ORIGINAL ARTICLE

Deep Learning Method for Denial of Service Attack Detection Based on Restricted Boltzmann Machine

Yadigar Imamverdiyev and Fargana Abdullayeva*

Abstract

In this article, the application of the deep learning method based on Gaussian–Bernoulli type restricted Boltzmann machine (RBM) to the detection of denial of service (DoS) attacks is considered. To increase the DoS attack detection accuracy, seven additional layers are added between the visible and the hidden layers of the RBM. Accurate results in DoS attack detection are obtained by optimization of the hyperparameters of the proposed deep RBM model. The form of the RBM that allows application of the continuous data is used. In this type of RBM, the probability distribution of the visible layer is replaced by a Gaussian distribution. Comparative analysis of the accuracy of the proposed method with Bernoulli–Bernoulli RBM, Gaussian–Bernoulli RBM, deep belief network type deep learning methods on DoS attack detection is provided. Detection accuracy of the methods is verified on the NSL-KDD data set. Higher accuracy from the proposed multilayer deep Gaussian–Bernoulli type RBM is obtained.

Keywords: deep learning; deep belief network; restricted Boltzmann machine; NSL-KDD

Introduction

In the CIA (confidentiality, integrity, and availability) triad, the primary aim of denial of service (DoS) attacks is the violation of the information availability. The occurrence of natural causes (e.g. technical accidents), errors (poor configuration of systems and services), and DoS attacks can result in the violation of availability of information resources, networks, and e-services.

DoS attacks are malicious actions that can prevent accessing of the legitimate users to the system, network, application software, or information. There are different types of DoS attacks, they become a single-source (launched from one system) and distributed (launched from numerous systems) form. Distributed DoS (DDoS) attacks are a specific type of DoS attacks. DDoS attacks target Internet infrastructure objects (including routers and Domain Name System (DNS) servers), bandwidths, and servers (including Hyper Text Transfer Protocol (HTTP), DNS, and other services).¹

Nowadays the implementation of DoS attacks does not require special knowledge in the information technology field. Thus, the realization of this attack can be

ordered at a cheap price or can be used as a special tool designed by hackers. There are economical (struggle against adversary), political, terrorism, and cybercrime motives of DoS attacks. There are facts of using DDoS attacks for political purposes against countries, political parties, and social movements.² According to the DDoS attacks in Q3 2017 report by Kaspersky laboratory, resources in 98 countries were attacked in the third quarter of 2017.³ Hackers can use the special technology and distributed infrastructure (botnets) to reach a huge speed of DDoS attacks.⁴ This allows isolating even the information infrastructure of the whole country from the Internet.

The problem of DDoS attacks detection has been widely studied in the past decade and is still of high interest. Machine learning methods have been developed to detect the DDoS attacks using classification and clustering algorithms. One of the drawbacks of traditional machine learning algorithms is that they use hand-crafted features for recognition task.

This is not “true” machine learning. It is desirable that the machine itself found and structured features for attack detection.

Institute of Information Technology, Azerbaijan National Academy of Sciences, Baku, Azerbaijan.

*Address correspondence to: Fargana Abdullayeva, Institute of Information Technology, Azerbaijan National Academy of Sciences, 9A, B. Vahabzade Street, AZ1141, Baku, Azerbaijan, E-mail: a_fargana@mail.ru

Motivations

The main purpose of the network attack detection system is to differentiate malicious behavior from normal network traffic. This type of analysis to meet with cybersecurity, in which the nature of the new attacks is not known in advance and appeared in real time on a continuous basis, should be carried out in a flexible and effective way. This process can be carried out by constructing the model of any type of attack that can influence the network, or simply by constructing a normal traffic common model.

Such a model is usually built on training data and is used to classify previously unknown or suspicious events. In the attack detection problem, the classification becomes one of the fundamental issues. Learning through the classification, the system recognizes complex traffic patterns, distinguishes different events according to matching templates, and makes intelligent decisions. For accurately classifying data to one of the attack types, machine learning methods such as neural networks⁵ support vector machines (SVMs) are used.^{6,7}

Currently, deep learning is one of the most intensive research trends in the field of artificial intelligence^{8,9} and opens wide opportunities to overcome the constraints of traditional machine learning methods. In traditional machine learning algorithms, the features are extracted by the humans. There is a special research direction—feature engineering. But in the big data processing, deep neural networks work better than a human in feature extraction. In the speech recognition, natural language processing, computer vision, and other fields, they represent better results than alternative methods.¹⁰

Successful applications of deep learning methods in different fields attract the attention in the cybersecurity field, too. Deep learning is often considered as identical with deep neural networks (DNNs). DNNs do not have a universally accepted definition, usually, the neural networks with more than one hidden layer are called DNNs. One of the reasons for the successful use of DNNs is that these networks automatically extract from data important features needed to solve the cyber attack detection problem. In traditional machine learning algorithms, features should be extracted by the human,¹¹ there is a special direction in machine learning research called feature engineering. But in big data processing, the DNNs work better than humans in feature extracting. DNN by using several processing layers learns the representation of the data with multilevel abstraction. Representation on each layer is formed on the basis of the representation of the previous layer.

Still, deep learning methods are less applied to DDoS attacks detection, only a few studies have been published.^{12,13}

The purpose of this article is to develop an approach based on DNNs for accurate and adaptive detection of DoS attacks in real time. To achieve this purpose, the self-learning discriminative restricted Boltzmann machine (RBM), based on the energy model of the network of stochastic neurons, is used. As the data classes are not used in the learning process, the RBM model can detect attacks in real time with high precision and adaptability.

Contributions

The contributions of this article are the following:

- Multilayer RBM-based DoS attack detection method is proposed. To obtain the accurate results in DoS attack detection, the hyperparameters of the proposed deep RBM model are optimized.
- RBM has been trained on standardized (normalized) data set. In the training process of the RBM, the labels of the data set are not used.
- Trained features of RBM are used as an input data for learning the next layers of the neural network. In the result, the high-dimensional unlabeled data have been selected as optimal small-sized features.
- The effectiveness of the RBM method is evaluated on the NSL-KDD data set. The advantage of detection accuracy of the proposed deep RBM model in comparison with the SVM, decision tree, Bernoulli-Bernoulli RBM, Gaussian-Bernoulli RBM, and deep belief network (DBN) is demonstrated.

The remainder of this article is organized as follows: DDoS Attacks and Detection Methods: A Brief Overview section introduces brief information on DDoS attacks. Related Works section presents the related work of DNNs and the related work of DDoS detection using deep learning. The Restricted BM section introduces the proposed solution of detecting DDoS attacks by RBMs. The Experiments and Discussion section presents results of experiments. Finally, the Conclusions section presents the concluding remarks.

DDoS Attacks and Detection Methods: A Brief Overview

The first large-scale DDoS attacks are conducted in early February 2000 to the large companies such as Yahoo!, eBay, and CNN, and since then, the development of preventive mechanisms against DDoS attacks

focuses on the center of attention of network practitioners and security researchers.

There are the following types of DDoS attacks:

- Volumetric attacks—catch the bandwidth of the target server; measurement unit—bits per second. Examples—ICMP-, UDP-, TCP-flood attacks.
- Protocol-based attacks—capture the resources of the target server; measurement unit—packages per second. Examples—Ping flood, SYN-flood, Smurf attacks.
- Application level attacks—by using the application level vulnerabilities target server is crashed. Examples are Hash DoS attacks and Teardrop attacks.

A great deal of research on the topic of DDoS attacks has been carried out and in recent years, there are several important review articles analyzing the taxonomy of DDoS attacks, attack mechanisms, proposed architectures, and methods for protection from DDoS attack; their advantages and disadvantages have been published. Suggested approaches for detecting DDoS attacks can be divided into initial approaches, statistical analysis, knowledge-based soft computing, and machine learning methods.^{14–19}

Statistical methods are based on quantitative analysis of traffic, and signature methods are based on quality analysis of traffic. Standard methods of statistical analysis and knowledge-based methods do not allow to detect previously unknown attacks, and, therefore, machine learning methods are widely used as a solution mechanism for this problem.

Related Works

Deep neural networks

There are many methods referred to as deep learning (DL) and in Ref.²⁰ they are divided into three groups: generative, discriminative, and hybrid. Generative architectural methods include autoencoders,²¹ recurrent neural networks,²² and Boltzmann machines (BMs).

Each layer of the deep network learns independently, bypassing the previous pretraining procedure. It then allows checking a good initial approach to run the backpropagation algorithm. Depending on the selected model, each layer may be RBM or CNN (convolutional neural network).²³

BM is a network of symmetrically connected stochastic binary units. The units are divided into two groups, by describing visible and hidden states (analogy with hidden Markov models). The states of the visible

and the hidden neurons are varied according to the probability activation functions.

RBM is a BM that has no relationship between the hidden layer neurons. Owing to the special bipartite graph structure, it is possible to clearly find the probabilities of the hidden layer neurons. If a sufficient number of neurons are used in the hidden layer, RBM can generate any discrete distribution.

RBM is a key structural unit for constructing the DBN. DBN is a multilayer network, in which the lower layers are sigmoid belief network and the upper layer is RBM.

Deep BM (DBM) is sometimes used in the pretraining step instead of the autoencoder. Multilayer architecture of the DBM is the main difference from RBM.

Hybrid DL architecture integrates the generative and discriminative architectures. DNN can be given as an example of hybrid architecture. In Ref.,²⁴ DNN is a cascade of fully connected hidden layers and often uses the RBM stack as a pretraining stage.

DDoS attack detection using deep learning

Computer network attacks detection is one of the areas that have been investigated for a long time and new ideas have been developed in numerous approaches. In Ref.,²⁵ a hybrid method based on a DBN and SVM is proposed. Here, the DBN is used for the feature selection and the SVM for classification, and by this method, the NSL-KDD data set is classified into five classes, such as Normal, Remote to Local (R2L), DoS, User to Root (U2R), and Probing. Based on the proposed concept, by using the DBN, the data set is reduced from 41 to 5 features and then SVM is applied to these reduced data and classification is performed.⁶ By applying sparse autoencoder and softmax regression type self-taught learning approach, a network attacks detection method is proposed. The effectiveness of the intrusion detection system constructed on self-taught learning has been tested on the NSL-KDD data set.

In Ref.²⁶ on the basis of autoencoder and DBN, the malware detection method is proposed. Here primarily by applying the autoencoder, the key features of the data are extracted and data volume reduction is carried out. Then by applying the DBN learning method, the malware code is detected. Conducted experiments show that the precision of the proposed hybrid method is higher than a single DBN method. The evaluation of the proposed method is conducted on the KDDCUP'99 data set. The data are classified into the probe, U2R,

R2L, DoS, and Normal classes. Ref.²⁷ offers a hybrid approach based on the combination of spectral clustering and DNN for intrusion detection in sensor networks. In the first stage, the network features are clustered and divided into k subsets. In the second stage, by applying deep learning network to the subsets generated in the clustering phase, more understandable features are obtained and at the end, by using a test set, the attacks are detected.

In Ref.²⁸ based on the multilayer DBN, the DoS attacks detection method is proposed. DBN consists of numerous RBMs. Here in advance in the learning process, the training of the RBM is carried out. Then the trained features of RBM are used as an input data for learning RBM of the next layer of the DBN stack. The effectiveness of the DBN method is tested on the KDD CUP 1999 data set. The detection accuracy of the DBN model is better than the SVM and ANN methods. In Ref.,²⁹ a network anomaly detection method based on a semisupervised approach is proposed. To carry out this analysis, the discriminative RBM tool is used. In the semisupervised anomaly detection system, the classifier is trained according to the normal profile of the data, any deviation from such state is modeled as an anomaly signal. To evaluate the effectiveness of the proposed method, the experiments are provided on "real" KDD '99 data set. Here instead of the 41 features of the data set, 28 features only related to network traffic are selected. In the selection of hyperparameters, the neural network learning rate is taken equal to 0.1, the number of epochs at first is taken as 15, then as 10 and 5, and experiments are carried out.

In Ref.³⁰ on the basis of deep learning in the software defined networking (SDN) environment, flow-based anomaly detection approach is proposed. In this study, a DNN for the intrusion detection is constructed, and the model training is conducted on the NSL-KDD data set. Six features of the NSL-KDD data set are used. To evaluate the effectiveness of the model, the accuracy, precision, recall, and f-measure metrics are used. To detect attacks, the feature vector of the model consists of six parameters of the NSL-KDD data set, such as duration, protocol_type, src_bytes, dst_bytes, count, and srv_count. The difference of this work from others is that it uses simplex initial processing and feature extraction technologies in the SDN context. A comparative analysis of the proposed neural network method with the existing J48, Naive Bayes (NB), NB tree, random forest, random tree, multilayer perceptron, and SVM methods on the accuracy metrics is conducted. When tested

on all the features of the NSL-KDD data set, the deep learning method obtains very low result compared with the mentioned methods, but when it is tested on six features, the method in terms of accuracy metric gets the high result and composed of 75.75%.

In Ref.³¹ an approach to detect malicious zero-day Adobe Flash applications is proposed. The proposed approach consists of three stages: (1) extraction, (2) conversion, and (3) classification. In the extraction stage header, tags and action features are extracted. The conversion phase performs the conversion of the data generated at the extraction phase into (0, 1) values or matrices. The classification phase carries out training of deep learning models based on the data acquired at the conversion stage. If raw values are given, it is scaled to between 0 and 1 to reduce computational complexity. If the sequence of instructions or API calls are given, it is projected to a $N \times M$ matrix, where N is the number of sequences and M is the number of types. At the end, the results of deep learning methods are combined with the ensemble method. The proposed detection system is tested on randomly collected malicious Flash applications. In the data set, benign and malicious classes are used. The difference of our approach from existing works is that in this article by adding the additional seven layers between the hidden and visible layers of RBM, multilayered deep RBM architecture is formed. And this quite increases the DoS attacks detection efficiency.

In Ref.,³² to detect DDoS attacks in the web-application layer deep learning method is used. It is even more difficult to detect application-level attacks than DDoS attacks on TCP and IP layers because in such attacks attackers launch the legitimate HTTP requests from the genuine network computers. In this article for the extraction of the higher level features, the Stacked AutoEncoder is applied.

In Ref.³³ for the detection of application-level DDoS attacks using encrypted protocols, the anomaly detection approach based on computed statistics from network packets is applied. In the proposed scheme, negotiations between the web server and its clients are clustered and normal user behavior is established. With the help of Stacked AutoEncoder, the distribution of negotiations on these clusters is analyzed. Client negotiations deviated from the normal samples are classified as an anomaly.

In Ref.,¹² deep learning is used for the distributed detection of attacks in the IoT fog ecosystem. Experiments have shown that the distributed detection system outperforms the centralized system, and in terms of detection

accuracy, it is effective against shallow neural networks.¹³ For detecting DDoS attacks in the SDN environment, a deep learning-based defense system model is proposed. The model can learn samples from network traffic sequences and track network attack activities chronologically. Using defense system based on this model, it is possible to efficiently clean DDoS attack traffic on SDN.

Summarizing the mentioned problems, the list of deep learning-based network intrusion detection methods can be described as follows (Table 1).

Restricted BM

RBM is a stochastic network of neurons made up of two layers: the visible and the hidden layers. The visible layer represents the collected data, whereas the hidden layer tries to learn features from the visible layer aiming to represent a probabilistic distribution of the data.³⁴

The network is called restricted because the neurons in a layer have connections only to the neurons in the other layer. Connections between the layers are symmetric and bidirectional, allowing information transfer in both directions.

The RBM is graphically illustrated in Figure 1, where m is the number of neurons in the visible layer ($v_1, \dots, v_m$), n is the number of neurons in the hidden layer ($h_1, \dots, h_n$), a and b are *bias* vectors, and w is the weight matrix.

RBM is an energy-based model that uses a layer of hidden variables to model a probabilistic distribution over visible variables.³⁵ The visible units constitute the first layer and correspond to the components of

an observation, that is, one visible unit for each feature of an input pattern. The hidden units model dependencies between the components of observations, that is, the dependencies between the features. Energy-based model means that the probability distribution over the variables v and h is defined by an entropy function. This function is described by Equations (1) and (2),³⁶ first in its vectorial form in bold and second in its extended form:

$$E(v, h) = -h^T W v - a^T v - b^T h. \quad (1)$$

$$E(v, h) = - \sum_{i,j=1}^{m,n} v_i h_j w_{ij} - \sum_{i=1}^m a_i v_i - \sum_{j=1}^n b_j h_j. \quad (2)$$

Using the entropy function, it is possible to assign a probability for each pair of neurons of the network, one in the visible layer and the other in the hidden layer, giving a probabilistic distribution described in Ref.³⁵:

$$p(v, h) = \frac{e^{-E(v, h)}}{\sum_{v, h} e^{-E(v, h)}}. \quad (3)$$

The probability of the vector from the visible layer v is given as the sum of all probabilities of the vector from the hidden layer, described in Ref.³⁵:

$$p(v) = \frac{\sum_h e^{-E(v, h)}}{\sum_{v, h} e^{-E(v, h)}}. \quad (4)$$

Table 1. Deep learning-based distributed denial of service detection methods

Article	Feature extraction	Classifier	Goal	Data set	Classes
25	DBN	DBN and SVM	Attack detection	NSL-KDD	Normal, R2L, DoS, U2R, and Probing
10	—	Sparse auto-encoder and soft-max regression type self-taught learning	Network attack detection	NSL-KDD	Normal, R2L, DoS, U2R, and Probing
26	AutoEncoder	DBN	Malicious code detection	KDDCUP'99	Normal, R2L, DoS, U2R, and Probing
27	Spectral clustering	Combination of the spectral clustering and deep neural network	Intrusion detection in sensor networks	KDD-Cup99, NSL-KDD, sensor network data set	Normal, R2L, DoS, U2R, and Probing
28	Normalized	DBN	Reducing volume of the data based on feature extraction and DoS attacks detection	KDD CUP 1999	Normal, R2L, DoS, U2R, and Probing
29	Normalized	Discriminative RBM	Classification of network traffic in normal and anomaly classes	KDD '99	Normal and anomaly
31	Normalized	Deep feed-forward neural network, deep recurrent neural network	Malicious zero-day Adobe Flash software detection	Synthetic data	Benign and malicious

DBN, deep belief network; DoS, denial of service; RBM, restricted Boltzmann machine; U2R, user to root; R2L, remote to local; SVMs, support vector machines.

FIG. 1. Restricted Boltzmann machine (RBM).

Since RBMs do not have connections between neighboring neurons of the same layer, the events are independent. This facilitates the calculations of the conditional probabilities, which are³⁵:

$$p(h|v) = \prod_j p(h_j|v). \quad (5)$$

$$p(v|h) = \prod_i p(v_i|h). \quad (6)$$

The first versions of RBM were developed to solve problems with binary data. Considering that, Equations (7) and (8) could be generalized using the correct probability distribution for binary data, given by Ref.³⁵:

$$p(h_j = 1|v) = \text{sigm}\left(b_j + \sum_{i=1}^m v_i w_{ij}\right) \quad (7)$$

$$p(v_i = 1|h) = \text{sigm}\left(a_i + \sum_{j=1}^n h_j w_{ij}\right), \quad (8)$$

where $\text{sigm}(x)$ is the sigmoid function.

Using RBM for only binary data obviously limits its potential for problem solving. To maximize the accuracy of the problem, the form of the RBM is used that allows dealing with continuous data. The best known type of such RBM is the Gauss type RBM. In this RBM, the probability distribution of the visible layer is changed to a Gaussian distribution. This variation is called Gaussian–Bernoulli RBM.³⁷ The probable

distribution of Gaussian–Bernoulli type RBM is defined as follows:

$$p(h_j = 1|v) = \text{sigm}\left(b_j + \frac{1}{\sigma^2} \sum_{i=1}^m v_i w_{ij}\right) \quad (9)$$

$$p(v_i = 1|h) = N\left(a_i + \sum_{j=1}^n h_j w_{ij}, \sigma^2\right). \quad (10)$$

Learning in RBM

The training of an RBM involves in minimizing the negative log-likelihood given by

$$\Delta w_{ij} = e \frac{\partial \log p(V)}{\partial w_{ij}} = e \langle v_i h_j \rangle_d - \langle v_i h_j \rangle_m, \quad (11)$$

where e is the learning rate, $\langle \bullet \rangle_d$ and $\langle \bullet \rangle_m$ are used to represent the expected values of the data and the model, respectively.

The expectation can be obtained by Equations (7) and (8) for binary data, whereas Equations (9) and (10) give expectations for continuous data.

Data set description

For the conducting experiments in the classification process, the NSL-KDD data set is used, which is a fundamental data set for the evaluation of the effectiveness of the proposed approaches in the field of creating the network intrusion detection systems.

Each record of the NSL-KDD data set consists of 41 features (e.g., protocol type, service, and flag) and these

Table 2. Evaluating the effectiveness of the methods

	<i>Accuracy</i>	<i>f-measure</i>	<i>g-mean</i>	<i>Precision</i>	<i>Recall</i>	<i>Sensitivity</i>	<i>Specificity</i>	<i>TN</i>	<i>TP</i>	<i>Average value</i>
SVM radial basis	0.7161	0.7400	0.7173	0.6096	0.9416	0.9416	0.5464	0.5464	0.9416	0.7445
SVM (epsilon-SVR)	0.7269	0.7550	0.7251	0.6139	0.9804	0.9804	0.5363	0.5363	0.9804	0.6998
Decision tree	0.6744	0.7190	0.6620	0.5710	0.9705	0.9705	0.4516	0.4516	0.9705	0.7157
Gaussian-Bernoulli RBM	0.7323	0.7530	0.7348	0.6233	0.9509	0.9509	0.5678	0.5678	0.9509	0.7591

TN, true negative; TP, true positive.

records are labeled as normal and specific types of attacks. Here the attacks are divided into four classes:

- DoS: For example, Neptune, Smurf, Pod and Tear-drop.
- R2L: Unauthorized access from a remote machine to a local. For example, Guess-password, Ftp-write, Imap, and Phf.
- U2R: Unauthorized access to root privileges. For example, Buffer-overflow, Load-module, Perl, and Spy.
- Probing: For example, Port-sweep, IP-sweep, Nmap, and Satan.

Experiments and Discussion

We implemented three deep learning and three traditional machine learning architectures: Bernoulli-Bernoulli RBM, Gaussian-Bernoulli RBM, DBN, SVM (radial basis), SVM (epsilon-SVR), and decision tree.

In the experiments for detecting DoS attacks, RBM is used, which consists of 7 layers, 100 neurons, and 38 visible neurons and are conducted on NSL-KDD data set containing five classes (probe, U2R, R2L, DoS, and normal).

The number of neurons in the layers is taken as 100, 100, 100, 100, 100, 100, and 100 respectively. Column-wise normalization of data set is between the [0, 1] interval. To train the network, 5 epochs are used. The weights for the used 100 hidden neurons are selected randomly. For the training and testing of the network, NSL-KDDTrain+_20Percent data set is used. So, the number of rows of train data contains 25,194 samples and the test data contain 4508 samples. Experiments are carried out on 38 features of the data set. Three features that do not have a significant impact on the effectiveness of DoS attack detection are not taken into account in the classification process. The activation function of the model is sigmoid.

To evaluate the results of the experiments accuracy, f-measure, g-mean, precision, recall, sensitivity, specificity, true negative (TN) rate, true positive (TP) rate,

and average value indexes are used. In Table 2, comparative analysis of RBM results with classic classification algorithms on various metrics is described.

As seen in Table 2, the results of the RBM algorithm outperform the results of the other algorithms. Thus, the accuracy of SVM (radial basis) algorithm was 0.7161, SVM (epsilon-SVR) was 0.7269, decision tree was 0.6744, but in RBM algorithm, this measure achieved 0.7323. As seen from here, the DoS attacks detection efficiency of the RBM algorithm is better than that of other algorithms.

To determine the significance of the proposed methods it is desirable to provide the statistical tests.^{38,39} Here RBM algorithm is launched on the NSL-KDD data set 22 times. The effectiveness of the method is evaluated on the basis of accuracy, f-measure, g-mean, precision, recall, sensitivity, specificity, TN rate, and TP rate metrics. For detailed information, see Supplementary Tables S1–S3 given in the Supplementary Data (Supplementary Data are available online at www.liebertpub.com/big).

FIG. 2. Boxplot representation of the average values of various metrics of deep learning methods for denial of service attack detection.

FIG. 3. Comparison of the methods on NSL-KDD data set.

To evaluate the robustness of the Bernoulli-Bernoulli RBM, Gaussian-Bernoulli RBM, and the DBN algorithms, the program in the Matlab environment is launched 22 times on the NSL-KDD data set and a boxplot representation based on the average value of results obtained on different metrics is created. Boxplot representation of the average values of various metrics for deep learning methods of DoS attack detection is shown in Figure 2.

FIG. 4. Accuracy dynamics of the RBM by number of epochs from two to five.

As seen from Figure 2, Gaussian-Bernoulli-type RBM gives better results than the other methods and because the deviation of this method is very small, the boxplot representation of this method is too tight. For better demonstration of the results, Figure 3 visually illustrates the comparison of methods. Notice that in these figures, we subtract the recall, sensitivity, and TP rate values from all the methods, thus the difference can be observed more clearly.

The worst, mean, and best case values obtained from 22 times running of each algorithm are given in Table 3. From Table 3, it is obvious that the worst result obtained by Gaussian-Bernoulli RBM is even better than the best results obtained by the other methods. DNNs are models with a large number of parameters and these parameters should be regulated during the training process. When learning such models, a large volume of data should be used. However, the size of the NSL-KDD data set used here is small. Training

Table 3. Comparison of the results of the deep learning methods according to the accuracy metric

Method	Worst	Mean	Best
Bernoulli-Bernoulli RBM	0.6422	0.7255	0.6828
Gaussian-Bernoulli RBM	0.6894	0.7591	0.7323
DBN	0.4343	0.5299	0.4809

FIG. 5. Detection rate dynamics of the RBM over various metrics by number of epochs from two to five.

the deep learning model on small data set with big iteration number leads to overfitting of the network⁴⁰ and this reduces the effectiveness of the model. To eliminate this problem, in this article, the number of iterations is taken small.

In this article, the experiments related to epoch number are also conducted. According to the number of epochs, the detection accuracy of DoS attacks by training the Gaussian–Bernoulli type RBM is described in Supplementary Table S4.

As given in Supplementary Table S4, the RBM's DoS attack detection accuracy gradually decreases as the number of epochs increases here. But the RBM's DoS attack detection accuracy gradually increases as the number of epochs increases from 2 to 5 (Supplementary Table S5) (see Supplementary Data).

For better demonstration of the results in Supplementary Table S5, Figure 4 visually shows the RBM DoS attack detection accuracy dynamics.

As seen from Figures 5 and 6, in fact, when the number of iterations over the data is taken less, the model detects DoS attacks with high precision (Fig. 5), but as the number of iterations increases, the accuracy of the model for DoS attacks detection gradually falls down (Fig. 6).

Since the TN rate has the same value as the specificity metric, and TN rate and sensitivity metric have the same values as the recall, the TN rate, TP rate, and sensitivity are not included in Figure 6.

FIG. 6. Accuracy dynamics of the RBM by number of epochs from 5 to 65.

Table 4. Restricted Boltzmann machine denial of service attack detection statistics based on the number of epochs

			No. of predicted points						
Classes		No. of real points	5 epochs	10 epochs	15 epochs	20 epochs	25 epochs	30 epochs	35 epochs
Normal	1	1935	2908	2780	2885	3009	1356	3227	2874
DoS	2	1526	1077	1555	844	895	1323	918	1027
U2R	3	46	0	0	0	0	0	0	0
R2L	4	513	0	0	429	520	1638	228	594
Probe	5	488	523	173	350	84	191	135	13
Total		4508	4508	4508	4508	4508	4508	4508	4508

As seen in Figure 5, if we increase the number of iterations from two to five, the DoS attack detection accuracy of the Gaussian–Bernoulli RBM gradually increases. But if we increase this value starting from 5 to 65, the detection accuracy of the algorithm gradually decreases. Visual representation of this case is depicted in Figure 6.

Attacks on the test data are divided into four categories: probe (e.g., IP sweep, vulnerability scanning), DoS (e.g., mail bomb, UDP storm), U2R (e.g., buffer overflow attacks, rootkits) and R2L (e.g., password guessing, worm attack).

Since the number of data in the probe and DoS classes is much larger than U2R and R2L, there is always a decrease in the attacks detection effectiveness of the algorithm.³⁶ This condition is associated with the inability of the algorithm to handle the U2R and R2L classes well. Therefore researchers try to study the attack classes separately. Below is the number of points in the DoS, U2R, R2L, probe (attacks), and normal classes, and the ability of the Gaussian–Bernoulli RBM method for detecting these points in the different iterations values is given (Table 4).

As the number of iterations increases in the Gaussian–Bernoulli RBM method, the points from the R2L class are detected, but in this case, the detection accuracy of the method decreases. This is due to the fact that there are many false detections allowed. In the experiments, the points from probe, DoS, and normal classes are detected with high accuracy; however, the other U2R and R2L attack classes give the worst result compared with the other classes. It should be noted that the U2R and R2L classes are in general less visible in data and a lower accuracy should, therefore, be expected.

Conclusions

In this article, the discriminative RBM was applied to the network attacks detection problem. The results of the experiments on the NSL-KDD data set showed that the proposed multilayer deep Gaussian–Bernoulli RBM method gave better results in detecting attacks

than the other Bernoulli–Bernoulli RBM and DBN type deep learning methods. This method also outperforms the results of SVM, radial basis, SVM (epsilon-SVR), decision tree type machine learning methods too.

We have shown that our model significantly improves the detection accuracy on DoS attack detection tasks compared with previous work. In future work we also plan on experimenting with LSTM decoders as well as deep and bidirectional LSTM encoders.

Acknowledgment

This work was supported by the Science Development Foundation under the President of the Republic of Azerbaijan—Grant No. EIF-KETPL-2-2015-1(25)-56/05/1.

Author Disclosure Statement

No competing financial interests exist.

References

- Mansfield-Devine S. DDoS goes mainstream: How headline-grabbing attacks could make this threat an organisation's biggest nightmare. *Netw Secur.* 2016;11:7–13.
- Nazario J. Politically motivated denial of service attacks. In: Czosseck C, Geers K, eds. *The virtual battlefield perspectives on cyber warfare.* Amsterdam: los Press, 2009, pp. 163–181.
- Kaspersky DDoS attacks in Q3 2017. Kaspersky. Available online at <https://securelist.com/ddos-attacks-in-q3-2017/83041/> (last accessed March 22, 2018).
- Kolias C, Kambourakis G, Stavrou A, Voas J. DDoS in the IoT: Mirai and other botnets. *Computer.* 2017;50:80–84.
- Kobojek P, Saeed K. Application of recurrent neural networks for user verification based on keystroke dynamics. *J Telecommun Inf Technol.* 2016;3:60–70.
- Chandola V, Banerjee A, Kumar V. Anomaly detection: A survey. *J ACM Comput Surv (CSUR).* 2009;41:1–58.
- Garcia-Teodoro P, Diaz-Verdejo J, Macia-Fernandez G, Vazquez E. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Comput Secur.* 2009;28:18–28.
- Aminanto ME, Kim K. Deep learning in intrusion detection system: An overview. In: *Proceedings of the International Research Conference on Engineering and Technology, Jakarta, Indonesia, 2016*, pp. 1–12. Available online at <https://pdfs.semanticscholar.org/c0fa/578c1fae002e02834806a576d811002cb4a4.pdf> (last accessed February 20, 2018).
- Wang Z. The applications of deep learning on traffic identification. *Blackhat USA, 2015*, pp. 1–10.
- Niyaz Q, Sun W, Javaid AY, Alam M. A Deep learning approach for network intrusion detection system. In: *Proceedings of the 9th EAI International*

- Conference on Bio-inspired Information and Communications Technologies (BICT), New York, 2016, pp. 21–26.
11. Imamverdiyev YN, Sukhostat LV. Network traffic anomalies detection based on informative features. *Radio Electron Comput Sci Control*. 2017;3:113–120.
 12. Diro AA, Chilamkurti N. Distributed attack detection scheme using deep learning approach for Internet of Things. *Future Gener Comput Syst*. 2018;82:761–768.
 13. Li C, Wu Y, Yuan X, et al. Detection and defense of DDoS attack-based on deep learning in OpenFlow-based SDN. *Int J Commun Syst*. 2018:1–15.
 14. Mirkovic J, Reiher P. A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM SIGCOMM Comput Commun Rev*. 2004;34:39–53.
 15. Peng T, Leckie C, Ramamohanarao K. Survey of network-based defense mechanisms countering the DoS and DDoS problems. *ACM Comput Surv*. 2007;39:1–42.
 16. Loukas G, Oke G. Protection against denial of service attacks: A survey. *Comput J*. 2010;53:1020–1037.
 17. Bhuyan MH, Kashyap HJ, Bhattacharyya DK, Kalita JK. Detecting distributed denial of service attacks: Methods, tools and future directions. *Comput J*. 2013;57:537–556.
 18. Tama BA, Rhee KH. Data mining techniques in DoS/DDoS attack detection: A literature review. In: *Proceedings of the 3rd International Conference on Computer Applications and Information Processing Technology (CAIPT 2015)*, Yangon, Myanmar, 2015, pp. 1–4.
 19. Girma A, Garuba M, Goel R. Advanced machine language approach to detect DDoS attack using DBSCAN clustering technology with entropy. In: *Latifi S, ed. Information Technology - New Generations. Advances in Intelligent Systems and Computing*, 2018, vol. 558. Cham, Switzerland: Springer, pp. 125–131.
 20. Deng L. A tutorial survey of architectures, algorithms, and applications for deep learning, APSIPA. *Trans Signal Inf Process*. 2014;3:1–29.
 21. Vincent P, Larochelle H, Lajoie I, et al. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J Mach Learn Res*. 2010;11:3371–3408.
 22. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput*. 1997;9:1735–1780.
 23. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. *Adv Neural Inf Process Syst*. 2012;25:1097–1105.
 24. Deng L, Yu D. Deep learning: Methods and applications. *Found Trends Signal Process*. 2014;7:197–387.
 25. Salama MA, Eid HF, Ramadan RA, et al. Hybrid Intelligent Intrusion Detection Scheme. In: *Gaspar-Cunha A, Takahashi R, Schaefer G, Costa L, eds. Soft Computing in Industrial Applications. Advances in Intelligent and Soft Computing*, 2011, vol. 96. Berlin, Heidelberg, Germany: Springer, pp. 293–303.
 26. Li Y, Ma R, Jiao R. A hybrid malicious code detection method based on Deep Learning. *Int J Secur Appl*. 2015;9:205–216.
 27. Ma T, Wang F, Cheng J, et al. A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks. *Sensors*. 2016;16:1–23.
 28. Gao N, Gao L, Gao Q. An intrusion detection model based on deep belief networks. In: *Proceedings of the Second International Conference on Advanced Cloud and Big Data (CBD)*, Huangshan, China, 2014; pp. 247–252.
 29. Fiore U, Palmieri F, Castiglione A, Santis AD. Network anomaly detection with the restricted Boltzmann machine. *Neurocomputing*. 2013;122:13–23.
 30. Tang TA, Mhamdi L, McLernon D, et al. Deep Learning approach for network intrusion detection in software defined networking. In: *Proceedings of the IEEE International Conference on Wireless Networks and Mobile Communications (WINCOM)*, Fez, Morocco, 2016, pp. 258–263.
 31. Jung W, Kim S, Choi S. Poster: Deep learning for zero-day flash malware detection. In: *Proceedings of the 36th IEEE Symposium on Security and Privacy*, San Jose, California, 2015. Available online at <https://www.ieee-security.org/TC/SP2015/program-posters.html> (last accessed January 10, 2018).
 32. Yadav S, Subramanian S. Detection of application layer DDoS attack by feature learning using stacked autoencoder. In: *Proceedings of the International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT)*, New Delhi, India, 2016, pp. 361–366.
 33. Zolotukhin M, Hämmäläinen T, Kokkonen T, Siltanen J. Increasing web service availability by detecting application-layer DDoS attacks in encrypted traffic. In: *Proceedings of the 23rd International Conference on Telecommunications (ICT)*, Thessaloniki, Greece, 2016, pp. 1–6; 2016.
 34. Memisevic R, Hinton GE. Learning to represent spatial transformations with factored higher-order Boltzmann machines. *Neural Comput*. 2010; 22:1473–1492.
 35. Larochelle H, Bengio Y. Classification using discriminative restricted Boltzmann machines. In: *Proceedings of the 25th ACM International Conference on Machine Learning*, Helsinki, Finland, 2008, pp. 536–543.
 36. Saied A, Overill RE, Radzik T. Artificial neural networks in the detection of known and unknown DDoS attacks: Proof-of concept. *Commun Comput Inf Sci*. 2014;430:300–320.
 37. Hinton GE, Salakhutdinov R. Reducing the dimensionality of data with neural networks. *Science*. 2006;313:504–507.
 38. Alguliyev RM, Aliguliyev RM, Isazade NR. An unsupervised approach to generating generic summaries of documents. *Appl Soft Comput*. 2015; 34:236–250.
 39. Alguliyev RM, Aliguliyev RM, Mehdiyev CA. pSum-SaDE: A modified p-median problem and selfadaptive differential evolution algorithm for text summarization. *Appl Comp Intell Soft Comput*. 2011; 2011:13.
 40. Xiao L, Shao Z, Liu G. K-means algorithm based on Particle Swarm Optimization algorithm for anomaly intrusion detection. In: *The sixth world congress on intelligent control and automation*, Dalian, China, 2006, pp. 5854–5854.

Cite this article as: Imamverdiyev Y, Abdullayeva F (2018) Deep learning method for denial of service attack detection based on restricted Boltzmann machine. *Big Data* 6:2, 159–169, DOI: 10.1089/big.2018.0023.

Abbreviations Used

BM	= Boltzmann machine
CNN	= convolutional neural network
DBM	= deep BM
DBN	= deep belief network
DDoS	= distributed DoS
DNN	= deep neural network
DoS	= denial of service
NB	= Naive Bayes
R2L	= remote to local
RBM	= restricted Boltzmann machine
RNN	= recurrent neural networks
SDN	= software defined networking
SVM	= support vector machine
TN	= true negative
TP	= true positive
U2R	= user to root