INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 12(5), 677-688, October 2017.

A New Deep Learning Approach for Anomaly Base IDS using Memetic Classifier

S. Mohammadi, A. Namadchian

Shahriar Mohammadi*, Amin Namadchian
Department of Industrial Engineering
aK.N. Toosi University of Technology,
Tehran, Iran
*Corresponding author: Mohammadi@kntu.ac.ir
anamadchian@mail.kntu.ac.ir

Abstract: A model of an intrusion-detection system capable of detecting attack in computer networks is described. The model is based on deep learning approach to learn best features of network connections and Memetic algorithm as final classifier for detection of abnormal traffic.
One of the problems in intrusion detection systems is large scale of features. Which makes typical methods data mining method were ineffective in this area. Deep learning algorithms succeed in image and video mining which has high dimensionality of features. It seems to use them to solve the large scale of features problem of intrusion detection systems is possible. The model is offered in this paper which tries to use deep learning for detecting best features.An evaluation algorithm is used for produce final classifier that work well in multi density environments.
We use NSL-KDD and Kdd99 dataset to evaluate our model, our findings showed 98.11 detection rate. NSL-KDD estimation shows the proposed model has succeeded to classify 92.72% R2L attack group.
Keywords: Deep learning, KDD99, memetic algorithm, NSL-Kdd, classification function, anomaly base intrusion detection, intrusion-detection system (IDS).

1 Introduction

In recent years, security industry has been actively playing roles in dealing with security threats against computer organizations and networks through employing various technologies such as encryption, authentication, and access control. However, all these technologies are also involved with their specific limitations, allowing an attacker to enter the system.IDSs [1] are security mechanisms which intelligently monitor computer and network systems in real time to detect intrusions and take quick appropriate measures in response [3].

Anomaly- and signature-based are two main methods used in IDSs. The former is based on the statistical description of users or application programs which is ultimately intended to detection any activity deviating from the profile of normal behavior, which in fact is an indication of an abnormal behavior conducted by users or application programs [7].

Signature-based IDSs work based on collecting and storing the signature of known attacks, and the IDS attempts to search the logged events for patterns matching the signature of stored attacks. Both methods have their specific advantages and drawbacks. Signature-based systems have an appropriate accuracy in detection of known attacks and generate few number of false positives. On the other hand, they are only capable of detecting previously modeled attacks. On the contrary, anomaly-based IDSs are capable of detecting new attacks, but also produce a large number of false positives and may identify a normal behavior as suspicious due to deviation from the defined threshold. Another challenge in using these systems is their difficulty in adapting to dynamic environments [10].

[1] Intrusion detection systems

Many challenges are faced in order to improve the IDSs. The large number of application programs has led to extensive features in describing normal behavior. Moreover, there are more complex attacks developed everyday taking advantage of several vulnerabilities in different application programs. As a result, many parameters of high dimensions are involved in the IDSs. As a solution, attempts were made in many studies to further simplify the problem through selecting effective features and decreasing their numbers [8].

Different studies attempted to select appropriate features for intrusion detection and to develop a learning model through an algorithm. [1]. classified these studies into four categories, namely classification, clustering, statistical, and information theory. In [12, 14], cross-entropy analysis was employed to derive the appropriate features for each attack type. All the mentioned methods are highly dependent on the dataset as they attempts to extract the appropriate features according to those of the dataset.

Similar to intrusion detection, we are also faced with the problem of selecting appropriate features from among a large number of features in the real environment in applications such as image, video, and sound processing. Recent studies have managed to achieve acceptable results in feature selection through a deep-learning approach [15].

The main idea behind deep-learning is the assumption that data are composition of factors or features created in a hierarchical manner. Many other general assumptions can further improve deep learning. These seemingly simple assumptions allow exponentially finding relationships between some of the regions and samples. This can be a solution to some of the high-dimensional challenges of the problem [6].

In the proposed algorithm, We used a deep learning algorithm for regression function is obtained through deep Auto-encoder.Then use Memetic algorithm to generate a liner classification function to detect attack. This algorithm helps system to bypass local minima and become convergent, faster. Therefore, unlike papers which used genetics [14], [13] we used all KDD training datasets and enhanced our precision as well.

The architecture and components of proposed system are introduced in section 2. Section 4 deals with deep learning algorithm and its characteristics for obtain linear classification function for each class. In section 5 you know how Memetic algorithm works to combine results of classification functions is implied. and finally discussion and conclusion are presented in section 7.

2 The proposed algorithm

Our proposed algorithm is composed of three phases. In first phase we prepare and normalized dataset.In second phase we use an deep learning Auto-Encoder model to produce a regression function and in last phase we use an Memetic for produce a classifier function by those model.As shown in Fig 1 our dataset compose from three parts: training and validation part for learning Auto-Encoder model and test part for evaluation the proposed algorithm.Each record in dataset represent by data and it's label, normalized data feed into input layer of deep Auto-encoder model, and labels are assigned as input the stochastic guardian descent to find optimal model that fit with validation data. After that, we apply Memetic algorithm on reached features as final classification function that can detect normal and abnormal traffic. Finally at test phase, this model is evaluated with test data.The details of proposed architecture will be in the next sections.

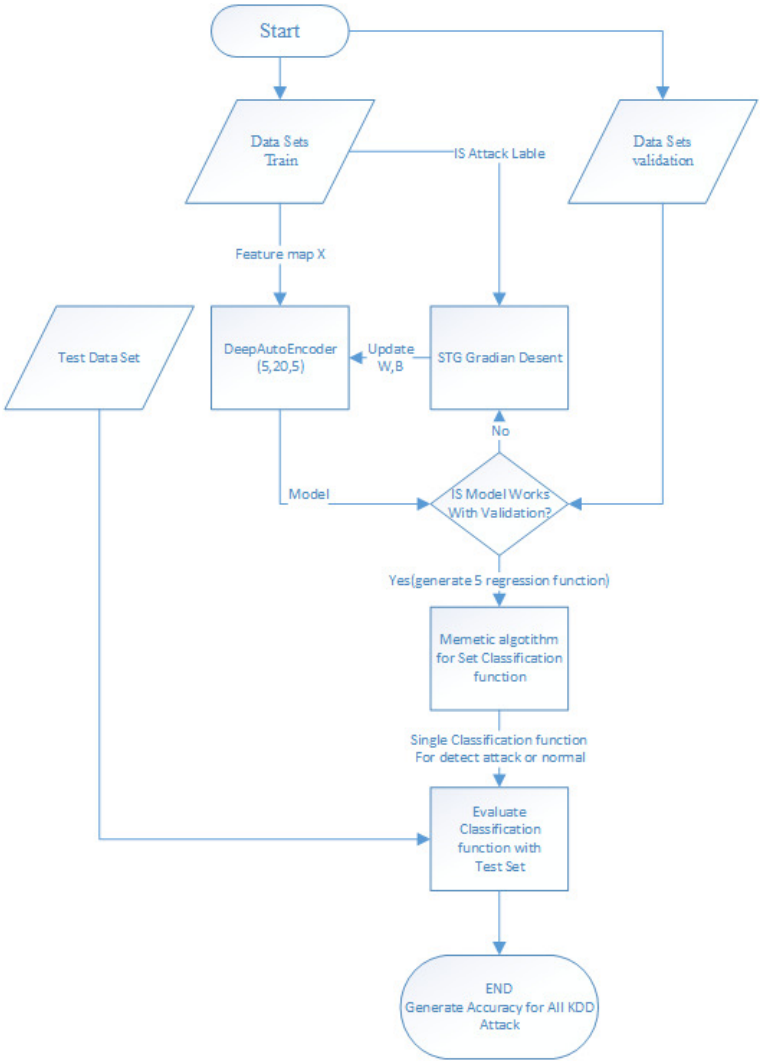

Figure 1: Architecture of the proposed algorithm

In first phase we must prepare our data for learning. For this purpose all extracted features from data must be normalized to real number with method that is described in section 3.

In next phase we train a deep learning for producing a model as regression function that can assign attack score which is explained with detail in section 4.

Our objective in the third phase is to combine the mentioned functions to learn classification that is able to report an attack is occurred or not occurring. For generating this function, the Memetic algorithm is used. This algorithm works based on an evaluation function.

After obtaining a proper classification, system may isolate attacks from ordinary traffic through this function. KDD99 dataset was used to test this algorithm.

3 Normalize dataset

The quality of analyzed data plays a significant role in enhancing the precision of data mining algorithms. We use a gradient descent optimization algorithms in our deep neural network model then need data with zero mean and equal variance. Z-score is one method that have a distribution with a mean of zero and a standard deviation of one and perfect for normalization.X is feature data, μ is mean and σ is standard deviation in equation 1.

$$X_{normalized} = \frac{X - \mu}{\sigma} \tag{1}$$

KDD use text and number. For text feature we assign a number for every value of text can give features. For example for service feature we use 0 for Ftp and 1 for Http and so on. After that all feature has a number value. We use this formula to assign a number between -1 and 1 to each features.

4 Deep learning approaches

Deep learning is a branch of machine learning based on several layer of non-linear operations that attempt to provide high-level abstractions for complex data. Several algorithms can be building blocks that put on each other as stacks to shape a deep architecture like: Convolutional neural networks, Deep Belief Network, Boltzmann machine, Restricted Boltzmann machine, neural networks, Auto-encoder, Gated Auto-encoder; and many models of this group are based on unsupervised learning.The output of each layer becomes the input of the next layer; therefore features that learned in upper layers are more abstracted [6].

The aim of an Auto-encoder is to learn a compressed representation for a set of data, typically for the purpose of dimensionality reduction. Auto-encoder is based on the concept of Sparse coding [11]. AE can be considered as a discriminative DNN in which the target output would be similar to the input, and the number of hidden layer nodes is lower than input. Therefore, it can be an unsupervised or supervised method. Its training measure is usually composed of two terms: minimization of the construction error and regularization policy. Considering different relations for regularization, various types of Auto-encoders such as de-noising AE, contractive AE, sparse AE can be created [4, 5, 9]. After training of the network is finished the hidden layer result considered as compressed representation of the input data.

We have five group data in KDD include attack and normal traffic. We use deep Auto-encoder that encode 41 features to 5 features group that show the score of is member of five group Dos , Normal , Probe , U2R , R2L. As shown in Fig 2, we design an Auto-encoder deep model composed of three layers of encoder-decoder and a regression at last layer. This supervised model received vectors with 41 dimension as input and attempt to learn a representation that can discriminant them according to their label. After training of this neural network is done, we have a feature vector with five dimension at last hidden layer of encoders as representation of input data attack score.This encoders can work as a regression function.

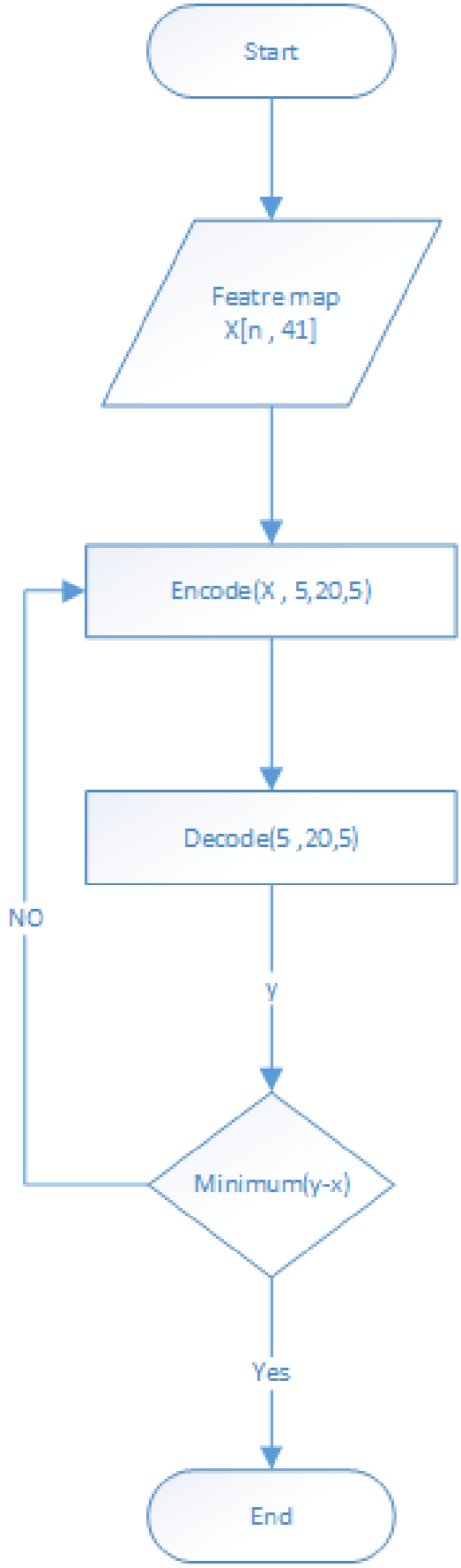

Figure 2: Architecture of the auto-encoder deep learning for generate regression function for five groups

Deep Auto-encoder can learn features as well as regression function.So we can use this algorithm without any feature selection.Our goal in this step is find best representation of data for five group in KDD after this step we need to combine this functions and generate one classification function.

5 Memetic as final classifier

Whenever connection information is bestowed to Auto-encoder of each class, it may have different features due to the mentioned connection belonging to attacks or normal traffic. Thus, we need an algorithm which concludes such results and makes decision about normal or abnormal condition of a connection. To do this, a linear memetic classifier were employed.

In the training phase, we evaluated data of each connection using regression function of each class and map to five numbers that show score of attack probabilities Then this new features is given to the Memetic classifier as input.We try in Memetic algorithm learn from high level features that show attack probabilities final decision about the normal or attack traffic.

As it can be seen in Fig 1 in our proposed algorithm has been used to obtain linear classification functions Values of each gene are coefficients of linear classifier function and are in the range of (-1023,1023). Evaluation function for each chromosome is as 2 which reflects classification precision:

$$CR = \frac{TP + TN}{SizeofDataSet} \tag{2}$$

Algorithm 1 Architecture of the memetic algorithm for combine five regression function for detect normal or attack

Data: Training Data Set
Result: Memetic classifier for intrusion detection
Initializes P randomly (P = Population);
Perform local search and find best fitness in its neighbourhood explored chromosome of each individual in P;
repeat
  Select parents from P;
  Generate offspring applying recombination To the parent selected;
  if an individual is selected to undergo mutation **then**
    Then apply local search;
  end if
  Evaluate fitness of current individual and its neighbours;
  Adopt best chromosome;
until best chromosome $fittness > \sigma$

For local search in algorithm 1 we use a simple local search algorithm like hill claiming, We try to find best fitness by explore neighbourhood of chromosome by changing the gene value.

Where σ (a double value that can be in 0,1 range) which shows precision rate. The more this value is close to 1 result more take runtime and the more precise classification functions. The fitness function for algorithm 1 can compute equation 1 shows in 2.

Algorithm 2 Fitness function for memetic algorithm

Data: individual chromosome gene[0-n] and learning dataset
Result: fitness of individual
sum=0;
for all connection record k in learning data set **do**
  $SelectedFeature_k$= deepEncode(x);
  if connecction k is attack **then**
    if $\sum_{i=0}^{n-1} gene_i * Normalized(k) < gene_n$ **then**
      sum = sum + 1
    end if
  else
    if $\sum_{i=0}^{n-1} gene_i * Normalized(k) >= gene_n$ **then**
      sum = sum + 1
    end if
  end if
end for
$fittness = \frac{sum}{SizeofDataSet}$;

As you can see in algorithm 2 if correction rate for a classification function compute as fitness each classification function is "N" gens that multiple to normalize(according section 3 method) feature of connection if the compute value less than last gen value classified as attack otherwise classified as normal.

6 Evaluation result

For evaluating intrusion detection, a proper dataset should be selected in the first phase which either meet the necessary standards or be comparable with other works. Kd99 [17] is a proper dataset. [16] produce a dataset called NSL-Kdd [2] was introduced that have some benefit to evaluate intrusion.We use both KDD and NSL-KDD datasets to test our algorithm in order to both preserve our work's comparability with its previous counterparts and to negate some drawbacks of KDD99 in our evaluation.

We use correction rate percent that show in equation 2 on testing part of KDD99 For evaluate our model. Results of the proposed algorithm are presented and compared with the similar algorithm in table 1.

Our deep Auto-encoder have four layer in each encoder and decoder phases. Memetic algorithm parameters is number of population, mutation probability and crossover probability were selected as 100, 0.03 and 0.9 respectively and DSCG local searching function was used in our simulation.

We compare our algorithm with four algorithms on KDD99 dataset in table 1 on the first column we use only Memetic algorithm without deep Auto-encoder on our model as you can see result improve with select five best regression function for all group. More abstraction and manifold learning feature of the deep Auto-encoder and diminution reduction cause better result. Tao Xia, et al. [18] try to use genetic with information theory on KDD99 we compare with their result on column "GA" you can see 42.49 percent better result in R2L attack. We have a little records on train for R2L and as result show feature generation and manifold learning of deep Auto-encoder can solve this problem better than information theory.Nahla Ben Amor, et al. [2]

[2]http://nsl.cs.unb.ca/NSL-KDD/

use the native Bayesian network and show its better result than decision Tree. Their result shows on "Native Bayes" and "Decision Tree" columns to show deep learning and Memetic have very competitive results than this ordinary neural networks.

Table 1: Comparison of our CR results with relevant studies

Class	our(deep learning)	Memetic	GA	Native Bayes	Decision Tree
Normal	98.11	97.22	98.34	97.68	99.50
DOS	98.75	98.4	99.33	96.65	97.24
Probe	83.34	81.23	93.95	88.33	77.92
R2L	48.35	42.23	5.86	8.66	0.52
U2R	74.28	66.22	63.64	11.84	13.60

The number of selected NSL-Kdd records is based on the classification hard scattering in kdd99, thus training algorithms precision is analyzed in a larger and more precise span. In NSL-KDD, numbers of records in both test and training sets are reasonable, thus intricate methods would be implemented without a random selection of dataset records [16].KddTrain+, was used for training phase and KddTest+, was used to evaluate the algorithm.Seven algorithms run three times on each record on NSL-KDD.SuccessfulPrediction is a number in the range of 0 to 21 which indicates how many time that algorithm be to succeed on correct detection that record.SuccessfulPrediction field is a criterion to detect difficulty of classification of present records of NSL-Kdd.

In Fig 3 - 8, system precision in terms of SuccessfulPrediction field is shown. For instance, in Fig 6 for R2L class, our system has succeeded to classify 90.72% of records correctly. It is worthy to say that 0 in the mentioned field indicates that all seven algorithms have failed to classify these records in all three times.

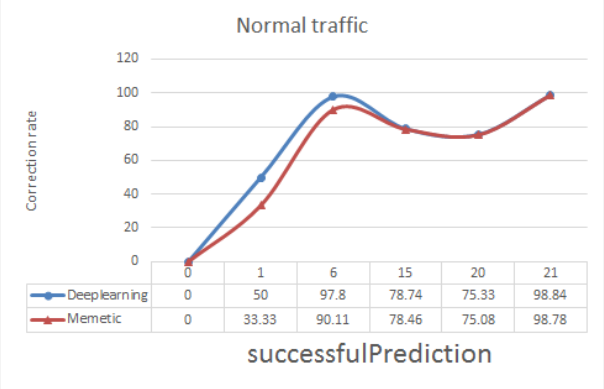

Figure 3: Comparison correction rate on NSLKDD normal attack our model with memetic algorithm

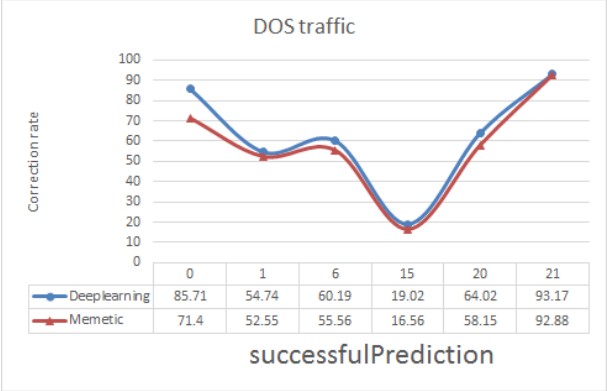

Figure 4: Comparison correction rate on NSLKDD DOS attack our model with memetic algorithm

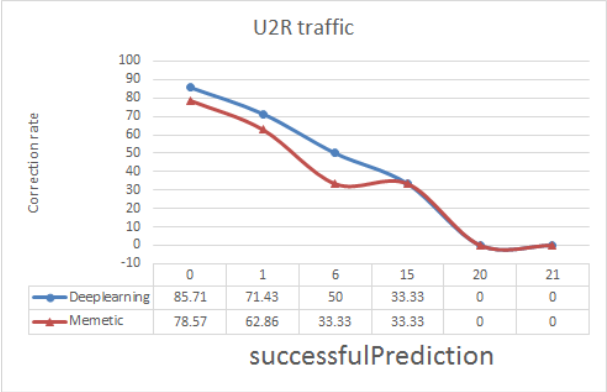

Figure 5: Comparison correction rate on NSLKDD U2R attack our model with memetic algorithm

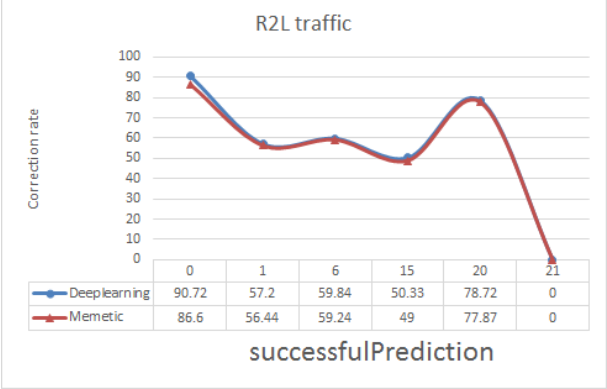

Figure 6: Comparison correction rate on NSLKDD R2L attack our model with memetic algorithm

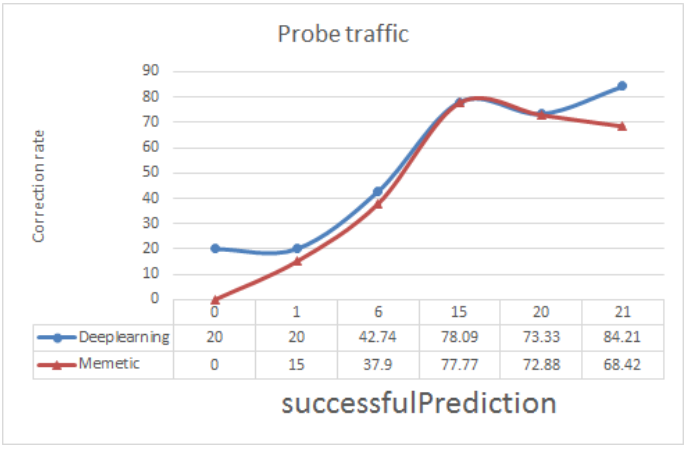

Figure 7: Comparison correction rate on NSLKDD probe attack our model with memetic algorithm

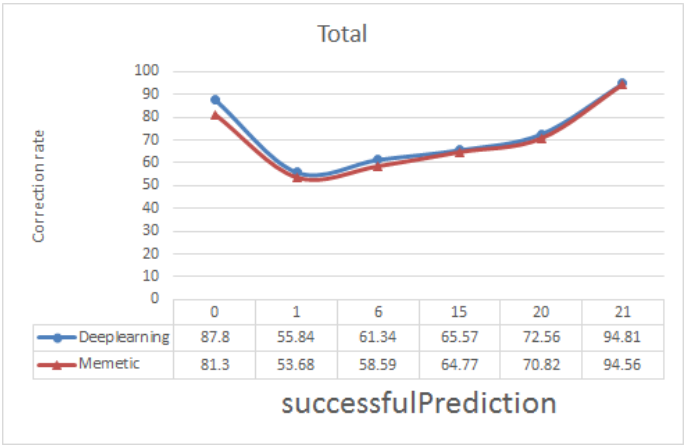

Figure 8: Comparison correction rate on all NSLKDD groups our model with memetic algorithm

7 Conclusion

Anomaly-based intrusion detection system was presented which can learn feature of attacks using DNN-Auto-Encoder and use the memetic algorithm for final classifier. The performance of present model show it can detect most of attacks correctly. Deep learning, which is proper for large scale features and Memetic algorithm which can solved local minimum optimization. In order to evaluate the performance of the proposed algorithm, results were obtained on KDD and NSL-Kdd datasets.

This study aimed at offering a deep learning model for anomaly detection engine. However future works include using deep recurrent network for on-line model. For example deep LSTM model for add signature-based intrusion detection systems as a supervisor on operational environment.

Signature-based intrusion detection systems have a trivial false positive rate but their detection rate is very low because they are able to detect only those attacks which follow their pattern. Therefore, when a signature-based intrusion detection system recognizes a connection

as a single attack, it is reliable mostly and it may even be used as an attack label to train an LSTM detection system.

Bibliography

[1] Ahmed M., Naser Mahmood A.,Hu J. (2016); A survey of network anomaly detection techniques, *Journal of Network and Computer Applications*, 60, 19–31, 2016.

[2] Amor N. B., Benferhat S., Elouedi Z. (2004); Naive bayes vs decision trees in intrusion detection systems, *Proc. of the 2004 ACM Symposium on Applied Computing, NY, USA. ACM*, 420–424, 2004.

[3] Axelsson S. (2000); Intrusion Detection Systems : A Survey and Taxonomy, *Computer Engineering*, 1–27, 2000.

[4] Bengio Y. (2013); Deep learning of representations: Looking forward, *Intl. Conf. on Statistical Language and Speech Processing*, 1–37, 2013.

[5] Bengio Y., Courville A. C., Vincent P. (2012); Unsupervised feature learning and deep learning: A review and new perspectives, *CoRR, abs/1206.5538*, 1, 2012.

[6] Bengio Y., Goodfellow I. J., Courville A. (2016); *Deep Learning*, The MIT Press, 2016.

[7] Bhuyan M. H., Bhattacharyya D. K., Kalita J. K. (2014); Network anomaly detection: methods, systems and tools, *Communications Surveys & Tutorials, IEEE*, 16(1), 303–336, 2014.

[8] Dang Y., Wang B., Brant R., Zhang Z., Alqallaf M., Wu Z. (2017); Anomaly detection for data streams in large-scale distributed heterogeneous computing environments, *ICMLG2017 5th Intl. Conf. on Management Leadership and Governance*, 121-121, 2017.

[9] Erhan D., Manzagol P.-A., Bengio Y., Bengio S., Vincent P. (2009); The difficulty of training deep architectures and the effect of unsupervised pre-training, *Artificial Intelligence and Statistics*, 153–160, 2009.

[10] García-Teodoro P., Díaz-Verdejo J., Maciá-Fernández G., Vázquez E. (2009); Anomaly-based network intrusion detection: Techniques, systems and challenges, *Computers & Security*, 28(1-2), 18–28, 2009.

[11] Ng A. (2011). Sparse autoencoder, *CS294A Lecture Notes*, 72, 1–19, 2011.

[12] Nguyen H., Franke K., Petrović S. (2010); Improving effectiveness of intrusion detection by correlation feature selection, In *ARES 2010 - 5th Intl. Conf. on Availability, Reliability, and Security*, 17–24, 2010.

[13] Owais S., Snasel V., Kromer P., Abraham A. (2008); Survey: Using Genetic Algorithm Approach in Intrusion Detection Systems Techniques, *2008 7th Computer Information Systems and Industrial Management Applications*, 300–307, 2008.

[14] Qu G., Hariri S., Yousif M. (2005), A new dependency and correlation analysis for features, *IEEE Transactions on Knowledge and Data Engineering*, 17(9), 1199–1206, 2005.

[15] Schmidhuber J. (2015), Deep learning in neural networks: An overview, *Neural Networks*, 61, 85–117, 2015.

[16] Tavallaee M., Bagheri E., Lu W., Ghorbani A. A. (2009), A detailed analysis of the KDD CUP 99 data set, *IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009*, 2009.

[17] University of California, I. KDD Cup 1999, 1999.

[18] Xia T., Qu G., Hariri S., Yousif M. (2005), An efficient network intrusion detection method based on information theory and genetic algorithm, *Performance, Computing, and Communications Conference, 2005. IPCCC 2005, 24th IEEE Intl.*, 11–17, 2005.