric mean network is used, it contains the total hidden units but with their outgoing weights halved. This gives exceptionally closely performance to averaging along a vast number of dropout networks. Mean network is actually as taking the geometric mean of the probability distributions over classes predicted by 2^N possible thinned networks (N is the number of units which can be dropped).

These thinned networks do not all make same predictions, and mean network prediction is destined to be a higher log probability for the right answer than the log probabilities denoted by the individual thinned networks.

Each thinned network estimator is defined as:

$$\boldsymbol{\hat{y} = arg\ max_y P(Y = y/z)} \quad (4)$$

The geometric mean of all predictions of these thinned nets, that each one can be computed as in equation (4):

$$\boldsymbol{\hat{y}geometric\ mean\ network} = \left(\prod_{i=1}^{\mathbf{k}} \hat{\boldsymbol{y}}_i\right)^{\frac{1}{k}} \quad (5)$$

Where k indicates to number of all thinned nets caused by dropout during training case.

Baghdad Science Journal
P-ISSN: 2078-8665
E-ISSN: 2411-7986

Performance Evaluation

Usually, the performance of ANIDS models are assessed in terms of accuracy, recall, precision, and F-score. NIDS needed high detection rate/accuracy. The confusion matrix is utilized to compute these metrics.

1. Confusion Matrix

A confusion matrix shows the quantity of incorrect and right forecasts come about by the classification model contrasted with the actual outcomes (the objective value) in the data. The matrix M×M, where M is number of labels values. In the confusion matrix, TP (true positive) is the quantity of attack records effectively arranged. TN (true negative) which is the quantity of normal records effectively classified. and the number of normal records incorrectly classified is FP (false positive). FN (false negative) is the number of attack records incorrectly classified. (P and N) positive, and negative samples, respectively [3].

Experimental Results

The experiments have been applied on NSL-KDD dataset to fit and test the model by two estimation methods (holdout, and 5-fold cross validation), and in two cases, one in total 41 features values, and the other in using feature selection method.

1. Experimental Results Using 41 Data Features

The experiments have been done by using total 41 NSL-KDD feature values.

A. Results by 5-folds Cross Validation

Cross validation method is a way to estimate the skill of model on unseen data, but it is the greater computational expense. This method systematically creates and evaluates multiple classifiers on multiple data subsets. As shown in Figure 4.

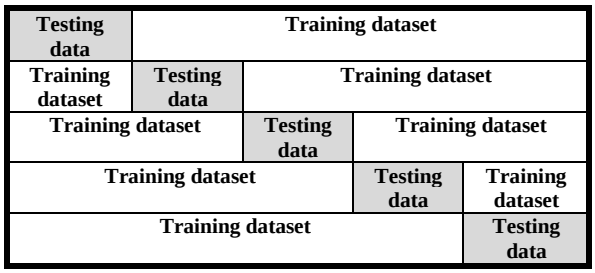

Figure 4. Five folds cross validation

In our experiment, 5_folds cross validation was used. The "KDDTrain.csv" (contains on125973 data points) has been partitioned into 100778 training data points, and 25196 testing data points (80% for training rate, and 20% for testing). First hidden layer units will be ReLU activation function, and second hidden layer units are sigmoid activation function, learning rate was (0.1), and the number of epochs was 150. the results are shown in Table 2.

Table 2. Experimental results by cross validation

#	# units in each layer	class 1 acc%	class 2 acc%	class 3 acc%	class 4 acc%	class 5 acc%	Avg acc%
1	30/30	85.51	85.16	85.38	86.32	85.75	85.62
2	60/60	84.93	82.98	85.44	84.44	85.23	84.6
3	100/100	85.93	86.23	85.80	85.45	85.90	85.86
4	200/200	82.59	79.81	82.74	82.26	85.75	82.63

After seeing these results, third result was selected, the model accuracy was 85.86%. This model setting will be selected to apply in the next holdout estimating method.

B. Results by Holdout Method

NSL_KDD dataset has 125,973 network traffic samples stored in" KDDTrain+.csv" file. This dataset has been partitioned into 100778 data points to train, and the remain 25192 points to test. Training data also partitioned into 67521points for training and the 33257 points to validate (training rate=67%, and validation rate=33%). See Fig. 5 followed.

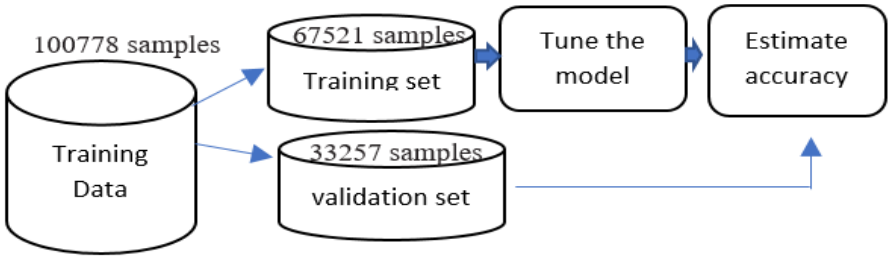

Figure 5. Training, and validation sub-dataset

Experiments have been implemented in **two stages** with different number of iterations. the starting setting to proposed model, consists of (41 input nodes, 100 ,100 ,5 output nodes) first hidden layer consisted of 100 units of ReLU activation function, second one had 100 units of sigmoid activation function, the learning rate was 0.1, dropout rate was 0.5, and number of epochs were 150. The **first** stage was utilized to identify the adequate type of activation function was utilized to hidden units in each layer, by shifting the activation function types among each ElU, ReLU, Tanh, and Sigmoid. Table 3 shows the results of experiment in this stage.

Table 3. Results with replacing among activation function types

#	Unit types in each hidden layer	Training Acc%	Testing acc %
1	ReLU/sigmoid	99.21	99.11
2	ElU/ sigmoid	99.7	99
3	ReLU/ tanh	99.29	99.26
4	ElU/ tanh	99	98.9

According to these results, third result was selected to next stage, it achieved a highest accuracy (99.26%). In second stage, adequate learning rate was determined. Table 4 shows the results. Here, the number of epochs were 300, and dropout rate was 0.5.

Table 4. Experiments results with different learning rate

#	Learning rate	Training Acc%	Testing Acc%
1	0.001	99.16	99.12
2	0.01	99.16	99
3	0.1	99.49	99.45

From these results, the proposed model accuracy reached to 99.45%, and it will be the proposed model accuracy.

C. Performance Evaluation

As we recommended previously, the performance of NIDS models are assessed by accuracy, recall, precision, and F-score. The confusion matrix is utilized to compute these parameters. Confusion matrix shows the quantity of incorrect and right forecasts come about by the classification model contrasted with the actual outputs, as shown in Table 5.

Table 5. Confusion matrix resulted by DNN model using 41 data features

		Predicted				
		DoS	**U2R**	**R2L**	**Probe**	**Normal**
Actual	**DoS**	9274	0	0	2	12
	U2R	0	4	1	0	9
	R2L	0	1	170	0	34
	Probe	0	0	0	2241	24
	Normal	15	0	17	22	13369

To compute the model performance accuracy, this formula will be used.

$$\text{Accuracy}=\frac{sum\ all\ TP's}{total\ all\ classification}=0.9945624 \quad (6)$$

The proposed model can classify the labeled testing dataset, as shown in Table 6.

Table 6. Labeled Testing Results

Class Labels	Class Size	Detected Size	Detection Rate
DOS	9289	9274	0.998385
U2R	5	4	0.8
R2L	188	170	0.904255
Probe	2265	2241	0.989404
Normal	13448	13369	0.994126
Total	25195	25058	0.994562

F-score, recall, and precision are important metrics. To calculate them, TP, FP, and FN parameters must be computed first, and onfusion matrix is used. See Table 7.

Table 7. F-measure, recall, and precision values

Class labels	TP	FP	FN	Precision	Recall	F1 score
DoS	**9274**	**15**	**14**	**0.9984**	**0.9985**	**0.998**
U2R	**4**	**1**	**10**	**0.8**	**0.285**	**0.42**
R2L	**170**	**18**	**35**	**0.9043**	**0.8293**	**0.865**
Probe	**2241**	**24**	**24**	**0.9894**	**0.9894**	**0.989**
Normal	**13369**	**79**	**54**	**0.9941**	**0.996**	**0.995**

Baghdad Science Journal
P-ISSN: 2078-8665
E-ISSN: 2411-7986

Figure 6 shows the accuracy history chart, that was recorded on model implementation during training phase:

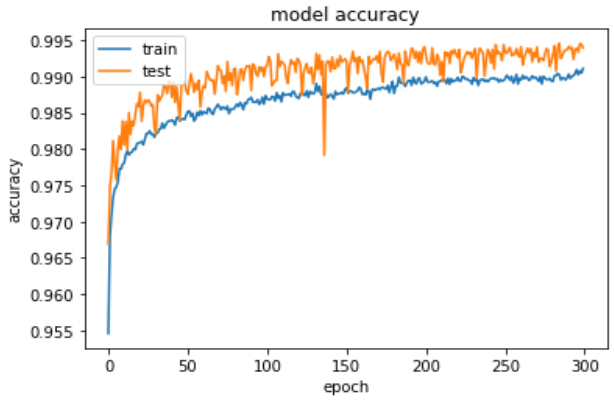

Figure 6. Accuracy chart during 300 epochs

2. Results with Using Feature Selection Method

The experiments have been done using feature selection technique called "SelectKBest" (a class of the **sklearn. feature_selection** module within python programing language used for selection/dimensionality reduction on sample dataset. This method is based on (F-test) for estimating the mutual information between features before scoring the features to k (highest scores, sets optionally by the user), **chi2** is used as f_regression metric in this method). These experiments have been implemented in **two** stages. **First** one was used to determine the best minimum number of features could be selected, and also determining the adequate dropout rate. The results were shown in Table 8, during this stage, learning rate was 0.1. First hidden layer unites were 100 ReLU activation functions, and second hidden layer unites were 100 sigmoid activation functions.

Table 8. Results with different feature numbers and dropout rate

#	epochs	Dropout rate	#Selected Features	Training Acc%	Test Acc%
1	150	0.5	10	94.81	94.66
2	150	0.5	15	96.42	96.37
3	150	0.5	22	96.76	96.74
4	350	None	22	99.29	98.73
5	350	0.5	22	99.21	99
6	350	0.5	25	98.86	98.82
7	350	0.2	22	98.76	98.63
8	277	0.5	38	99.43	99.2
9	250	0.5	36	99.41	99.27
10	280	0.5	34	99.48	99.19
11	150	0.5	37	99.25	99.14
12	277	0.5	32	99.12	98.8
13	250	None	36	99.57	99
14	230	0.2	36	99.59	99.1

From these results, the ninth one was chosen. Now the final setting to proposed model was illustrated. The accuracy of the proposed model reached to 99.27%. Table 9 illustrates the confusion matrix resulted from this experiment.

Table 9. The confusion matrix of DNN model in feature selection method.

		Predicted				
		DoS	**U2R**	**R2L**	**Probe**	**Normal**
Actual	**DoS**	9046	0	0	2	17
	U2R	0	4	1	0	5
	R2L	0	0	119	0	79
	Probe	0	0	0	2326	34
	Normal	6	4	7	29	13516

Table 10 shows how the proposed DNN model classification to the labeled testing data.

Table 10. Labeled Testing Results.

Class labels	Class size	Detected size	Detection rate
DOS	9052	9046	0.999337
U2R	8	4	0.50
R2L	127	119	0.937008
Probe	2357	2326	0.986848
Normal	13651	13516	0.990111
Total	25195	25011	0.992697

Table 11 shows the F-measure, recall, and precision metric values to the proposed model.

Table 11. F-measure, recall, and precision metric values.

Class labels	TP	FP	FN	Precision	Recall	F1 score
DoS	9046	6	19	0.99933	0.997	0.998
U2R	4	4	6	0.50	0.40	0.444
R2L	119	8	79	0.93700	0.601	0.732
Probe	2326	31	34	0.98684	0.985	0.986
Normal	13516	135	46	0.99011	0.996	0.993

Figure 7 shows the accuracy chart of the proposed model using 3 6features, during the training phase.

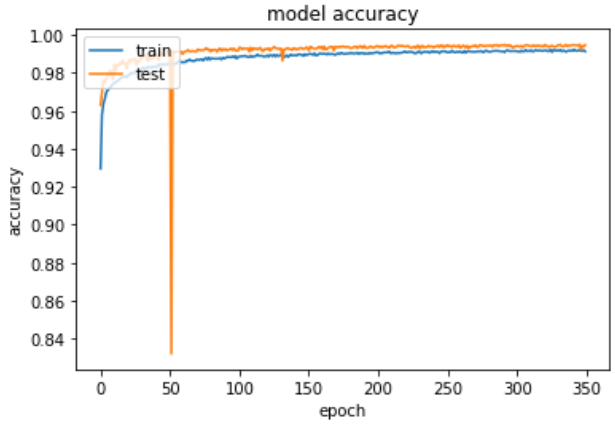

Figure 7. The classifier accuracy chart during training

In spite of this model has accuracy reached to 99.27% less than last one 99.46%, but, this model is the robust, because of when you are looking at Fig. 5 about the training chart of the model with whole 41 NSL_KDD, you can watch the training and validation curves are noisy during their raising along epochs raising, but in Fig. 6 about training the model using feature selection method (36 data features) the curves are smother than last one, also the training phase time has less consumption, without feature selection method, the time has more consumption than without that method. The average time to each epoch without feature selection method was 11 seconds, but with selection method was 9 seconds. This means the time consumption to train the model without selection method (300 epochs * 11 seconds= 3300 seconds) when the accuracy reached to 99.45%, but with feature selection method (250 epochs * 9 seconds=2250) to get accuracy to 99.27%.

Conclusions:

We have performed a deep learning algorithm such as DNN model to detect network intrusion. A dropout technique and soft-max regression-based NIDS were presented. The main point of this research paper is exhibiting that deep learning technique can be applied in intrusion identification domain. The DNN classifier was optimized by dropout to reduce the overfitting, this technique utilized to temporarily removing some hidden units in the deep neural network alongside the entirety of its incoming and outgoing connections, to learn more robust attributes. The model implemented on two cases. One with total NSL_KDD features data, its performance achieved to 99.45%, and other case with feature selection method which its performance achieved to 99.27%. The experimental results showed that the proposed model has the ability to learn proficiency in real time with or without using feature selection method, as a superior generative model and perform well on intrusion identification issues.

Authors' declaration:

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are mine ours. Besides, the Figures and images, which are not mine ours, have been given the permission for re-publication attached with the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee in AL-Nahrain University.

References:

1. Salama MA, Eid HF, Ramadan RA, Darwish A, Hassanien AE. Hybrid intelligent intrusion detection scheme. Soft computing in industrial applications: Springer; 2011. p. 293-303.
2. Jain S, Kumar A, Mandal S, Ong J, Poutievski L, Singh A, et al., editors. B4: Experience with a globally-deployed software defined WAN. ACM SIGCOMM Computer Communication Review; 2013: ACM.
3. Xu D, Ricci E, Yan Y, Song J, Sebe N. Learning deep representations of appearance and motion for anomalous event detection. arXiv preprint arXiv:151001553. 2015.
4. Javaid A, Niyaz Q, Sun W, Alam M, editors. A deep learning approach for network intrusion detection system. Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS); 2016: ICST (Institute for Computer Sciences, Social-Informatics and .
5. Karatas G, Demir O, Sahingoz OK, editors. Deep learning in intrusion detection systems. 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT); 2018: IEEE.
6. Debar H, Becker M, Siboni D, editors. A neural network component for an intrusion detection system. IEEE symposium on security and privacy; 1992.

7. Sommer R, Paxson V, editors. Outside the closed world: On using machine learning for network intrusion detection. Security and Privacy (SP), 2010 IEEE Symposium on; 2010: IEEE.
8. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research. 2014;15(1):1929-58.
9. Najeeb RF, Dhannoon BN. Classification for intrusion detection with different feature selection methods: a survey (2014–2016).IJARCSSE.2017;7(5).
10. Naoum RS, Abid NA, Al-Sultani ZN. An enhanced resilient backpropagation artificial neural network for intrusion detection system. IJCSNS. 2012;12(3):11.
11. Chae H-s, Jo B-o, Choi S-H, Park T-k. Feature selection for intrusion detection using nsl-kdd. RACS. 2013:184-7.
12. Gao N, Gao L, Gao Q, Wang H, editors. An intrusion detection model based on deep belief networks. Advanced Cloud and Big Data (CBD), 2014 Second International Conference on; 2014: IEEE.
13. Putchala MK. Deep learning approach for intrusion detection system (ids) in the internet of things (iot) network using gated recurrent neural networks (gru). 2017.
14. Najeeb RF, Dhannoon BN. Improving Detection Rate of the Network Intrusion Detection System Based on Wrapper Feature Selection Approach. Iraqi Journal of science. 2018;59(1B):426-33.
15. Vinayakumar R, Alazab M, Soman K, Poornachandran P, Al-Nemrat A, Venkatraman S. Deep Learning Approach for Intelligent Intrusion Detection System. IEEE Access. 2019;7:41525-50.
16. Hijazi A, Flaus J-M. A Deep Learning Approach for Intrusion Detection System in Industry Network2019.
17. Buduma N, Locascio N. Fundamentals of deep learning: Designing next-generation machine intelligence algorithms: "O'Reilly Media, Inc."; 2017.
18. Min S, Lee B, Yoon S. Deep learning in bioinformatics. Briefings in bioinformatics. 2017;18(5):851-69.
19. Gal Y, Ghahramani Z, editors. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. international conference on machine learning; 2016.
20. Dash M, Liu H. Feature selection for classification. Intelligent data analysis. 1997;1(1-4):131-56.
21. Eesa AS, Orman Z, Brifcani AMA. A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems.Expert Syst Appl. 2015;42(5):2670-9.
22. Tavallaee M, Bagheri E, Lu W, Ghorbani AA, editors. A detailed analysis of the KDD CUP 99 data set. Computational Intelligence for Security and Defense Applications, 2009 CISDA 2009 IEEE Symposium on; 2009: IEEE.

نهج الكشف عن التسلل على أساس الشبكة العصبية العميقة وتقنية التسقيط

زيد خلف حسين **بان نديم ذنون**

قسم الحاسبات، كلية العلوم، جامعة النهرين، بغداد، العراق.

الخلاصة:

فيما يتعلق بأمان نظام الكمبيوتر، تعد أنظمة كشف التسلل هي من المكونات الأساسية لتمييز الهجمات في المرحلها المبكرة. حيث انها تراقب وتحلل محطات الشبكة، وتبحث عن سلوكيات غير طبيعية أو توقعات هجومية لكشفها في وقت مبكر. ومع ذلك، نشات العديد من التحديات أثناء تطوير انظمة الكشف من حيث كونه نظام مرن ونشط للهجمات غير المتوقعة. في هذه الرسالة ، نقترح مصنف متكون من الشبكة العصبية العميقة لتكوين نظام كشف الخروقات الشبكي. حيث ان هذا المصنف مُحسن باستخدام تقنية التسقيط الذي يعمل على تجاهل بعض الوحدات في الطبقات المخفية، مؤقتًا في الشبكة العصبية العميقة في مرحلة التدريب، مما يؤدي إلى نتائج تصنيف جيدة بحيث يقلل على النموذج او المصنف من الوقوع في مشكلة (Overfitting). تحاول تقنية التسقيط إضافة ضوضاء معينة تسمى (ضوضاء برنولي) إلى مخرجات الوحدة المخفية عند تمريرها الامامي للبيانات في الشبكة، في مرحلة للتدريب. اذا كانت هذه الضوضاء أصفار فانها توقف او تثبط جزء من عدد الوحدات العصبية في الطبقة التي تتعرض للتعطيل، في حالة الشبكة العصبية تحوي على n من الوحدات المخفية، فان مجموع الشبكات العصبية الرقيقة المحتملة عددها 2^n. وهذه الشبكات العصبية الرقيقة تشترك في الاوزان. لذلك يتم تدريب عدد قليل من الشبكات الرقيقة ويحصلون على نموذج تدريب واحد فقط. في مرحلة الاختبار، تحسب شبكة المتوسط الهندسي لتنبؤات جميع الشبكات الرقيقة في وقت الاختبار. النتائج التجريبية اجريت على بيانات NSL_KDD. تم استخدام طبقة مخرجات (SoftMax) مع دالة فقدان الانتروبيا المتقاطعة لتمكين المصنف في التصنيفات المتعددة بما في ذلك خمس فئات، واحد طبيعي (Normal) والأربعة الأخرى هي هجمات (Dos و R2L و U2L و Probe). استخدمت الدقة لتقييم أداء النموذج ووصلت دقة اداء المصنف الى 99.46%. يتم تقليل وقت الكشف في الغالب في مصنفات انظمة كشف الخروقات الشبكي باستخدام تقنية اختيار الصفة. حيث تم تحسين أداء نظام كشف التسلل في الكشف عن الهجمات بواسطة مصنف الشبكة العصبية العميقة وخوارزمية اختيار الصفة. وحققت دقة مقدارها 99.27٪.

الكلمات المفتاحية: أمان الشبكات، نظام كشف الخروقات الشبكي، التعلم العميق، التسقيط، اختيار الصفة.