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 Acquisition (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 (2018), 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 efficiency of such mechanisms, reliable datasets that contain both benign and several attacks, meets real world criteria and that is publicly available 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,
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 a 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
Deep Learning Techniques for Cyber Security Intrusion Detection: A Detailed Analysis
Ferrag • Maglaras • Janicke

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

<table>
<thead>
<tr>
<th>Deep Learning Technique</th>
<th>IDS</th>
<th>Dataset Used</th>
<th>No. of times cited (as of 30/05/2019)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deep neural network</td>
<td>Tang et al. (2016)</td>
<td>NSL-KDD dataset</td>
<td>110</td>
</tr>
<tr>
<td>Deep neural network</td>
<td>Potluri and Diedrich (2016)</td>
<td>NSL-KDD dataset</td>
<td>37</td>
</tr>
<tr>
<td>Deep neural network</td>
<td>Zhou et al. (2018)</td>
<td>4 types of attacks (DDoS, R2L, U2R, and PROBING)</td>
<td>0</td>
</tr>
<tr>
<td>Deep neural network</td>
<td>Feng et al. (2019)</td>
<td>KDD Cup 1999 dataset</td>
<td>1</td>
</tr>
<tr>
<td>Deep neural network</td>
<td>Zhang et al. (2019)</td>
<td>KDD Cup 1999 dataset</td>
<td>0</td>
</tr>
<tr>
<td>Deep neural network</td>
<td>Roy et al. (2017)</td>
<td>KDD Cup 1999 dataset</td>
<td>23</td>
</tr>
<tr>
<td>Feed forward deep neural network</td>
<td>Kasingir and Sun (2019)</td>
<td>NSL-KDD dataset</td>
<td>0</td>
</tr>
<tr>
<td>Recurrent neural network</td>
<td>Yn et al. (2017)</td>
<td>KDD Cup 1999 dataset</td>
<td>86</td>
</tr>
<tr>
<td>Recurrent neural network</td>
<td>Tang et al. (2018)</td>
<td>NSL-KDD dataset</td>
<td>9</td>
</tr>
<tr>
<td>Recurrent neural network</td>
<td>Jiang et al. (2018)</td>
<td>NSL-KDD dataset</td>
<td>22</td>
</tr>
<tr>
<td>Convolutional neural network</td>
<td>Basumalik et al. (2019)</td>
<td>IEEE-30 bus and IEEE-118 bus</td>
<td>1</td>
</tr>
<tr>
<td>Convolutional neural network</td>
<td>Fu et al. (2016)</td>
<td>Credit card transaction data</td>
<td>47</td>
</tr>
<tr>
<td>Convolutional neural network</td>
<td>Feng et al. (2019)</td>
<td>KDD Cup 1999 dataset</td>
<td>1</td>
</tr>
<tr>
<td>Convolutional autoencoder</td>
<td>Yu et al. (2017)</td>
<td>Contagio-CTU UNB dataset</td>
<td>17</td>
</tr>
<tr>
<td>Restricted Boltzmann machine</td>
<td>Alrawashdeh and Purdy (2016)</td>
<td>KDD Cup 1999 dataset</td>
<td>35</td>
</tr>
<tr>
<td>Restricted Boltzmann machine</td>
<td>Ackawi et al. (2018)</td>
<td>SICK dataset</td>
<td>4</td>
</tr>
<tr>
<td>Restricted Boltzmann machine</td>
<td>Faere et al. (2013)</td>
<td>KDD Cup 1999 dataset</td>
<td>176</td>
</tr>
<tr>
<td>Restricted Boltzmann machine</td>
<td>Salama et al. (2011)</td>
<td>NSL-KDD dataset</td>
<td>96</td>
</tr>
<tr>
<td>Restricted Boltzmann machine</td>
<td>Gao et al. (2014)</td>
<td>KDD Cup 1999 dataset</td>
<td>69</td>
</tr>
<tr>
<td>Restricted Boltzmann machine</td>
<td>Alom et al. (2015)</td>
<td>NSL-KDD dataset</td>
<td>59</td>
</tr>
<tr>
<td>Restricted Boltzmann machine</td>
<td>Fung et al. (2017)</td>
<td>Real online network traffic</td>
<td>16</td>
</tr>
<tr>
<td>Restricted Boltzmann machine</td>
<td>Osuam et al. (2019)</td>
<td>KDD Cup 1999 dataset</td>
<td>3</td>
</tr>
<tr>
<td>Deep belief network</td>
<td>Zhao et al. (2017)</td>
<td>KDD Cup 1999 dataset</td>
<td>13</td>
</tr>
<tr>
<td>Deep auto-encoder</td>
<td>Shone et al. (2018)</td>
<td>NSL-KDD dataset</td>
<td>66</td>
</tr>
<tr>
<td>Deep auto-encoder</td>
<td>Khan et al. (2019)</td>
<td>UNSW-NB15 dataset</td>
<td>0</td>
</tr>
<tr>
<td>Deep auto-encoder</td>
<td>Papamarkozias et al. (2019)</td>
<td>NSL-KDD dataset</td>
<td>1</td>
</tr>
<tr>
<td>Denoising auto-encoder</td>
<td>Abouelef et al. (2019)</td>
<td>KDD Cup 1999 dataset</td>
<td>1</td>
</tr>
</tbody>
</table>

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, CAIDA, LBNL, CDX, KYOTO, TWENDE, UMASS, and ADF2013), which are not yet used by deep learning techniques. In our work, we use a new real traffic data set “CSE-CIC-IDS2018” developed by the Communications Security Establishment (CSE) & the Canadian Institute for Cybersecurity (CIC).

1 https://registry.opendata.aws/cse-cic-ids2018/
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), and deep belief networks (DBNs). The generative/unsupervised models include restricted Boltzmann machine (RBM), deep belief networks (DBN), and Deep Boltzmann machines (DBM), 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 supervised learning; 2) Deep networks for unsupervised 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 $D_M(a_M, \alpha_M, a_M)$. $a_M \in N$ is the number of neurons in the layer. $\alpha_M : \mathbb{R}^{n_M} \rightarrow \mathbb{R}^{n_M}$ is the affine transformation defined by the matrix $W_M$ and the vector $b_M$. $\alpha_M : \mathbb{R}^{n_M} \rightarrow \mathbb{R}^{n_M}$ 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.

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 $VW_M \in \mathbb{R}^{n_M \times n_M}$ 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.

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

Algorithm 1 DNN network based on MLP

1. Choose a learning pair $(x, y)$;
2. $b_M = x$;
3. for $M = 1$ to $N$ do
4. \[ g_M = a_M (h_M - 1) = W_M \times h_M - 1 + b_M; \]
5. \[ h_M = \alpha_M(g_M) \]
6. end for

Algorithm 2 Recurrent neural network

1. Choose a learning pair $(x(t), y(t))$;
2. $b_M(t) = x(t)$, \forall t \in [1, t_f]$;
3. for $M = 1$ to $N$ do
4. \[ g_M(t) = W_M \times h_M(t-1) + VW_M \times b_M(t-1) + b_M; \]
5. \[ h_M(t) = \alpha_M(g_M(t)); \]
6. end for
7. end for
Deep Learning Techniques for Cyber Security Intrusion Detection: A Detailed Analysis
Ferrag • Maglaras • Janicke

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:

$$E_n(V, H, G) = \sum_i \sum_j (V_j W_{ij} H_j) + \sum_i b_i V_i - \sum_j c_j H_j$$

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

$$\text{Prob}(V, H, G) = \frac{1}{Z(G)} e^{-E_n(V, H, G)}$$

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

$$Z(G) = \sum_{V \in V} \sum_{H \in H} e^{-E_n(V, H, G)}$$

where curved letters $\forall$ and $\forall$ 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
probability of generating a visible vector, \( V \), can be calculated as:

\[
Prob(V) = \sum_H Prob(H \mid W) \cdot Prob(V \mid H, W)
\]  

(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:

\[
En(V, H, G) = -V^T W^1 H^1 - V^1 W^2 H^2 - V^2 W^3 H^3
\]  

(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:

\[
Prob(V, G) = \frac{1}{Z(G)} \sum_H e^{-En(V, H, G)}
\]  

(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 follows:

\[
encoder_G(x) = s(Wx + b)
\]  

(10)

\[
decoder_G(y) = s(W'y + b')
\]  

(11)

5. EXPERIMENTATION

We use the CSE-CIC-IDS2018 dataset for the experiments. Table 2 summarizes the statistics of attacks in Training and Test datasets. The experiment is performed on Google Colaboratory 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.

\[
DR_{Attack} = \frac{TP_{Attack}}{TP_{Attack} + FN_{Attack}}
\]  

(12)

\[
FAR_{Attack} = \frac{FP_{Attack}}{TP_{Attack} + FP_{Attack}}
\]  

(13)
Table 3: Confusion matrix

<table>
<thead>
<tr>
<th>Class</th>
<th>Positive cases</th>
<th>Negative cases</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Negative class</td>
<td>True negative (TN)</td>
</tr>
<tr>
<td></td>
<td>Positive class</td>
<td>True positive (TP)</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Attack Type</th>
<th>DBM</th>
<th>DNN</th>
<th>CNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>DR DoS att-attack-Hulk</td>
<td>91.32%</td>
<td>91.72%</td>
<td>93.07%</td>
</tr>
<tr>
<td>DR DoS att-attack-SlowHTTPTest</td>
<td>93.73%</td>
<td>95.27%</td>
<td>96.40%</td>
</tr>
<tr>
<td>DR DoS att-attack-Slowloris</td>
<td>92.10%</td>
<td>97.13%</td>
<td>97.42%</td>
</tr>
<tr>
<td>DR DoS att-attack-GoldenEye</td>
<td>96.87%</td>
<td>97.18%</td>
<td>97.31%</td>
</tr>
<tr>
<td>DR DoS att-attack-LOIC-UDP</td>
<td>97.34%</td>
<td>97.18%</td>
<td>97.88%</td>
</tr>
<tr>
<td>DR DoS att-attack-LOIC-HTTP</td>
<td>97.22%</td>
<td>97.12%</td>
<td>97.70%</td>
</tr>
<tr>
<td>DR Botnet</td>
<td>96.42%</td>
<td>98.10%</td>
<td>98.98%</td>
</tr>
<tr>
<td>DR Infiltration</td>
<td>97.51%</td>
<td>97.84%</td>
<td>97.76%</td>
</tr>
</tbody>
</table>

\[
TNR_{\text{BENIGN}} = \frac{TN_{\text{BENIGN}}}{TN_{\text{BENIGN}} + FP_{\text{NONBENIGN}}} \quad (13)
\]

\[
FAR = \frac{FP_{\text{NONBENIGN}}}{TP_{\text{BENIGN}} + FP_{\text{NONBENIGN}}} \quad (14)
\]

\[
\text{Accuracy} = \frac{TP_{\text{BENIGN}} + TN_{\text{BENIGN}}}{TP_{\text{BENIGN}} + TN_{\text{BENIGN}} + FP_{\text{NONBENIGN}} + FN_{\text{NONBENIGN}}} \quad (15)
\]

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 highest 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 Infiltration 97.874%. The convolutional neural network gives the highest detection rate for four attacks type, including, DDOS attack-LOIC 98.923%, DDOS attack-LOIC-UDP 97.889%, 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

<table>
<thead>
<tr>
<th>Attack Type</th>
<th>RBM</th>
<th>DBN</th>
<th>DBM</th>
<th>DA</th>
</tr>
</thead>
<tbody>
<tr>
<td>DR SSH-BruteForce</td>
<td>97.316%</td>
<td>98.212%</td>
<td>96.215%</td>
<td>98.101%</td>
</tr>
<tr>
<td>DR FTP-BruteForce</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>DR Brute Force -XSS</td>
<td>83.16%</td>
<td>92.28%</td>
<td>92.10%</td>
<td>95.23%</td>
</tr>
<tr>
<td>DR Brute Force -Web</td>
<td>82.22%</td>
<td>91.42%</td>
<td>91.25%</td>
<td>95.31%</td>
</tr>
<tr>
<td>DR SQL Injection</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>DR DoS attacks-Hulk</td>
<td>91.32%</td>
<td>91.72%</td>
<td>93.07%</td>
<td>92.11%</td>
</tr>
<tr>
<td>DR DoS attacks-SlowHTTPTest</td>
<td>93.73%</td>
<td>95.27%</td>
<td>96.40%</td>
<td>97.10%</td>
</tr>
<tr>
<td>DR DoS attacks-Slowloris</td>
<td>92.10%</td>
<td>97.13%</td>
<td>97.42%</td>
<td>97.70%</td>
</tr>
<tr>
<td>DR DoS attacks-GoldenEye</td>
<td>96.87%</td>
<td>97.18%</td>
<td>97.31%</td>
<td>97.71%</td>
</tr>
<tr>
<td>DR DoS attacks-LOIC-UDP</td>
<td>97.34%</td>
<td>97.18%</td>
<td>97.88%</td>
<td>98.18%</td>
</tr>
<tr>
<td>DR DoS attacks-LOIC-HTTP</td>
<td>97.22%</td>
<td>97.12%</td>
<td>97.70%</td>
<td>97.98%</td>
</tr>
<tr>
<td>DR Botnet</td>
<td>96.42%</td>
<td>98.10%</td>
<td>98.98%</td>
<td>99.78%</td>
</tr>
<tr>
<td>DR Infiltration</td>
<td>97.51%</td>
<td>97.84%</td>
<td>97.76%</td>
<td>97.65%</td>
</tr>
</tbody>
</table>

5.2. Results

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

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Accuracy and training time (s)</th>
<th>DNN</th>
<th>RNN</th>
<th>CNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>HN = 15</td>
<td>ACC 96.55%</td>
<td>96.87%</td>
<td>96.91%</td>
<td>96.91%</td>
</tr>
<tr>
<td>LR=0.1</td>
<td>Time 26.1</td>
<td>26.3</td>
<td>28.4</td>
<td></td>
</tr>
<tr>
<td>HN = 15</td>
<td>ACC 96.55%</td>
<td>96.87%</td>
<td>96.91%</td>
<td>96.91%</td>
</tr>
<tr>
<td>LR=0.5</td>
<td>Time 19.1</td>
<td>29.2</td>
<td>27.3</td>
<td></td>
</tr>
<tr>
<td>HN = 30</td>
<td>ACC 96.66%</td>
<td>96.89%</td>
<td>96.93%</td>
<td>96.93%</td>
</tr>
<tr>
<td>LR=0.1</td>
<td>Time 88.1</td>
<td>91.3</td>
<td>88.6</td>
<td></td>
</tr>
<tr>
<td>HN = 60</td>
<td>ACC 96.70%</td>
<td>96.93%</td>
<td>96.92%</td>
<td>96.92%</td>
</tr>
<tr>
<td>LR=0.1</td>
<td>Time 180.2</td>
<td>197.5</td>
<td>192.2</td>
<td></td>
</tr>
<tr>
<td>HN = 60</td>
<td>ACC 96.90%</td>
<td>96.97%</td>
<td>96.97%</td>
<td>96.97%</td>
</tr>
<tr>
<td>LR=0.5</td>
<td>Time 179.3</td>
<td>192.2</td>
<td>189.1</td>
<td></td>
</tr>
<tr>
<td>HN = 60</td>
<td>ACC 96.95%</td>
<td>96.96%</td>
<td>96.96%</td>
<td>96.96%</td>
</tr>
<tr>
<td>LR=0.5</td>
<td>Time 177.7</td>
<td>190.6</td>
<td>182.6</td>
<td></td>
</tr>
<tr>
<td>HN = 100</td>
<td>ACC 97.10%</td>
<td>97.11%</td>
<td>97.22%</td>
<td>97.22%</td>
</tr>
<tr>
<td>LR=0.1</td>
<td>Time 395.2</td>
<td>341.5</td>
<td>336.9</td>
<td></td>
</tr>
<tr>
<td>HN = 100</td>
<td>ACC 97.18%</td>
<td>97.22%</td>
<td>97.31%</td>
<td>97.31%</td>
</tr>
<tr>
<td>LR=0.5</td>
<td>Time 390.2</td>
<td>334.7</td>
<td>331.2</td>
<td></td>
</tr>
</tbody>
</table>

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 highest detection rate for four attacks type, namely, Brute Force -XSS 92.281%, Brute Force -Web 91.427%, DoS attacks-Hulk 98.922%.
Deep Learning Techniques for Cyber Security Intrusion Detection: A Detailed Analysis

Ferrag • Maglaras • Janicke

Deep discriminative models

- RBM
- DBN
- DBM
- DA

Generative/unsupervised models

- DNN
- RNN
- CNN

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

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Accuracy and training time (s)</th>
<th>RBM</th>
<th>DBN</th>
<th>DBM</th>
<th>DA</th>
</tr>
</thead>
<tbody>
<tr>
<td>HN = 15, LR=0.01</td>
<td>ACC 96.55%</td>
<td>96.65%</td>
<td>96.55%</td>
<td>96.55%</td>
<td></td>
</tr>
<tr>
<td>HN = 15, LR=0.1</td>
<td>10.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td></td>
</tr>
<tr>
<td>HN = 19, LR=0.5</td>
<td>18.8</td>
<td>28.1</td>
<td>28.1</td>
<td>28.1</td>
<td></td>
</tr>
<tr>
<td>HN = 19, LR=0.1</td>
<td>19.0</td>
<td>29.1</td>
<td>29.1</td>
<td>29.1</td>
<td></td>
</tr>
<tr>
<td>HN = 19, LR=0.01</td>
<td>21.0</td>
<td>31.1</td>
<td>31.1</td>
<td>31.1</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Accuracy and training time (s)</th>
<th>RBM</th>
<th>DBN</th>
<th>DBM</th>
<th>DA</th>
</tr>
</thead>
<tbody>
<tr>
<td>HN = 100, LR=0.01</td>
<td>ACC 97.10%</td>
<td>97.10%</td>
<td>97.10%</td>
<td>97.10%</td>
<td></td>
</tr>
<tr>
<td>HN = 100, LR=0.1</td>
<td>340.1</td>
<td>340.1</td>
<td>340.1</td>
<td>340.1</td>
<td></td>
</tr>
<tr>
<td>HN = 100, LR=0.5</td>
<td>340.1</td>
<td>340.1</td>
<td>340.1</td>
<td>340.1</td>
<td></td>
</tr>
</tbody>
</table>

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 get 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...
Deep Learning Techniques for Cyber Security Intrusion Detection: A Detailed Analysis
Ferrag ● Maglaras ● Janicke

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


