Intrusion Detection using Deep Belief Network

KAMRAN RAZA*, AND SYED HASAN ADIL*

RECEIVED ON 10.06.2014 ACCEPTED ON 17.10.2014

ABSTRACT

This paper proposes an intrusion detection technique based on DBN (Deep Belief Network) to classify four intrusion classes and one normal class using KDD-99 dataset. The proposed technique is based on two phases: in first phase it removes the class imbalance problem and in the next, it applies DBN followed by FFNN (Feed-Forward Neural Network) to build a prediction model. The obtained results are compared with those given in [9]. The prediction accuracy of our model shows promising results on both intrusion and normal patterns.

Key Words: **Network Security;Deep Belief Network;Feed Forward Neural Network; Intrusion Detection.**

1. INTRODUCTION

An IDS (Intrusion Detection System) identifies any activity which is conflicting with the acceptable use of any computing resource [1]. Before the advent of modern computing techniques, this detection was done manually by the system administrators. Consequently, many opportunities were available to illegally penetrate into the system.

IDSs are mostly classified into two types (i) misuse detection, which uses predefined intrusion signatures to detect intrusion patterns; and (ii) anomaly detection, which learns the normal behavior and any variation in above defined threshold is considered as an intrusion [1-4]. Each type of IDS has its own advantages and disadvantages. Therefore, modern IDSs use a hybrid of both techniques to enhance detection rate.

Modern IDSs apply many advanced approaches to identify intrusive activities in the network. These techniques include machine learning algorithms like Neural Networks, statistical techniques like Hidden Markov Model and Rule based techniques like Decision Trees [2-5,11]. These modern techniques do not use static rules to detect intrusion but rather learn to classify intrusion and non-intrusion from the labelled dataset. Hence, it can easily detect unknown attacks without predefinition of a new rule for each variant of previously known attack. In order to compare the accuracy of IDS, researchers frequently use KDD-99 [6] dataset (i.e. a multi-class labelled intrusion dataset).

Due to the complex nature of the intrusion classification problem, we need to have a deep learning model which can correctly classify incoming network data into normal and attack classes. For deep network based models, DBN [7-8] is a proven technique successfully applied to solve many complex classification problems. DBN was proposed back in early 90s but due to the deep architecture, the model became very complex and thus required high computing power and efficient algorithm implementation which was not available at that time. Therefore, researchers were initially reluctant to use DBN; but now the easy availability of very powerful modern computing machines and efficient algorithms [7] has enabled researchers to easily utilize deep architectures in many recent works [9,11,13,15]. DBN algorithm learns at multiple levels due to which it can

* Faculty of Engineering, Sciences & Technology, Iqra University, Karachi, Pakistan

model complex functions mapping without any manual interaction. It trains single layer at a time by using an unsupervised algorithm. It gradually learns more complex patterns as it moves from initial to the final layer. DBNs have also been successfully used to initialize deep supervised feed forward neural networks [8]. Therefore, our proposed technique also uses DBN for initializing the proposed deep network.

RNN (Recurrent Neural Network) is a type of multi-layer perceptron which contains feed-forward and atleast one feed-back connection resulting in a loop like structure. Additionally, RNN also have some type of memory to deal with specific implementation of different RNN architectures [9,12]. These RNN type includes BPTT (Back-Propagation Through Time), RTRL (Real-Time Recurrent Learning), and EKF (Extended Kalman Filtering) techniques. RNN based architectures are especially ideal for temporal and sequencing related pattern recognition problems [12,15].

The rest of the paper is organized as follows: Section 2 describes the proposed technique. Section 3 describes the workflow and individual components of the proposed model. Section 4 describes the evaluation criteria. In Section 5, we compare the results of our proposed approach with existing techniques [9]. In Section 6, we present our discussions and conclusions.

2. TECHNIQUE

Table 1 shows the details about 494021 training and 292298 testing records obtained from KDD-99 corrected dataset. It is important to note that Normal and DOS belongs to major class in the dataset while Probe, U2R and R2L belong to minor class. The dataset has class imbalance problem due to the large contribution difference between major and minor classes. Due to class imbalance problem many prediction model completely failed to classify minor classes.

In order to resolve the class imbalance problem in the dataset, we applied SMOTE [10] class imbalance removing technique to generate new (i.e. synthetic) dataset. For training of the prediction model, we used the new dataset (i.e. the dataset generated by SMOTE (Synthetic Minority Over-Sampling)) as shown in Table 2.

Initialization of weight is an important factor for rapid convergence for any neural network algorithm. In case of complex classification, proper, improper or random weight may take very long or even fail to properly classify the data. Therefore, we used DBN to initialize the weights for the model before training the classification model using feedforward neural network. Our approach proves that initializing weights using DBN outperforms all the previous approaches.

Our prediction model is based on the following three basic steps:

(1) Eliminate class imbalance problem from the actual dataset using SMOTE.

(2) Estimate initial weight of the prediction model by applying DBN algorithm (Fig. 1) [7] using the following steps,
   (a) In the first step RBM is used to train the input/visible layer of the DBN model as $x = h^{(0)}$. The output generated by this step is used to train the next/hidden layer.
   (b) In the second step next/hidden layer is also trained by RBM as samples of $p(h^{(1)} | h^{(0)})$ or mean activations $p(h^{(1)} = 1 | h^{(0)})$.
   (c) Repeat the previous two steps until the weights of the DBN model converge to some fixed values.

(3) Develop actual prediction model using FFNN (Fig. 2) with the initial model estimated by DBN algorithm.

(4) Calculate the accuracy of the prediction model using test dataset.

TABLE 1. CONTRIBUTION OF EACH CLASS INSTANCE IN KDD-99

Title of Dataset	Normal Class (%)	DOS Class (%)	Probe Class (%)	U2R Class (%)	R2L Class (%)	Total
KDD 10% Corrected Training Data	97278	391458	4107	52	1126	494021
KDD 10% Corrected Test Data	60591	223298	2377	39	5993	292298

The best parameters configuration for DBN algorithm used to train the model is shown in Table 3.

The best parameters configuration of FFNN algorithm used to train the model are shown in Table 4.

3. WORKFLOW OF THE PROPOSED TECHNIQUE

Workflow of the proposed DBN based IDS is shown in Fig. 3. The proposed model is based on two phases: (I) Training Phase (II) Testing phase. The details of individual tasks performed during each phase of the workflow are discussed below:

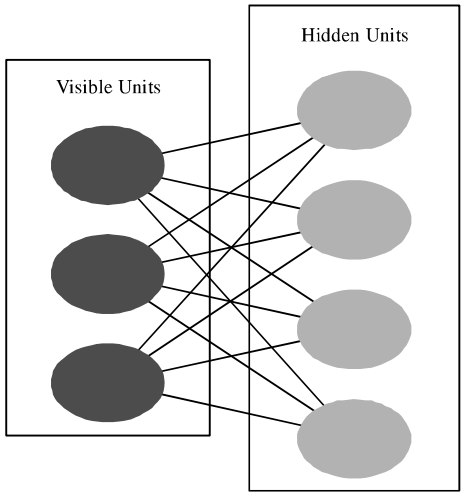

FIG. 1. RESTRICTED BOLTZMANN MACHINE

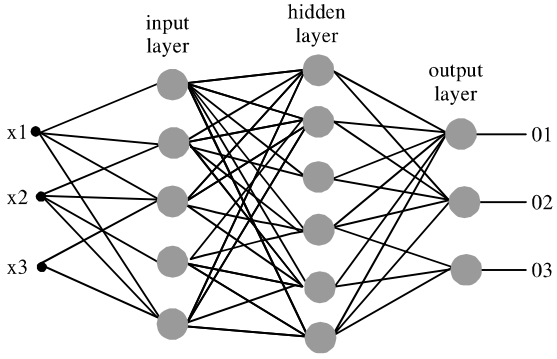

FIG. 2. FEED FORWARD NEURAL NETWORK

3.1 Training Phase

The training phase includes the development of IDS model. The inputs of this phase are training data (i.e. KDD-99 dataset) and output of this phase is the IDS model. Following paragraphs explain the individual steps involve in the training phase as shown in Fig. 3.

The IDS training dataset is used to train the model using a particular learning algorithm. We used KDD-99 training dataset to train our proposed deep belief network based model.

The KDD-99 training dataset contains various types of attributes including continuous, non-continuous and symbolic values. Each continuous attribute has its own value range which differs from the other attributes. On the other hand, each non-continuous and symbolic attribute has limited numbers of possible discrete values which also differ from one attribute to others. In this step we have performed feature scaling to all type of attributes and converted them into values in the interval [0,1]. Due to feature scaling, all attributes have

TABLE 3. SETUP PARAMETERS FOR DBN

Epochs	10
Batch Size	100
Learning Rate	1
Nodes in Input Layer	41
Hidden Layers	3
Nodes in Hidden Layer	50

TABLE 4. SETUP PARAMETERS FOR FFNN

Epochs	25
Batch size	100
Learning Rate	0.1
Nodes in Input Layer	41
Hidden Layers	3
Nodes in Hidden Layer	50
Output Layers	5

TABLE 2. CONTRIBUTION OF EACH CLASS INSTANCE AFTER SMOTE ALGORITHM

Title of Dataset	Normal Class (%)	DOS Class (%)	Probe Class (%)	U2R Class (%)	R2L Class (%)	Total
KDD 10% Corrected Training Data	559186	391458	726993	671070	726993	3075700

comparable values and the learning algorithm will not give importance to any particular attribute based on their large values.

The next step simply checks the distribution of classes in the dataset. If the dataset have uneven distribution of classes then it moves to SMOTE algorithm step which removes the class imbalance problem.

The SMOTE algorithm step removes the class imbalance problem using SMOTE algorithm. The new dataset generated after SMOTE algorithm does not suffers with this problem.

DBN first trains a layer of features that receive input directly and then treats the activations of the trained features as if they were input and learn features of features in a second hidden layer. In each step DBN actually tries to learn from most simple concept to most advanced concept and then it moves back to calculate the error between the actual feature and the estimated feature learnt using the algorithm. The algorithm performs multiple iterations to reduce the error till the desired tolerance level. The weights calculated by DBN will be used by the FFNN algorithm in the next step to build the model.

Next, the FFNN algorithm is initialized using weights obtained from the previous step (i.e. using DBN) instead of random weights which may lead towards poor model. The FFNN algorithm is then applied to training dataset to obtain a model for the proposed intrusion detection system.

After completing the training phase of the proposed technique, the trained model will move to testing phase for measuring its accuracy on the test dataset.

3.1 Testing Phase

This phase includes the testing of IDS model. The inputs of this phase are testing data (i.e. KDD-99

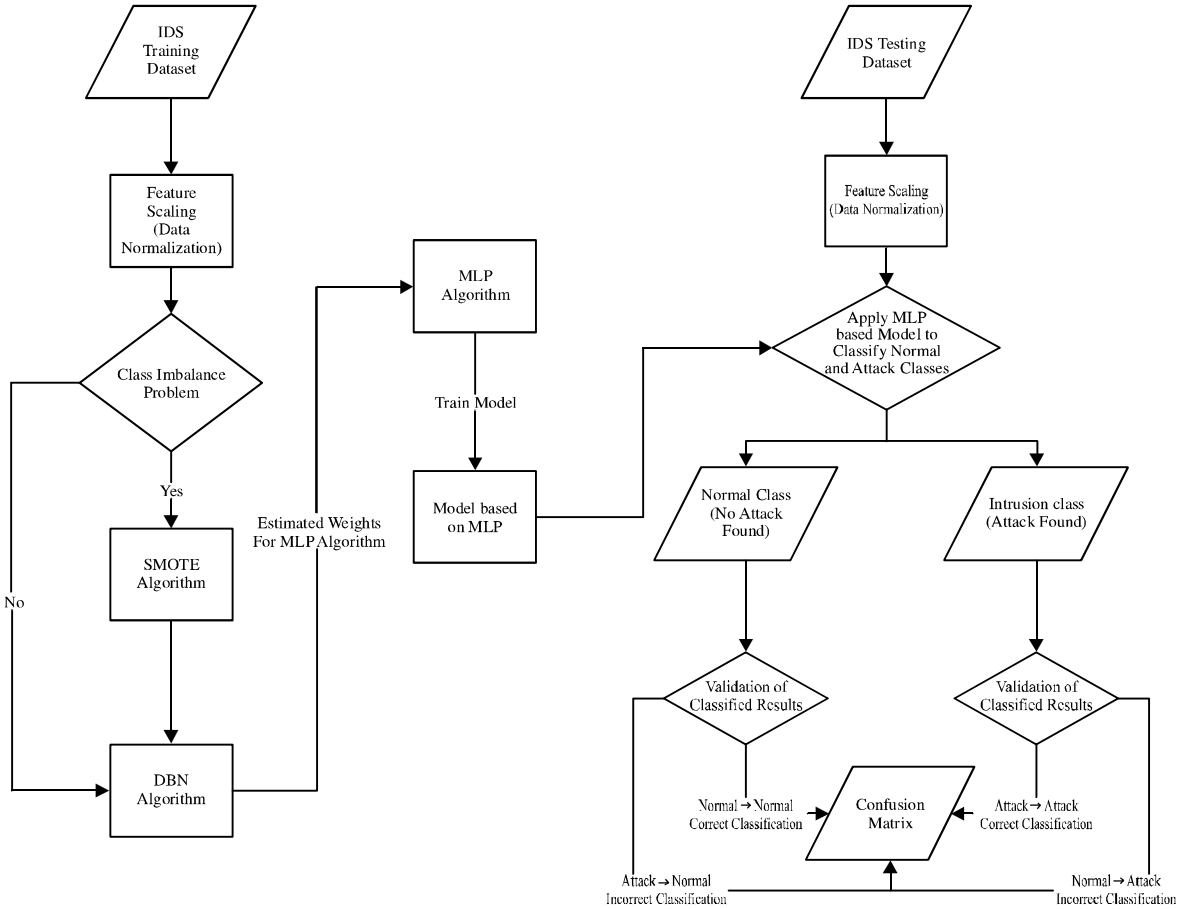

FIG. 3. WORKFLOW OF PROPOSED INTRUSION DETECTION TECHNIQUE

dataset) and the IDS model while the output is the confusion matrix. Following paragraphs explain the individual steps of the testing phase as shown in Fig. 3. We used KDD-99 test dataset to measures the accuracy of the proposed model.

The feature scaling step performs the same scaling technique on the KDD-99 test dataset as we applied on training dataset in the previous phase.

In the next step we apply FFNN model for classification which classifies the test dataset using the previously built model and estimates the accuracy of the model.

Next, we compare the actual class of each individual test instance with the corresponding class predicted by our proposed model.

In the last step, we create a confusion matrix for evaluation of our proposed model as shown in Table 5. Confusion matrix provides a simple approach for result evaluation. A good prediction model has maximum values in the diagonal and minimum values in the off-diagonal cells. For our model, confusion matrix represents True Positive if class is Normal and also classified as Normal, True Negative Class is of type Intrusion and also classified as Intrusion, False Positive if class is of type Intrusion but classified as Normal, and False Negative if class is of type Normal but classified as Intrusion.

4. EVALUATION TECHNIQUE

The proposed model is tested using three standard metrics developed specifically for evaluating IDS are FAR (False Alarm Rate), DR (Detection Rate) and CPE (Cost Per Example) [9]. The corresponding formula for these metrics are given below:

$$FAR = \frac{No.\ of\ normal\ connections\ classified\ as\ attacks}{Total\ No.\ of\ normal\ connections}$$

$$DR = \frac{No.\ of\ correctly\ \det ected\ attacks}{Total\ No.\ of\ attacks}$$

$$CPE = \frac{1}{T} * \sum_{i=1}^{m} \sum_{j=1}^{m} CM(i,\ j) * C(i, j)$$

Here, T is Total count of data instances, m is Total count of classes, i is Represents row, j is Represents column, $CM(i, j)$ is Total count of data instance classified correctly, when i is j.

Total count of data instance classified incorrectly, when i<>j.

$C(i, j)$ = Cost associated with incorrectly classified instance belonging to class i into class j, \when i<>j. Cost is zero when i = j.

5. RESULTS

The accuracy of our prediction model is mentioned in Table 5. Table 6 shows the confusion matrix which specifies the classification and misclassification performed by the model. The cost for CPE calculation is shown in Tables 7-8 we have showed comparative analysis in terms of DR, FAR and CPE between the proposed and existing techniques. The details analysis shown that our proposed technique achieved 97.1% DR which means it outperforms all other existing techniques including the previously known best DR of 94.1% [9]. Similarly, our solution achieved 0.0821 value of CPE which again outperforms all other techniques including the previous best CPE of 0.1666 [9]. However, our technique achieved 2.8% value of FAR which outperform many existing techniques but failed to achieve better result than the previous best FAR of 0.38% [9]. The detailed analysis between previously known best technique (Reduced size Recurrent Neural Network) and our proposed technique (Deep Belief Network) showed that our proposed model improved prediction accuracy in comparison with Reduced size RNN [9] (i.e. Fig. 4). Fig. 4 clearly shows that our predication model increased the overall prediction accuracy as well as correctly recognized the U2R minor class which was totally missed with Reduced Size RNN.

6. CONCLUSION

This research presented a prediction model based on DBN. FFNN is used to predict intrusion and normal data instances in KDD-99 dataset. Results have shown that our prediction model has significantly improved the classification accuracy of KDD-99 dataset. The proposed technique shows better result for DR and CPE when compared to other recent works. However, our technique is comparatively less accurate for FAR when compared with some other recent works.

TABLE 5. PREDICTION MODEL ACCURACY OF EACH CLASS ON TEST DATASET

Title of Dataset	Normal Class (%)	DOS Class (%)	Probe Class (%)	U2R Class (%)	R2L Class (%)	Total (%)
KDD 10% Corrected Data	97.2	99.1	84.8	66.7	26.6	97.1

TABLE 6. CONFUSION MATRIX OBTAINED OF EACH CLASS ON TEST DATASET

Actual\Predicted	Normal Class (%)	DOS Class (%)	Probe Class (%)	U2R Class (%)	R2L Class (%)	Total (%)
Normal Class	58906	682	419	371	213	60591
DOS Class	1215	221235	820	8	20	223298
Probe Class	354	4	2015	4	0	2377
U2R Class	8	0	0	26	5	39
R2L Class	4259	0	29	116	1589	5993
Total	64742	221921	3283	525	1827	292298

TABLE 7. COST MATRIX USED FOR CALCULATION OF FAR, DR AND CPE [9]

Actual\Predicted	Normal Class (%)	DOS Class (%)	Probe Class (%)	U2R Class (%)	R2L Class (%)
Normal Class	0	2	1	2	2
DOS Class	2	0	1	2	2
Probe Class	1	2	0	2	2
U2R Class	3	2	2	0	2
R2L Class	4	2	2	2	0

TABLE 8. COMPARATIVE ANALYSIS BETWEEN VARIOUS PROPOSED MODELS [9]

Model	DR	FAR	CPE
KDD Winner	91.8	0.60	0.2331
PNRule	91.1	0.40	0.2371
SOFM	71.6	28.37	N/A
Jordan ANN	62.9	37.09	N/A
RNN	73.1	26.85	N/A
Clustering	93	10	N/A
K-nearest neighbor	91	8	N/A
Support vector machine (SVM)	98	10	N/A
Fuzzy association rules	91	3.34	N/A
Reduced size RNN	94.1	0.38	0.1666
FFNN	80.0	0.28	0.4170
Elman	87.9	1.57	0.2972
Proposed DBN based FFNN	**97.1**	**2.8**	**0.0821**

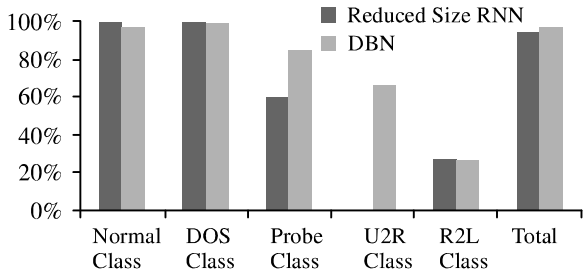

FIG. 4. COMPARATIVE ANALYSIS BETWEEN ACCURACY OF REDUCED SIZE RNN AND DBN-FFNN

ACKNOWLEDGMENT

The authors acknowledge the support of Faculty of Engineering, Science and Technology, Iqra University, Karachi, Pakistan, for providing research facilities and financial assistance.

REFERENCES

[1] Kemmerer, R.A., and Vigna, G., "Intrusion Detection: A Brief History and Overview", Computer, Volume 35, No. 4, pp. 27-30, 2002.

[2] Srinivasulu, P., Nagaraju, D., Kumar, P.R., and Rao, K.N., "Classifying the Network Intrusion Attacks using Data Mining Classification Methods and their Performance Comparison", International Journal of Computer Scince and Network Security, Volume 9, No. 6, June, 2009.

[3] Davide, A., Roberto, T., and Giorgio, G., "HMMPayl: An Intrusion Detection System based on Hidden Markov Models", Computers & Security, Volume 30, pp. 221-241, 2012.

[4] Nagaraju, D., Srinivasulu, P., Kumari, V.V., and Govardhan, A., "Intrusion Detection System uing Bayesian Network and Hidden Markov Model", C3IT, Procedia Technology, Volume 4, pp. 506-514, 2012.

[5] Nadeem, Q., and Kamran, R., "Effect of Feature Selection, Synthetic Minority Over-Sampling (SMOTE) and Under-Sampling on Class Imbalance Classification", Proceedings of 14th International Conference on Modelling and Simulation, pp. 145-150, 2012.

[6] KDD Cup 1999. Available on: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. January 2014.

[7] Hinton, G.E., Osindero, S., and Teh, T., "A Fast Learning Algorithm for Deep Belief Nets", Neural Computation , Volume 18, pp. 1527-1554, 2006.

[8] Bengio, Y., "Learning Deep Architectures for AI", Technical Report 1312, Universite de Montreal, 2007.

[9] Mansour, S., Zahra, J., and Ali, F., "Intrusion Detection Using Reduced-Size RNN Based on Feature Grouping", Neural Computing and Applications, Volume 21, No. 6, pp. 1185-1190.

[10] Nathalie, J., and Shaju, S., "The Class Imbalance Problem: A Systematic Study", Intelligent Data Analysis, Volume 6, pp. 429-449, IOS Press,.

[11] Feng, W., Zhang, Q., Hu, G., and Huang, J.X., "Mining Network Data for Intrusion Detection through Combining SVMs with Ant Colony Networks", Future Generation Computer Systems, Volume 37, pp. 127-140, 2014.

[12] Jaeger, H., "Tutorial on Training Recurrent Neural Networks, Covering BPPT, RTRL, EKF and the Echo State Network Approach", GMD-Forschungszentrum Informationstechnik, 2002.

[13] Kuremoto, T., Kimura, S., Kobayashi, K., and Obayashi, M., "Time Series Forecasting Using a Deep Belief Network with Restricted Boltzmann Machines", Neurocomputing, Volume 137, pp. 47-56, 2014.

[14] Sarikaya, R., Hinton, G.E., and Deoras, A., "Application of Deep Belief Networks for Natural Language Understanding", IEEE/ACM Transactions on Audio, Speech & Language Processing, Volume 22, No. 4, pp. 778-784, 2014.

[15] Nascimento, J.C., Silva, J.G., Marques, J.S., and Lemos, J.M., "Manifold Learning for Object Tracking with Multiple Nonlinear Models", IEEE Transactions on Image Processing, Volume 23, No. 4, pp. 1593-1605, 2014.