Chapter 1
Hybrid Intelligent Intrusion Detection Scheme

Mostafa A. Salama, Heba F. Eid, Rabie A. Ramadan, Ashraf Darwish, and Aboul Ella Hassanien

Abstract This paper introduces a hybrid scheme that combines the advantages of deep belief network and support vector machine. An application of intrusion detection imaging has been chosen and hybridization scheme have been applied to see their ability and accuracy to classify the intrusion into two outcomes: normal or attack, and the attacks fall into four classes; R2L, DoS, U2R, and Probing. First, we utilize deep belief network to reduct the dimensionality of the feature sets. This is followed by a support vector machine to classify the intrusion into five outcome; Normal, R2L, DoS, U2R, and Probing. To evaluate the performance of our approach, we present tests on NSL-KDD dataset and show that the overall accuracy offered by the employed approach is high.

Key words: Deep Belief Network (DBN); Network Intrusion detection system; Support Vector Machines (SVMs); Dimensional reduction.

Mostafa A. Salama
Department of Computer Science, British University in Egypt, Cairo, Egypt e-mail: mostafa.salama@gmail.com

Heba F. Eid
Faculty of Science, Al-Azhar University,Cairo, Egypt e-mail: heba.fathy@yahoo.com

Rabie A. Ramadan
Cairo University, Faculty of Engineering, Computer Engineering Department, Cairo, Egypt e-mail: rabieramadan@gmail.com

Ashraf Darwish
Faculty of Science, Helwan University, Cairo, Egypt e-mail: amodarwish@yahoo.com

Aboul Ella Hassanien
Faculty of Computers and Information, Cairo University e-mail: aboitcairo@gmail.com

1.1 Introduction

The Internet and online procedures is an essential tool of our daily life today. They have been used as an important component of business operation [1]. Therefore, network security needs to be carefully concerned to provide secure information channels. Intrusion detection (ID) is a major research problem in network security, where the concept of ID was proposed by Anderson in 1980 [2]. ID is based on the assumption that the behavior of intruders is different from a legal user [3]. The goal of intrusion detection systems (IDS) is to identify unusual access or attacks to secure internal networks [4] Network-based IDS is a valuable tool for the defense-in-depth of computer networks. It looks for known or potential malicious activities in network traffic and raises an alarm whenever a suspicious activity is detected. In general, IDSs can be divided into two techniques: misuse detection and anomaly detection [5, 6]. Misuse intrusion detection (signature-based detection) uses well-defined patterns of the malicious activity to identify intrusions [7,8]. However, it may not be able to alert the system administrator in case of a new attack. Anomaly detection attempts to model normal behavior profile. It identifies malicious traffic based on the deviations from the normal patterns, where the normal patterns are constructed from the statistical measures of the system features [9]. The anomaly detection techniques have the advantage of detecting unknown attacks over the misuse detection technique [10]. Several machine-learning techniques including neural networks, fuzzy logic [11], support vector machines (SVM) [9, 11] have been studied for the design of IDS. In particular, these techniques are developed as classifiers, which are used to classify whether the incoming network traffics are normal or an attack. In this paper, we propose an anomaly intrusion detection scheme using Deep Belief Network (DBN) based on Restricted Boltzmann Machine (RBM) [12, 13]. DBN is used as feature reduction method [14] that is followed by SVM classifier. We evaluate the effectiveness of the proposed DBN-SVM scheme by conducting several experiments on NSL-KDD dataset. We examine the performance of the DBN-SVM scheme in comparison with standalone DBN and standalone SVM classifier. Also, DBN as a feature reduction method is compared with other known feature reduction methods. The rest of this paper is organized as follows: Section II gives an overview of RBM architecture and DBN. Section III describes DBN classifier and the proposed DBN-SVM intrusion detection scheme. The experimental results and conclusions are presented in Section IV and V respectively.

1.2 AN OVERVIEW

This section discusses the deep belief network structure including the exploration of the restricted Boltzmann machine.

1.2.1 Restricted Boltzmann Machine

RBM is an energy-based undirected generative model that uses a layer of hidden variables to model a distribution over visible variables [14, 15]. The undirected model for the interactions between the hidden and visible variables is used to ensure that the contribution of the likelihood term to the posterior over the hidden variables is approximately factorial which greatly facilitates inference [16]. Energy-based model means that the probability distribution over the variables of interest is defined through an energy function. It is composed from a set of observable variables $V = \{v_i\}$ and a set of hidden variables $H = \{h_j\}$, i node in the visible layer, j node in the hidden layer. It is restricted in the sense that there are no visible-visible or hidden-hidden connections. The steps of the RBM learning algorithm can be declared as follows:

1. Due to the conditional independence (no connection) between nodes in the same layer (Property in RBM), the conditional distributions are given in Equations (1) and (2).

$$\begin{cases} P(H|V) = \prod_j p(h_j|v) \\ p(h_j = 1|v) = f(a_i + \sum_i w_{ij} v_i) \\ p(h_j = 0|v) = 1 - p(h_j = 1|v); \end{cases} \tag{1.1}$$

   And

$$\begin{cases} P(H|V) = \prod_i p(v_i|h) \\ p(v_i = 1|h) = f(b_j + \sum_j w_{ij} h_j) \\ p(v_i = 0|h) = 1 - p(v_i = 1|h); \end{cases} \tag{1.2}$$

   Where f is a sigmoid function (σ) which takes the form $\sigma(z) = 1/1+e^{-z}$ for binary data vector.
2. The likelihood distribution between hidden and visible units is defined as:

$$P(v, h) = \frac{e^{-E(v,h)}}{\Sigma_i e^{-E(v_i,h)}} \tag{1.3}$$

   Where $E(x, h){=}{-}\bar{h}wv - \bar{b}v - \bar{c}h$,
   And $\bar{h}$,$\bar{b}$, $\bar{c}$ are the transposes of matrices h, b and c.

3. The average of the log likelihood with respect to the parameters is given by

$$\begin{aligned} \Delta w_{ij} &= \varepsilon^{*}(\delta \log p(v)/\delta w_{ij}) \\ &= \varepsilon(< x_i h_j >_{data} - < v_i h_j >_{model}) \end{aligned} \tag{1.4}$$

$$\Delta v_i = \varepsilon(< v_i^2 >_{data} - < v_i^2 >_{model}) \tag{1.5}$$

$$\Delta h_i = \varepsilon(< h_i^2 >_{data} - < h_i^2 >_{model}) \tag{1.6}$$

4. The term $<>_{model}$ takes exponential time to compute exactly so the Contrastive Divergence (CD) approximation to the gradient is used instead [6]. Contrastive divergence is a method that depends on the approximation that is to run the sampler for a single Gibbs iteration, instead until the chain converges. In this case the term $<>_1$ will be used such that it represents the expectation with respect to the distribution of samples from running the Gibbs sampler initialized at the data for one full step, the new update rule will be.

$$\Delta w_{ij} = \varepsilon(< v_i h_j >_{data} - < v_i h_j > 1) \tag{1.7}$$

$$\Delta v_i = \varepsilon(< v_i^2 >_{data} - < v_i^2 > 1) \tag{1.8}$$

$$\Delta h_i = \varepsilon(< h_i^2 >_{data} - < h_i^2 > 1) \tag{1.9}$$

The Harmonium RBM is an RBM with Gaussian continuous hidden nodes [6]. Where f is normal distribution function which takes the form shown in Equation (10)

$$P(h_j = h|x) = N(c_j + w_j.x, 1) \tag{1.10}$$

Harmonium RBM is used for a discrete output in the last layer of a deep belief network in classification.

1.2.2 Deep Belief Network

The key idea behind training a deep belief network by training a sequence of RBMs is that the model parameters θ , learned by an RBM define both $p(v \mid h, \theta)$ and the prior distribution over hidden vectors, $p(h \mid \theta)$, so the probability of generating a visible vector, v, can be written as:

$$p(v) = \Sigma_h p(h \mid \theta).p(v \mid h, \theta) \tag{1.11}$$

After learning θ, $p(v \mid h, \theta)$ is kept while $p(h \mid \theta)$ can be replaced by a better model that is learned by treating the hidden activity vectors $H = h$ as the training data (visible layer) for another RBM. This replacement improves a variation lower bound on the probability of the training data under the composite model. The study in [17] proves the following three rules:

1. Once the number of hidden units in the top level crosses a threshold; the performance essentially flattens at around certain accuracy.
2. The performance tends to decrease as the number of layers increases.
3. The performance increases as we train each RBM for an increasing number of iterations.

In case of not using class labels and back-propagation in the DBN Architecture (unsupervised training) [14], DBN could be used as a feature extraction method for dimensionality reduction. On the other hand, when associating class labels with feature vectors, DBN is used for classification. There are two general types of DBN classifier architectures which are the Back-Propagation DBN (BP-DBN) and the Associate Memory DBN (AM-DBN) [8]. For both architectures, when the number of possible classes is very large and the distribution of frequencies for different classes is far from uniform, it may sometimes be advantageous to use a different encoding for the class targets than the standard one-of-K softmax encoding.

1.3 Hybrid Intelligent Intrusion Detection Scheme

This section shows DBN as a standalone classifier and the proposed DBN-SVM hybrid scheme.

1.3.1 DBN Classifier

In the paper, the Constructed DBN will be composed of two RBMs, lower and higher RBM layers. The number of visible nodes of lower RBM is attribute number and the number of hidden nodes of the higher RBM is the available class number. While the number of hidden nodes in the lower RBM layer and number of visible nodes in the higher RBM layer are the same and equal to a random number, e.g. 13. Each hidden node in the higher RBM represents one of the classes under testing, such that if ”0” is the class label associated with the input, then the first node in the hidden nodes in the higher RBM is 1, and the rest of nodes will be of value 0. e.g. If the output in the first hidden node is 0.6 and if the class label is 0, which means that the expected output in this node is 1, then there is an error of a value of 0.4. Algorithm 1 shows the steps of DBN classifier.

Algorithm 1 DBN classifier

```
Use the training dataset to train the lower RBM layer.
Used output of the lower RBM layer to train the higher RBM layer.
Test the output in the higher RBM layer hidden nodes according to the output class label.
Back-propagate the error to fix the weights of the network.
Run the complete dataset through the network to produce a reduced output of the data.
for  each object in the testing dataset do
    Run the input through the trained DBN network to produced an output in the hidden nodes of the higher RBM layer.
    Get the node of the maximum output value.
    Assign a class label that correspond to this node (of maximum output).
    if  Assigned class label is equal to actual class label then
        object is classifier correctly
    end if
end for
Calculate the sum of the correctly classified object to find the classification accuracy.
```

The training of the two restricted Boltzmann machines is required to initialize the weights of the deep belief network. So the network may have a better performance than using random weights.

1.3.2 DBN-SVM Hybride Scheme

The proposed hybrid intelligent intrusion detection network system is composed of three main phases; Preprocessing phase, DBN feature reduction phase and classification phase. Figure 1.1 describes the structure of the hybrid intelligent intrusion detection network system.

1.3.2.1 NSL-KDD Dataset Preprocessing

Pre-processing of NSL-KDD dataset contains three processes; (1) Mapping symbolic features to numeric value, (2) Data scaling, since the data have significantly varying resolution and ranges. The attribute data are scaled to fall within the range [0, 1]. and (3) Assigning attack names to one of the five classes, 0 for *normal*, 1 for *DoS* (Denial of Service), 2 for *U2R* (User to Root), 3 for *R2L* (Remote to Local) , and 4 for *Probe*.

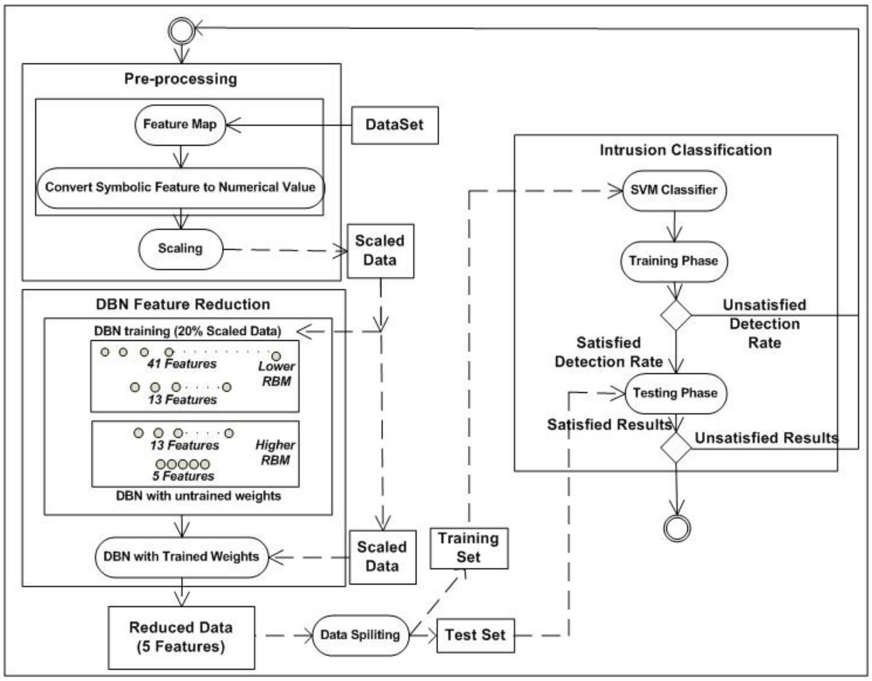

Fig. 1.1 Hybrid Intelligent Intrusion Detection Network Scheme

1.3.2.2 DBN Feature Reduction

In this paper, DBN has been used as dimensionality reduction method with back-propagation to enhance the reduced data output. The DBN Network has the BP-DBN structure that is constructed of 2 RBM layers, the first RBM layer efficiently reduces the data(e.g. from 41 to 13 feature and the second from 13 features to 5 output features based on NSL-KDD data).

1.3.2.3 Intrusion Classification

The 5 features output from the DBN where pass to the SVM classifier to be classified. SVM is a classification technique based on Statistical Learning Theory (SLT). It is based on the idea of a hyper plane classifier, where it first maps the input vector into a higher dimensional feature space and then obtains the optimal separating hyper-plane. The goal of SVM is to find a decision boundary (i.e. the separating hyper-plane) so that the margin of separation between the classes is maximized [4].

1.4 Experimental Results and Discussion

1.4.1 Dataset characteristics

The data used in classification is NSL-KDD, which is a new dataset for the evaluation of researches in network intrusion detection system. NSL-KDD consists of selected records of the complete KDD'99 dataset [18]. NSL-KDD dataset solve the issues of KDD'99 benchmark [19]. Each NSL-KDD connection record contains 41 features (e.g., protocol type, service, and flag) and is labeled as either normal or an attack, with one specific attack type. The attacks fall into four classes:

- DoS e.g Neptune, Smurf, Pod and Teardrop.
- R2L: unauthorized access to local from a remote machine e.g Guess-password, Ftp-write, Imap and Phf.
- U2R: unauthorized access to root privileges e.g Buffer-overflow, Load-module, Perl and Spy.
- Probing eg. Port-sweep, IP-sweep, Nmap and Satan.

The NSL-KDD training set contains a total of 22 training attack types, with an additional 17 types in the testing set only.

1.4.2 DBN Structure

Deep Belief network has been used in two different ways, either as a dimensionality reduction method before applying SVM as a classifier or as a classifier by itself. Support Vector machine is a parameterized method, in this paper the default parameters of SVM has been used. The RBM training is considered as weights initializer. The number of RBM structures in the used DBN is two. The number of features are 41, 13 and 4 in the first, second and last layer in the DBN Network. The number of Gipps iteration is 150. Classification is applied on different training percentage.

1.4.3 Experiments and Analysis

The NSL- KDD dataset are taken to evaluate the proposed DBN-SVM intrusion detection scheme. All experiments have been performed using Intel Core 2 Duo 2.26 GHz processor with 2 GB of RAM and weka software [21].

1.4.3.1 Case 1: DBN vs. SVM vs. DBN-SVM Scheme

A comparison between SVM, DBN and the proposed DBN-SVM scheme is shown in Table I. The classification accuracy achieved using DBN as dimensional reduction method before SVM is improved than using SVM or DBN as standalone classifier. Also the testing speed of DBN-SVM scheme is improved which is important for real time network applications. One of the conclusions that appear during experiment is that the accuracy starts to increase, when number of Gipps methods is 100 and reaches high performance then starts to deteriorate again.

Table 1.1 SVM, DBN AND THE HYRIDE DBN-SVM Scheme TESTING ACCURACY AND SPEED

Training percentage	SVM	DBN	DBN-SVM
20%	82.30 (10.4 Sec)	89.63 (0.31 Sec)	90.06 (2.54Sec)
30%	87.6 (10.4 Sec)	89.44 (0.26 Sec)	91.50 (2.54Sec)
40%	88.33 (16.67Sec)	89.54 (0.24 Sec)	92.84 (3.07 Sec)

1.4.3.2 Case 2: DBN as feature reduction method vs. different feature reduction methods

We compared the DBN as a feature reduction method with other feature reduction methods like PCA, Gain Ratio and chi square. Using DBN, PCA, Gain Ratio and chi square the 41 features of the NSL- KDD dataset is reduced to 13 features. Table II gives the testing performance accuracy of the reduced data using SVM classifier. Table II illustrate that DBN gives better performance than the other reduction methods.

Table 1.2 Performance accuracy of DBN with different feature reduction methods

Training percentage	PCA	Gain-Ratio	Chi-Square	DBN
20%	68.72	65.83	66.0	90.06
30%	68.98	65.88	65.68	91.50
40%	71.01	70.99	65.82	92.84

1.5 Conclusion

Deep Belief network has proved a good addition to the field of network intrusion classification. In comparison with known classifier and feature reduction methods, DBN provides a good result as a separate classifier and as a feature reduction method. In this paper we proposed a hybrid DBN-SVM intrusion detection scheme, where DBN is used as a feature reduction method and SVM as a classifier. We examine the performance of the proposed DBN-SVM scheme by reducing the 41-dimensional of NSL-KDD dataset to approximately 87% of its original size and then classify the reduced data by SVM. The DBN-SVM scheme shows higher percentage of classification than SVM and enhances the testing time due to data dimensions reduction. Also, we compare the performance of the DBN as a feature reduction method with PCA, Gain Ratio and Chi-Square feature reduction method.

References

1. T. Shon and J. Moon, "A hybrid machine learning approach to network anomaly detection", Information Sciences, vol.177, pp. 3799-3821, 2007.
2. J.P. Anderson, "Computer security threat monitoring and surveillance",Technical Report, James P. Anderson Co., Fort Washington, PA, April 1980.
3. W. Stallings, "Cryptography and network security principles and practices", USA: Prentice Hall, 2006.
4. C. Tsai , Y. Hsu, C. Lin and W. Lin, "Intrusion detection by machine learning: A review", Expert Systems with Applications, vol. 36, pp.11994-12000, 2009.
5. E. Biermann, E. Cloete and L.M. Venter, "A comparison of intrusion detection Systems", Computer and Security, vol. 20, pp. 676-683, 2001.
6. T. Verwoerd and R. Hunt, "Intrusion detection techniques and approaches", Computer Communications, vol. 25, pp.1356-1365, 2002.
7. K. Ilgun, R.A. Kemmerer and P.A. Porras, "State transition analysis:A rule-based intrusion detection approach" IEEE Trans. Software Eng. vol. 21, pp. 181-199, 1995.
8. D. Marchette, "A statistical method for profiling network traffic". In proceedings of the First USENIX Workshop on Intrusion Detection and Network Monitoring (Santa Clara), CA. pp. 119-128, 1999.
9. S. Mukkamala, G. Janoski and A.Sung, "Intrusion detection: support vector machines and neural networks" In proceedings of the IEEE International Joint Conference on Neural Networks (ANNIE), St. Louis, MO, pp. 1702-1707, 2002.
10. E. Lundin and E. Jonsson, "Anomaly-based intrusion detection: privacy concerns and other problems", Computer Networks, vol. 34, pp. 623-640, 2002.
11. S. Wu and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: A review", Applied Soft Computing, vol.10, pp. 1-35, 2010.
12. A. R. Mohamed, G. Dahl and G. E. Hinton, "Deep belief networks for phone recognition", NIPS 22 workshop on deep learning for speech recognition, 2009.
13. G. E. Hinton, "A fast learning algorithm for deep belief nets", Neural Computation, vol. 18, pp. 15271554, 2006.
14. A. K. Noulas and B.J.A. Krse, "Deep Belief Networks for Dimensionality Reduction", Belgian-Dutch Conference on Artificial Intelligence, Netherland, 2008.

15. H.Larochelle and Y.Bengio, ”Classification using discriminative restricted boltzmann machines”, In Proceedings of the 25th international conference on Machine learning,vol. 307, pp. 536-543, 2008.
16. L. McAfee, ”Document Classification using Deep Belief Nets”, CS224n, Sprint 2008.
17. H. Larochelle, Y. Bengio, J. Louradour and P. Lamblin, ”Exploring Strategies for Training Deep Neural Networks”, Journal of Machine Learning Research, vol.10, pp.1-40, 2009.
18. Ira Cohen, Qi Tian, Xiang Sean Zhou and Thoms S.Huang, ”Feature Selection Using Principal Feature Analysis”, In Proceedings of the 15th international conference on Multimedia, Augsburg, Germany, September 25-29, 2007.
19. KDD’99 dataset, http://kdd.ics.uci.edu/databases, Irvine, CA, USA, July, 2010.
20. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, ”A Detailed Analysis of the KDD CUP 99 Data Set” In Proceeding of the 2009 IEEE symposium on computational Intelligence in security and defense application (CISDA), 2009.
21. Weka: Data Mining Software in java http://www.cs.waikato.ac.nz/ml/weka/