http://dx.doi.org/10.14236/ewic/icscsr19.16

Deep Learning Techniques for Cyber Security Intrusion Detection : A Detailed Analysis

Mohamed Amine Ferrag
Department of Computer Science
Guelma University, Algeria
ferrag.mohamedamine@univ-guelma.dz

Leandros Maglaras
School of Computer Science and Informatics
De Montfort University, UK
leandros.maglaras@dmu.ac.uk

Helge Janicke
School of Computer Science and Informatics
De Montfort University, UK
heljanic@dmu.ac.uk

Richard Smith
School of Computer Science and Informatics
De Montfort University, UK
rgs@dmu.ac.uk

In this study, we present a detailed analysis of deep learning techniques for intrusion detection. Specifically, we analyze seven deep learning models, including, deep neural networks, recurrent neural networks, convolutional neural networks, restricted Boltzmann machine, deep belief networks, deep Boltzmann machines, and deep autoencoders. For each deep learning model, we study the performance of the model in binary classification and multiclass classification. We use the CSE-CIC-IDS 2018 dataset and TensorFlow system as the benchmark dataset and software library in intrusion detection experiments. In addition, we use the most important performance indicators, namely, accuracy, detection rate, and false alarm rate for evaluating the efficiency of several methods.

Deep Learning, intrusion detection, Cyber Security, machine learning

1. INTRODUCTION

The major target of cyber attacks is a country's Critical National Infrastructure (CNI) such as ports, hospitals, water, gas or electricity producers, which use and rely upon Supervisory Control and Data Acquisitions(SCADA) and Industrial Control Systems (ICS) to manage their production. Protection of CNIs becomes an essential issue to be considered. Generally, available protective measures are classified according to legal, technical, organizational, capacity building, and cooperation aspects. Except from regulations and policies that may be used to tackle cyber attacks to CNIs specific practical measures need to be taken in order for these regulations to be effective Maglaras et al. (2018).

Along with other preventive security mechanisms, such as access control and authentication, intrusion detection systems (IDS) are deployed as a second line of defense Ahmim et al. (2018). IDS based on some specific rules or patterns of normal behavior of the system can distinguish between normal and malicious actions Ahmim et al. (2018). The necessity of cyber physical security is rising and traditional methods may not be effective anymore Stewart et al. (2017). According to Dewa and Maglaras (2016), data mining and its core feature which is knowledge discovery can significantly help in creating Data mining based IDSs that can achieve higher accuracy to novel types of intrusion and demonstrate more robust behaviour compared to traditional IDSs.

Moreover, many researchers struggle to find comprehensive and valid datasets to test and evaluate their proposed techniques and having a suitable dataset is a significant challenge itself. In order to test the efficiceny of such mechanisms, reliable datasets that contain both bening and several attacks, meets real world criteria and that is publicly avaialble is needed Sharafaldin et al. (2018).

Our contributions in this work are:

- We review the deep learning techniques papers applied to cyber security intrusion detection.
- We present all datasets used by the deep learning techniques papers applied to cyber security intrusion detection.
- We analyze seven deep learning techniques according to two models,

namely, deep discriminative models and generative/unsupervised models.

- We study the performance of each deep learning model in binary classification and multiclass classification using CSE-CIC-IDS 2018 dataset and TensorFlow system.

The rest of this paper is organized as follows. Section 2 gives the intrusion detection systems based on deep learning techniques. In Section 3, we present the different datasets used by deep learning techniques papers applied to intrusion detection. In Section 4, we present seven deep learning approaches. In Section 5, we study the performance of each deep learning technique in binary classification and multiclass classification. Lastly, Section 6 presents conclusions.

2. A REVIEW OF INTRUSION DETECTION SYSTEMS BASED ON DEEP LEARNING TECHNIQUES

This section describes the intrusion detection systems based on deep learning techniques.

Zhou et al. (2018) proposed a system that uses a deep neural network model to help classify cyber-attacks. Specifically, the system uses three phases, namely, data acquisition (DAQ), data pre-processing, and deep neural network classification. The system achieves an accuracy of 0.963 SVM model with learning rate 0.01, training epochs 10, and input units 86. The results show outperform slightly compared to the following traditional machine learning algorithms: random forest, linear regression, and k-nearest neighborhood.

Tang et al. (2016) describe an IDS system that employs deep learning technique in software-defined networking. The proposed IDS system is implemented in the SDN controller which can monitor all the OpenFlow switches and request all network statistic. The study used NSL-KDD dataset under 2-class classification (normal and anomaly class), where the dataset consisted of four categories, namely, DoS attacks, R2L attacks, U2R attacks, Probe attacks. The experimental results reported that the learning rate of 0.001 performs better than others with the highest receiver operating characteristic curve (AUC).

The framework proposed by Kim et al. (2016) use the KDD Cup 1999 dataset to perform long short term memory architecture to a recurrent neural network for intrusion detection. The study used (41 features) as an input vector (4 attacks and 1 nonattack) as the output vector. They used a time step size 100, batch size 50, and epoch 500. The attack detection performance is reported as 98.8% among the total attack instances.

Integrating a recurrent neural network in an IDS system was attempted by Yin et al. (2017) for supervised classification learning. The study used NSL-KDD dataset as benchmark dataset under three performance indicators, including, accuracy, true positive rate, and false positive rate. The anomaly detection performance is reported as higher accuracy when there are 80 hidden nodes and the learning rate is 0.1. The paper also states the benefits of a recurrent neural network for intrusion detection.

In another study, Tang et al. (2018) suggested a gated recurrent unit recurrent neural network for intrusion detection in software-defined networking. The paper states a detection rate of 89% using a minimum number of features. The NSL-KDD dataset is used in the network performance with four evaluation metrics, including, precision, recall, F-measure, and accuracy.

A multi-channel intelligent attack detection system that uses long short term memory recurrent neural networks is described by Jiang et al. (2018). The NSL-KDD dataset is used to evaluate the performance of the proposed intelligent attack detection system. The performance of the long short term memory recurrent neural network is reported as 99.23% detection rate with a false alarm rate of 9.86% and an accuracy of 98.94%.

The convolutional neural networks were used by Basumallik et al. (2019) for packet-data anomaly detection in phasor measurement units-based state estimator. They use a convolutional neural network-based data filter in order to extract event signatures (features) from phasor measurement units. The IEEE-30 bus and IEEE-118 bus system are used as the phasor measurement unit buses. The study states a probability of 0.5 with 512 neurons at a fully connected layer and a 98.67% accuracy. The authors claim that convolutional neural network-based filter has a superior performance over other filters, including, recurrent neural network, long short-term memory, support vector machine, bagged, and boosted.

The framework developed by Fu et al. (2016) uses a convolutional neural network in order to capture the intrinsic patterns of fraud behaviors, especially for credit card fraud detection. Zhang et al. (2018) employed the convolutional neural network and used the commercial bank B2C online transaction data for training and testing. The data of one month were divided into training sets and test sets. The study states a precision rate of 91% and the recall rate of

Table 1: *Deep learning techniques for intrusion detection and dataset they use*

Deep Learning Technique	IDS	Dataset Used	No. of times cited (as of 30/05/2019)
Deep neural network	Tang et al. (2016)	NSL-KDD dataset	110
Deep neural network	Potluri and Diedrich (2016)	NSL-KDD dataset	37
Deep neural network	Kang and Kang (2016)	Vehicular network communication	137
Deep neural network	Zhou et al. (2018)	4 types of attacks (DOS, R2L, U2R, and PROBING)	0
Deep neural network	Feng et al. (2019)	KDD Cup 1999 dataset	1
Deep neural network	Zhang et al. (2019)	KDD Cup 1999 dataset	0
Deep neural network	Roy et al. (2017)	KDD Cup 1999 dataset	23
Feed forward deep neural network	Kasongo and Sun (2019)	NSL-KDD dataset	0
Recurrent neural network	Kim et al. (2016)	KDD Cup 1999 dataset	86
Recurrent neural network	Yin et al. (2017)	NSL-KDD dataset	100
Recurrent neural network	Tang et al. (2018)	NSL-KDD dataset	9
Recurrent neural network	Jiang et al. (2018)	NSL-KDD dataset	22
Convolutional neural network	Basumallik et al. (2019)	IEEE-30 bus and IEEE-118 bus	1
Convolutional neural network	Fu et al. (2016)	Credit card transaction data	47
Convolutional neural network	Zhang et al. (2018)	Commercial bank B2 online transaction data	3
Convolutional neural network	Feng et al. (2019)	KDD Cup 1999 dataset	1
Convolutional autoencoder	Yu et al. (2017)	Contagio-CTU-UNB dataset	17
Restricted Boltzmann machine	Alrawashdeh and Purdy (2016)	KDD Cup 1999 dataset	35
Restricted Boltzmann machine	Aldwairi et al. (2018)	ISCX dataset	4
Restricted Boltzmann machine	Fiore et al. (2013)	KDD Cup 1999 dataset	176
Restricted Boltzmann machine	Salama et al. (2011)	NSL-KDD dataset	96
Restricted Boltzmann machine	Gao et al. (2014)	KDD Cup 1999 dataset	69
Restricted Boltzmann machine	Alom et al. (2015)	NSL-KDD dataset	59
Restricted Boltzmann machine	Yang et al. (2017)	Real online network traffic	16
Restricted Boltzmann machine	Otoum et al. (2019)	KDD Cup 1999 dataset	3
Deep belief network	Zhao et al. (2017)	KDD Cup 1999 dataset	13
Deep auto-encoder	Shone et al. (2018)	NSL-KDD dataset	66
Deep auto-encoder	Khan et al. (2019)	UNSW-NB15 dataset	0
Deep auto-encoder	Papamartzivanos et al. (2019)	NSL-KDD dataset	1
Denoising auto-encoder	Abusitta et al. (2019)	KDD Cup 1999 dataset	1

94%. These results are increased by 26% and 2%, respectively, compared with the work proposed by Fu et al. (2016).

The restricted Boltzmann machine was used for intrusion detection by Fiore et al. (2013). They use a discriminative restricted Boltzmann machine in order to combine the expressive power of generative models with good classification. The KDD Cup 1999 dataset was used, with a set of 41 features describing various aspects. The study used only part of the total training data, namely, those containing ’normal’ connections (97,278 instances).

Salama et al. (2011) combine the restricted Boltzmann machine and support vector machine for intrusion detection. The NSL-KDD dataset was used, which the training set contains a total of 22 training attack types, with an additional 17 types in the testing set. The study states that this combination shows a higher percentage of classification than support vector machine.

3. PUBLIC DATASETS

Table 1 lists the representative deep learning techniques papers applied to intrusion detection that were reviewed, including the number of times they have been cited and the dataset used. We can observe that most papers use four datasets, including, KDD Cup 1999 dataset, NSL-KDD dataset, and UNSW-NB15 dataset. However, these datasets are outdated and of very limited practical value for a modern IDS. Note that there are others IDSs dataset evaluation framework (e.g., DEFCON, CAIDAs, LBNL, CDX, KYOTO, TWENTE, UMASS, and ADFA2013), which are not yet used by deep learning techniques. In our work, we use a new real traffic data set "CSE-CIC-IDS2018[1]" developed by the Communications Security Establishment (CSE) & the Canadian Institute for Cybersecurity (CIC).

[1] https://registry.opendata.aws/cse-cic-ids2018/

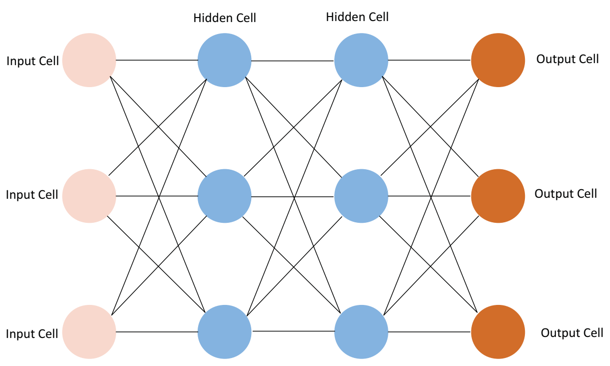


Figure 1: *Deep neural network.*

4. DEEP LEARNING APPROACHES

According to Deng and Yu (2014), deep learning techniques can be classified into two models, namely, 1) deep discriminative models and 2) generative/unsupervised models. The deep discriminative models include deep neural networks (DNNs), recurrent neural networks (RNNs), convolutional neural networks (CNNs). The generative/unsupervised models include restricted Boltzmann machine (RBMs), deep belief networks (DBNs), deep Boltzmann machines (DBMs), and Deep autoencoders (DA). Depending on how these Deep learning techniques are intended for use, these techniques can be categorized into three major classes, including, 1) Deep networks for unsupervised or generative learning; 2) Deep networks for supervised learning; and 3) Hybrid deep networks.

4.1. Deep discriminative models

4.1.1. Deep neural networks (DNNs)

Deep Neural Network is multilayer perceptrons (MLP) with a number of layers superior to three. MLP is a class of feed forward artificial neural network, which is defined by the n layers that compose it and succeed each other, as presented in Figure 1.

The layer $M \in [1, N]$ of a DNN network is defined by $D_M(a_M, \alpha_M, n_M)$. $a_M \in \mathbb{N}$ is the number of neurons in the layer. $\alpha_M : \mathbb{R}^{a_{M-1}} \to \mathbb{R}^{a_M}$ is the affine transformation defined by the matrix W_M and the vector b_M. $n_M : \mathbb{R}^{a_M} \to \mathbb{R}^{a_M}$ is the transfer function of the layer M. The matrix W_M is called the weight matrix between the layer $M-1$ and the layer M. The vector b_M is called the bias vector of the layer M. Refer to Figure 1 and Liu et al. (2017), deep neural network algorithm based on MLP is described as Algorithm 1.

Algorithm 1 DNN network based on MLP

Choose a learning pair $(x,\ c)$;
$h_0 = x$;
for $M = 1$ `to` N **do**
    $g_M = n_M\,(h_{M-1}) = W_M \times h_{M-1} + b_M$;
    $h_M = \alpha_M(g_M)$
end for

4.1.2. Recurrent neural networks (RNNs)

A recurrent neural network is a neuron network, which the connection graph contains at least one cycle. There are many types of RNNs such as Elman networks proposed by Elman (1990), Jordan networks proposed by Jordan (1997) and Echo State networks proposed by Jaeger and Haas (2004). Currently, RNN based on Long Short-Term Memory (LSTM) is the most used. The RNN is defined by adding an interconnection matrix $VW_M \in \mathbb{R}^{a_M \times a_M}$ to the layer $M \in [1, N]$ in order to obtain a layer M' of the recurrent network. Refer to Figure 2 and Gelly and Gauvain (2017), recurrent neural network algorithm is described as Algorithm 2.

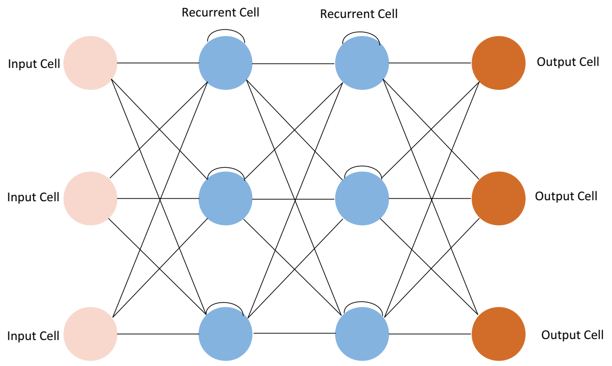


Figure 2: *Recurrent neural network*

Algorithm 2 Recurrent neural network

Choose a learning pair $(x(t),\ c(t))$;
$h_0\,(t) = x\,(t)\,,\ \forall t \in [1, t_f]$;
for $M = 1$ `to` N **do**
    for $t = 1$ `to` t_f **do**
        $g_M\,(t) = W_M \times h_{M-1}\,(t) + VW_M \times h_M\,(t-1) + b_M$;
        $h_M\,(t) = \alpha_M(g_M\,(t))$;
    end for
end for

4.1.3. Convolutional neural networks (CNNs)

A convolutional neural network is defined as a neural network that extracts features at a higher resolution, and then convert them into more complex features at a coarser resolution, as presented in Figure 3. There are many types of CNNs such as ZFNet proposed by Zeiler and Fergus (2014), GoogleNet proposed by Szegedy et al. (2015), and ResNet proposed by He et al. (2016). Therefore, CNN is based on three types of layers, including, convolutional, pooling, and fully-connected layers. Refer to Gu et al. (2018), the feature value at location (x, y) in the k-th feature map

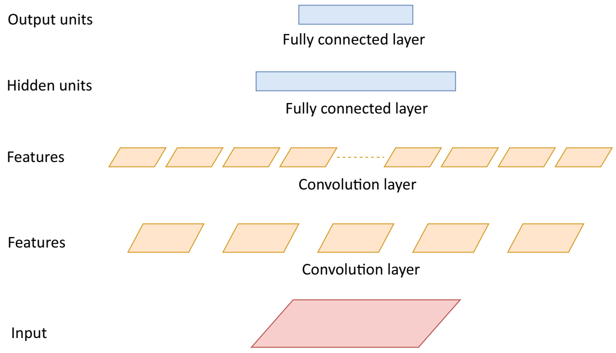


Figure 3: *Convolutional neural network*

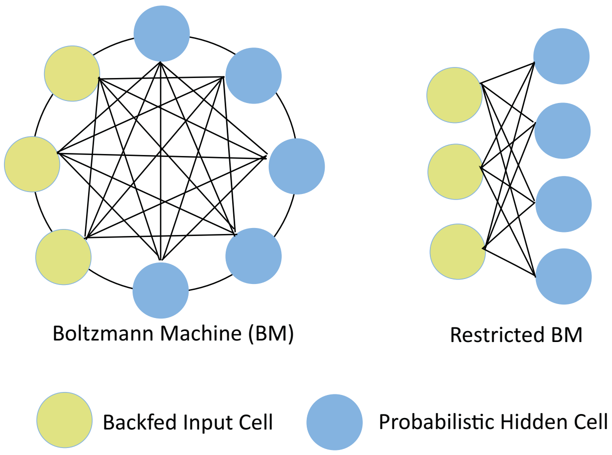


Figure 4: *Restricted Boltzmann machine*

of M-th layer can be calculated as follow:

$$feature^{M}_{x,y,k} = W^{M\,T}_{k} X^{M}_{x,y} + b^{M}_{k} \tag{1}$$

where $X^{M}_{x,y}$ is the input patch centered at location (x, y), W^{M}_{k} is the weight vector of the k-th filter, and b^{M}_{k} is bias term of the M-th layer.

The activation value $activ^{M}_{x,y,k}$ and pooling value $pool^{M}_{x,y,k}$ of convolution feature $feature^{M}_{x,y,k}$ can be calculated as follow

$$activ^{M}_{x,y,k} = activation(feature^{M}_{x,y,k}) \tag{2}$$

$$pool^{M}_{x,y,k} = pooling\left(feature^{M}_{a,c,k}\right), \ \forall(a,c) \in \mathcal{R}_{x,y} \tag{3}$$

where $\mathcal{R}_{x,y}$ is a local neighbourhood around location at location (x, y). The nonlinear activation function $activation(\cdot)$ are be ReLU, sigmoid, and tanh. The pooling operation $pooling(\cdot)$ are average pooling and max pooling.

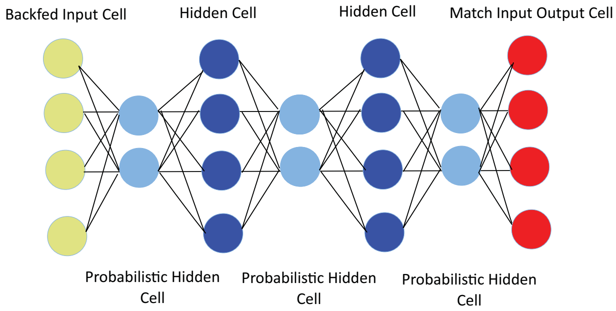


Figure 5: *Deep belief network.*

4.2. Generative/unsupervised models

4.2.1. Restricted Boltzmann machine (RBMs)
An RBM is an undirected graphic model $G = \{W_{ij}, b_i, c_j\}$, as presented in Figure 4. There are two layers, including, the hidden layer and the visible layer. The two layers are fully connected through a set of weights W_{ij} and $\{b_i, c_j\}$. Note that there is no connection between the units of the same layer. Refer to Fischer and Igel (2012), the configuration of the connections between the visible units and the hidden units has an energy function, which can be defined as follow:

$$En\,(V,H,G) = -\sum_{i}\sum_{j} V_j H_j W_{ij} - \sum_{i\in V} b_i V_i - \sum_{j\in H} c_j H_j \tag{4}$$

Based on this energy function, the probability of each joint configuration can be calculated according to the Gibbs distribution as follow:

$$Prob\,(V,H,G) = -\frac{1}{Z(G)} e^{-En(V,H,G)} \tag{5}$$

where Z is the partition function, which can be calculated as follow:

$$Z\,(G) = \sum_{V\in\mathcal{V}}\sum_{H\in\mathcal{V}} e^{-En(V,H,G)} \tag{6}$$

where curved letters $\mathcal{V}$ and $\mathcal{V}$ are used to denote the space of the visible and hidden units, respectively.

4.2.2. Deep belief networks (DBNs)
A DBN is multi-layer belief network, where each layer is Restricted Boltzmann Machine, as presented in Figure 5. The DBN contains a layer of visible units and a layer of hidden units. The layer of visible units represent the data. The layer of hidden units learns to represent features. Refer to Hinton (2009), the

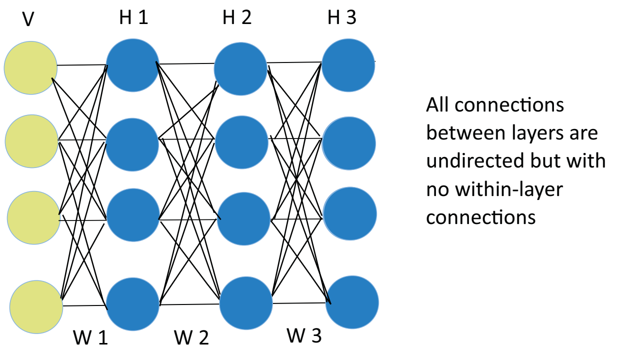


Figure 6: *Deep Boltzmann machine.*

probability of generating a visible vector, V , can be calculated as:

$$Prob(V) = \sum_H Prob\,(H \mid W)\,Prob(V|H, W) \qquad (7)$$

where $Prob\,(H \mid W)$ is the prior distribution over hidden vectors.

4.2.3. Deep Boltzmann machines (DBMs)
A DBM is a network of symmetrically coupled stochastic binary units, which contains a set of visible units and a sequence of layers of hidden units, as presented in Figure 6. Refer to Salakhutdinov and Larochelle (2010), a DBM with three hidden layers can be defined by the energy of the state $\{V, H\}$ as:

$$En\,(V, H, G) == -V^T W^1 H^1 - V^1 W^2 H^2 - V^2 W^3 H^3 \qquad (8)$$

where $H = \{H^1, H^2, H^3\}$ are the set of hidden units, and $G = \{W^1, W^2, W^3\}$ are the model parameters. The probability that the model assigns to a visible vector V can be defined as:

$$Prob\,(V, G) = \frac{1}{Z(G)} \sum_H e^{-En(V,H,G)} \qquad (9)$$

4.2.4. Deep auto encoders (DA)
An autoencoder consists of two parts, the encoder and the decoder, as presented in Figure 7. Refer to Vincent et al. (2010), these two parts can be defined as follow:

$$encoder_G\,(x) = s(Wx + b) \qquad (10)$$

$$decoder_{G'}\,(y) = s\,(W'y + b') \qquad (11)$$

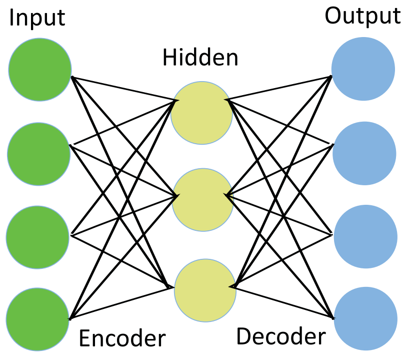


Figure 7: *Deep auto encoder.*

Table 2: *Attack Types in CSE-CIC-IDS2018 dataset*

Category	Attack Type	Flow Count	Training	Test
Brute-force	SSH-Bruteforce	230	184	46
	FTP-BruteForce	611	489	122
Web attack	Brute Force -XSS	187589	7504	1876
	Brute Force -Web	193360	15469	3867
	SQL Injection	87	70	17
DoS attack	DoS attacks-Hulk	466664	18667	4667
	DoS attacks-SlowHTTPTest	139890	55956	13989
	DoS attacks-Slowloris	10990	4396	1099
	DoS attacks-GoldenEye	41508	16603	4151
DDoS attack	DDOS attack-HOIC	686012	27441	6860
	DDOS attack-LOIC-UDP	1730	1384	346
	DDOS attack-LOIC-HTTP	576191	23048	5762
Botnet	Bot	286191	11448	2862
Infilteration	Infilteration	161934	6478	1620
Benign	/	12697719	50791	12698
Total	/	15450706	231127	57782

where $G = \{W, b\}$; $G' = \{W', b'\}$; W is a $d' \times d$ weight matrix; x is an input vector; y is the hidden representation; b is an offset vector of dimensionality d'.

5. EXPERIMENTATION

We use the CSE-CIC-IDS2018 dataset [2] for the experiments. Table 2 summarizes the statistics of attacks in Training and Test datasets. The experiment is performed on Google Colaboratory[3] under python 3 using TensorFlow and Graphics Processing Unit (GPU).

5.1. Performance metrics

We use the most important performance indicators, including, detection rate (DR), false alarm rate (FAR) and accuracy (ACC). Table 3 shows the four possible cases of correct and wrong classification.

$$DR_{Attack} = \frac{TP_{Attack}}{TP_{Attack} + FN_{Attack}} \qquad (12)$$

[2]https://registry.opendata.aws/cse-cic-ids2018/
[3]https://colab.research.google.com

Table 3: *Confusion matrix*

		Predicted class	
		Negative class	Positive class
Class	Negative class	True negative (TN)	False positive (FP)
	Positive class	False negative (FN)	True positive (TP)

Table 4: *Performance of deep discriminative models relative to the different attack type and benign*

	DNN	RNN	CNN
TNR (BENIGN)	96.915%	98.112%	98.914%
DR SSH-Bruteforce	100%	100%	100%
DR FTP-BruteForce	100%	100%	100%
DR Brute Force -XSS	83.265%	92.182%	92.101%
DR Brute Force -Web	82.223%	91.322%	91.002%
DR SQL Injection	100%	100%	100%
DR DoS attacks-Hulk	93.333%	94.912%	94.012%
DR DoS attacks-SlowHTTPTest	94.513%	96.123%	96.023%
DR DoS attacks-Slowloris	98.140%	98.220%	98.120%
DR DoS attacks-GoldenEye	92.110%	98.330%	98.221%
DR DDOS attack-HOIC	98.640%	98.711%	98.923%
DR DDOS attack-LOIC-UDP	97.348%	97.118%	97.888%
DR DDOS attack-LOIC-HTTP	97.222%	98.122%	98.991%
DR Botnet	96.420%	98.101%	98.982%
DR Infilteration	97.518%	97.874%	97.762%

$$TNR_{BENIGN} = \frac{TN_{BENIGN}}{TN_{BENIGN}+FP_{BENIGN}} \quad (13)$$

$$FAR = \frac{FP_{BENIGN}}{TN_{BENIGN}+FP_{BENIGN}} \quad (14)$$

$$Accuracy = \frac{TP_{Attack}+TN_{BENIGN}}{TP_{Attack}+FN_{Attack}+TN_{BENIGN}+FP_{BENIGN}} \quad (15)$$

where TP, TN, FP, and FN denote true positive, true negative, false positive, and false negative, respectively.

5.2. Results

Table 4 shows the performance of deep discriminative models relative to the different attack type and benign. It shows that deep neural network gives the highest true negative rate with 96.915%. The recurrent neural network gives the higest detection rate for seven attacks type, namely, Brute Force -XSS 92.182%, Brute Force -Web 91.322%, DoS attacks-Hulk 94.912%, DoS attacks-SlowHTTPTest 96.123%, DoS attacks-Slowloris 98.220%, DoS attacks-GoldenEye 98.330%, and Infilteration 97.874%. The convolutional neural network gives the higest detection rate for four attacks type, including, DDOS attack-HOIC 98.923%, DDOS attack-LOIC-UDP 97.888%, and DDOS attack-LOIC-HTTP 98.991%, and Botnet 98.982%.

Table 5: *Performance of generative/unsupervised models relative to the different attack type and benign*

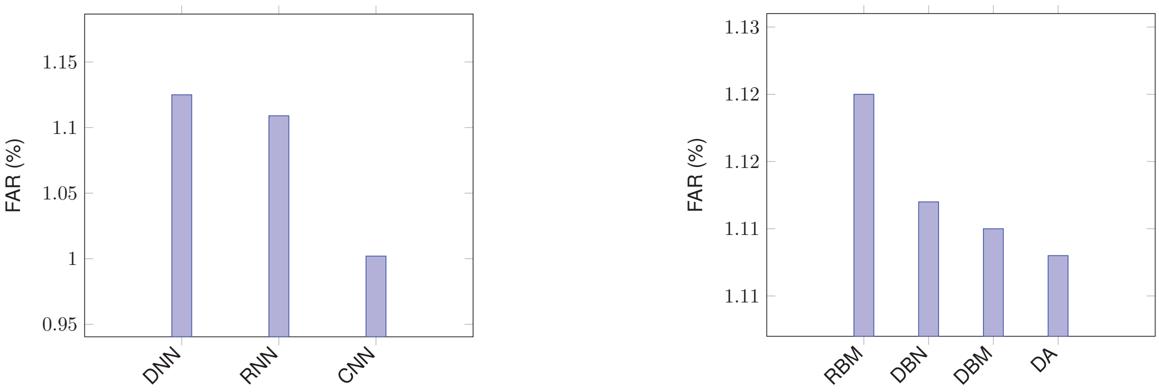
	RBM	DBN	DBM	DA
TNR (BENIGN)	97.316%	98.212%	96.215%	98.101%
DR SSH-Bruteforce	100%	100%	100%	100%
DR FTP-BruteForce	100%	100%	100%	100%
DR Brute Force -XSS	83.164%	92.281%	92.103%	95.223%
DR Brute Force -Web	82.221%	91.427%	91.254%	95.311%
DR SQL Injection	100%	100%	100%	100%
DR DoS attacks-Hulk	91.323%	91.712%	93.072%	92.112%
DR DoS attacks-SlowHTTPTest	93.313%	95.273%	95.993%	94.191%
DR DoS attacks-Slowloris	97.040%	97.010%	97.112%	97.120%
DR DoS attacks-GoldenEye	92.010%	97.130%	97.421%	96.222%
DR DDOS attack-HOIC	97.541%	97.211%	97.121%	96.551%
DR DDOS attack-LOIC-UDP	96.148%	96.122%	96.654%	96.445%
DR DDOS attack-LOIC-HTTP	96.178%	97.612%	97.121%	97.102%
DR Botnet	96.188%	97.221%	97.812%	97.717%
DR Infilteration	96.411%	96.712%	96.168%	97.818%

Table 6: *The accuracy and training time of deep discriminative models with different learning rate and hidden nodes*

Parameters	Accuracy and training time (s)	DNN	RNN	CNN
HN = 15	ACC	96.552%	96.872%	96.915%
LR=0.01	Time	20.2	30.3	28.4
HN = 15	ACC	96.651%	96.882%	96.912%
LR=0.1	Time	19.1	29.2	27.2
HN = 15	ACC	96.653%	96.886%	96.913%
LR=0.5	Time	18.9	29.1	27.1
HN = 30	ACC	96.612%	96.881%	96.922%
LR=0.01	Time	88.1	91.3	89.6
HN = 30	ACC	96.658%	96.888%	96.926%
LR=0.1	Time	87.9	90.9	88.5
HN = 30	ACC	96.662%	96.891%	96.929%
LR=0.5	Time	86.1	90.3	87.9
HN = 60	ACC	96.701%	96.903%	96.922%
LR=0.01	Time	180.2	197.5	192.2
HN = 60	ACC	96.921%	96.970%	96.975%
LR=0.1	Time	179.3	192.2	189.1
HN = 60	ACC	96.950%	96.961%	96.992%
LR=0.5	Time	177.7	190.6	182.6
HN = 100	ACC	97.102%	97.111%	97.222%
LR=0.01	Time	395.2	341.5	338.9
HN = 100	ACC	97.187%	97.229%	97.312%
LR=0.1	Time	391.1	336.9	332.5
HN = 100	ACC	97.281%	97.310%	97.376%
LR=0.5	Time	390.2	334.7	331.2

HN: Hidden Nodes; LR: Learning Rate

The performance of generative/unsupervised models relative to the different attack type and benign, is shown in Table 5. It shows that deep belief network gives the highest true negative rate with 98.212% and the higest detection rate for four attacks type, namely, Brute Force -XSS 92.281%, Brute Force -Web 91.427%, DoS attacks-Hulk



***(a)** Deep discriminative models* ***(b)** Generative/unsupervised models*

***Figure 8:** Performance of deep learning techniques in term of false alarm rate*

***Table 7:** The accuracy and training time of generative/unsupervised models with different learning rate and hidden nodes*

Parameters	Accuracy and training time (s)	RBM	DBN	DBM	DA
HN = 15	ACC	96.551%	96.852%	96.911%	96.912%
LR=0.01	Time	20.0	30.1	28.3	28.3
HN = 15	ACC	96.642%	96.871%	96.901%	96.902%
LR=0.1	Time	19.0	29.1	27.1	27.2
HN = 15	ACC	96.651%	96.885%	96.910%	96.911%
LR=0.5	Time	18.8	28.1	26.2	27.1
HN = 30	ACC	96.602%	96.844%	96.918%	96.917%
LR=0.01	Time	88.0	90.4	89.5	88.6
HN = 30	ACC	96.656%	96.884%	96.922%	96.923%
LR=0.1	Time	87.4	90.7	88.3	88.2
HN = 30	ACC	96.661%	96.890%	96.925%	96.924%
LR=0.5	Time	86.1	90.3	87.9	87.10
HN = 60	ACC	96.691%	96.883%	96.912%	96.913%
LR=0.01	Time	180.1	196.5	191.1	191.4
HN = 60	ACC	96.920%	96.967%	96.972%	96.971%
LR=0.1	Time	179.1	192.1	189.0	189.1
HN = 60	ACC	96.947%	96.960%	96.991%	96.992%
LR=0.5	Time	177.6	190.5	181.4	181.4
HN = 100	ACC	97.101%	97.108%	97.211%	97.221%
LR=0.01	Time	394.1	340.4	339.1	337.11
HN = 100	ACC	97.186%	97.227%	97.300%	97.311%
LR=0.1	Time	390.0	334.8	330.1	331.7
HN = 100	ACC	97.280%	97.302%	97.371%	97.372%
LR=0.5	Time	390.1	344.7	351.5	341.3

HN: Hidden Nodes; LR: Learning Rate

91.712%, and DDOS attack-LOIC-HTTP 97.612%. The deep auto encoders gives the higest detection rate for three attacks type, namely, Brute Force - Web 95.311%, DoS attacks-Slowloris 97.120%, and Infilteration 97.818%. The deep Boltzmann machine gives the higest detection rate for five attacks type, namely, DoS attacks-Hulk 93.072%, DoS attacks-SlowHTTPTest 95.993%, DoS attacks-GoldenEye 97.421%, DDOS attack-LOIC-UDP 96.654%, and Botnet 97.812%.

Table 6 presents the accuracy and training time of deep discriminative models with different learning rate and hidden nodes. Compared to both deep neural network and recurrent neural network, the convolutional neural network gets a higher accuracy 97.376%, when there are 100 hidden nodes and the learning rate is 0.5.

Table 7 demonstrates the accuracy and training time of generative/unsupervised models with different learning rate and hidden nodes. The deep auto encoders gets a higher accuracy 97.372%, when there are 100 hidden nodes and the learning rate is 0.5 compared to three techniques, including, restricted Boltzmann machine, deep belief network, and deep boltzmann machine.

The performance of deep learning techniques in term of false alarm rate is depicted in Figure 8. In the generative/unsupervised models, mean false alarm rate of the convolutional neural network is better than both deep neural network and recurrent neural network. In the deep discriminative models, mean false alarm rate of the deep autoencoders is better than three techniques, including, restricted Boltzmann machine, deep belief network, and deep Boltzmann machine.

6. CONCLUSION

In this paper, we conducted a comparative study of deep learning techniques for intrusion detection, namely, deep discriminative models and generative/unsupervised models. Specifically, we analyzed seven deep learning approaches, including, deep neural networks, recurrent neural networks, convolutional neural networks, restricted Boltzmann machine, deep belief networks, deep Boltzmann machines, and deep autoencoders. These machine learning methods are compared using the CSE-CIC-IDS 2018 dataset with three important performance indicators, namely, accuracy, detection rate, and false alarm rate.

REFERENCES

Abusitta, A., M. Bellaiche, M. Dagenais, and T. Halabi (2019). A deep learning approach for proactive multi-cloud cooperative intrusion detection system. *Future Generation Computer Systems*.

Ahmim, A., M. Derdour, and M. A. Ferrag (2018). An intrusion detection system based on combining probability predictions of a tree of classifiers. *International Journal of Communication Systems 31*(9), e3547.

Ahmim, A., L. Maglaras, M. A. Ferrag, M. Derdour, and H. Janicke (2018). A novel hierarchical intrusion detection system based on decision tree and rules-based models. *arXiv preprint arXiv:1812.09059*.

Aldwairi, T., D. Perera, and M. A. Novotny (2018). An evaluation of the performance of restricted boltzmann machines as a model for anomaly network intrusion detection. *Computer Networks 144*, 111–119.

Alom, M. Z., V. Bontupalli, and T. M. Taha (2015). Intrusion detection using deep belief networks. In *2015 National Aerospace and Electronics Conference (NAECON)*, pp. 339–344. IEEE.

Alrawashdeh, K. and C. Purdy (2016). Toward an online anomaly intrusion detection system based on deep learning. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 195–200. IEEE.

Basumallik, S., R. Ma, and S. Eftekharnejad (2019). Packet-data anomaly detection in pmu-based state estimator using convolutional neural network. *International Journal of Electrical Power & Energy Systems 107*, 690–702.

Deng, L. and D. Yu (2014). Deep learning: methods and applications. *Foundations and Trends® in Signal Processing 7*(3–4), 197–387.

Dewa, Z. and L. A. Maglaras (2016). Data mining and intrusion detection systems. *International Journal of Advanced Computer Science and Applications 7*(1), 62–71.

Elman, J. L. (1990). Finding structure in time. *Cognitive science 14*(2), 179–211.

Feng, F., X. Liu, B. Yong, R. Zhou, and Q. Zhou (2019). Anomaly detection in ad-hoc networks based on deep learning model: A plug and play device. *Ad Hoc Networks 84*, 82–89.

Fiore, U., F. Palmieri, A. Castiglione, and A. De Santis (2013). Network anomaly detection with the restricted boltzmann machine. *Neurocomputing 122*, 13–23.

Fischer, A. and C. Igel (2012). An introduction to restricted boltzmann machines. In *iberoamerican congress on pattern recognition*, pp. 14–36. Springer.

Fu, K., D. Cheng, Y. Tu, and L. Zhang (2016). Credit card fraud detection using convolutional neural networks. In *International Conference on Neural Information Processing*, pp. 483–490. Springer.

Gao, N., L. Gao, Q. Gao, and H. Wang (2014). An intrusion detection model based on deep belief networks. In *2014 Second International Conference on Advanced Cloud and Big Data*, pp. 247–252. IEEE.

Gelly, G. and J.-L. Gauvain (2017). Optimization of rnn-based speech activity detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 26*(3), 646–656.

Gu, J., Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al. (2018). Recent advances in convolutional neural networks. *Pattern Recognition 77*, 354–377.

He, K., X. Zhang, S. Ren, and J. Sun (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Hinton, G. E. (2009). Deep belief networks. *Scholarpedia 4*(5), 5947.

Jaeger, H. and H. Haas (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *science 304*(5667), 78–80.

Jiang, F., Y. Fu, B. B. Gupta, F. Lou, S. Rho, F. Meng, and Z. Tian (2018). Deep learning based multi-channel intelligent attack detection for data security. *IEEE Transactions on Sustainable Computing*.

Jordan, M. I. (1997). Serial order: A parallel distributed processing approach. In *Advances in psychology*, Volume 121, pp. 471–495. Elsevier.

Kang, M.-J. and J.-W. Kang (2016). Intrusion detection system using deep neural network for in-vehicle network security. *PloS one 11*(6), e0155781.

Kasongo, S. M. and Y. Sun (2019). A deep learning method with filter based feature engineering for wireless intrusion detection system. *IEEE Access 7*, 38597–38607.

Khan, F. A., A. Gumaei, A. Derhab, and A. Hussain (2019). Tsdl: A twostage deep learning model for efficient network intrusion detection. *IEEE Access*.

Kim, J., J. Kim, H. L. T. Thu, and H. Kim (2016). Long short term memory recurrent neural network classifier for intrusion detection. In *2016 International Conference on Platform Technology and Service (PlatCon)*, pp. 1–5. IEEE.

Liu, W., Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi (2017). A survey of deep neural network architectures and their applications. *Neurocomputing 234*, 11–26.

Maglaras, L. A., K.-H. Kim, H. Janicke, M. A. Ferrag, S. Rallis, P. Fragkou, A. Maglaras, and T. J. Cruz (2018). Cyber security of critical infrastructures. *ICT Express 4*(1), 42–45.

Otoum, S., B. Kantarci, and H. T. Mouftah (2019). On the feasibility of deep learning in sensor network intrusion detection. *IEEE Networking Letters*.

Papamartzivanos, D., F. G. Mármol, and G. Kambourakis (2019). Introducing deep learning self-adaptive misuse network intrusion detection systems. *IEEE Access 7*, 13546–13560.

Potluri, S. and C. Diedrich (2016). Accelerated deep neural networks for enhanced intrusion detection system. In *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–8. IEEE.

Roy, S. S., A. Mallik, R. Gulati, M. S. Obaidat, and P. Krishna (2017). A deep learning based artificial neural network approach for intrusion detection. In *International Conference on Mathematics and Computing*, pp. 44–53. Springer.

Salakhutdinov, R. and H. Larochelle (2010). Efficient learning of deep boltzmann machines. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 693–700.

Salama, M. A., H. F. Eid, R. A. Ramadan, A. Darwish, and A. E. Hassanien (2011). Hybrid intelligent intrusion detection scheme. In *Soft computing in industrial applications*, pp. 293–303. Springer.

Sharafaldin, I., A. H. Lashkari, and A. A. Ghorbani (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *ICISSP*, pp. 108–116.

Shone, N., T. N. Ngoc, V. D. Phai, and Q. Shi (2018). A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence 2*(1), 41–50.

Stewart, B., L. Rosa, L. A. Maglaras, T. J. Cruz, M. A. Ferrag, P. Simões, and H. Janicke (2017). A novel intrusion detection mechanism for scada systems which automatically adapts to network topology changes. *EAI Endorsed Trans. Indust. Netw. & Intellig. Syst. 4*(10), e4.

Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9.

Tang, T. A., L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho (2016). Deep learning approach for network intrusion detection in software defined networking. In *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pp. 258–263. IEEE.

Tang, T. A., L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho (2018). Deep recurrent neural network for intrusion detection in sdn-based networks. In *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, pp. 202–206. IEEE.

Vincent, P., H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research 11*(Dec), 3371–3408.

Yang, J., J. Deng, S. Li, and Y. Hao (2017). Improved traffic detection with support vector machine based on restricted boltzmann machine. *Soft Computing 21*(11), 3101–3112.

Yin, C., Y. Zhu, J. Fei, and X. He (2017). A deep learning approach for intrusion detection using recurrent neural networks. *Ieee Access 5*, 21954–21961.

Yu, Y., J. Long, and Z. Cai (2017). Network intrusion detection through stacking dilated convolutional autoencoders. *Security and Communication Networks 2017*.

Zeiler, M. D. and R. Fergus (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer.

Zhang, H., X. Yu, P. Ren, C. Luo, and G. Min (2019). Deep adversarial learning in intrusion detection:

A data augmentation enhanced framework. *arXiv preprint arXiv:1901.07949*.

Zhang, Z., X. Zhou, X. Zhang, L. Wang, and P. Wang (2018). A model based on convolutional neural network for online transaction fraud detection. *Security and Communication Networks 2018*.

Zhao, G., C. Zhang, and L. Zheng (2017). Intrusion detection using deep belief network and probabilistic neural network. In *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, Volume 1, pp. 639–642. IEEE.

Zhou, L., X. Ouyang, H. Ying, L. Han, Y. Cheng, and T. Zhang (2018). Cyber-attack classification in smart grid via deep neural network. In *Proceedings of the 2nd International Conference on Computer Science and Application Engineering*, pp. 90. ACM.