An Improved Genetic Algorithm and Deep Trust System for Intrusion Detection Based on IoT

[1]U Durga Premchand, [2] Dr S. Jhansi Rani
[1]M.Tech Scholar, Department of Computer Science and System Engineering,
[2]Department of Computer Science and System Engineering,
Andhra University College of Engineering (A), Visakhapatnam, AP, India.

Abstract: With the advent of the Internet of Things, the security of the network layer in the Internet of Things is getting more and more attention. Traditional intrusion detection technologies cannot be well adapted in the complex Internet environment of the Internet of Things. Therefore, it is extremely urgent to study the intrusion detection system corresponding to today's Internet of Things security. This paper presents an intrusion detection model based on improved Genetic Algorithm and Deep Trust System. Facing different types of attacks, through multiple iterations of the GA, the optimal number of hidden layers and number of neurons in each layer are generated adaptively, so that the intrusion detection model based on the DTS achieves a high detection rate. Finally, the NSL-KDD dataset was used to simulate and evaluate the model algorithm. Experimental results show that the improved intrusion detection model combined with DTS can effectively improve the recognition rate of intrusion attacks and reduce the complexity of the network.

Index Terms **– Internet of Things security; Intrusion detection; Deep Trust System; Genetic Algorithm.**

I. INTRODUCTION

With the rapid development, Internet of Things (IoT) technology has been widely used, from traditional equipment to common household appliances, which has greatly improved our quality of life [1]. However, IoT systems have become an ideal target of cyber attackers because of its distributed nature, large number of objects and openness [2-5]. In addition, because many IoT nodes collect, store and process private information, they are apparent targets for malicious attackers [6]. Therefore, to maintain the security of the IoT system is becoming a priority of the successful deployment of IoT networks [7]. To detect intruders is one important step in ensuring the security of the IoT networks. Intrusion detection is one of several security mechanisms for managing security intrusions, which can be detected in any of four layers of IoT architecture shown in Fig. 1. The Network Layer not only serves as a backbone for connecting different IoT devices, but also provides opportunities for deploying network-based security defence mechanisms such as Network Intrusion Detection Systems (NIDS) [8].

There are many intrusion detection methods, such as methods based on statistical analysis [9], cluster analysis [10], artificial neural network [11] or deep learning [12]. Among these methods, intrusion detection based on deep learning performs better than other methods [13]. The reason is that deep learning has strong abilities, such as self-learning, self-adaptation, good generalization, and detection against unknown attack behavior. For the deep learning algorithm, a network structure may have great detection accuracy for one attack type, but it may not have a good detection effect when facing other attacks. Therefore, we hope to design a self-adaptive model to change the network structure for different attack types, so that our intrusion detection model can maintain a high detection rate continuously.

In this paper, a new IoT intrusion detection model is proposed by introducing genetic algorithm into deep Trust System to optimize the number of hidden layers and neurons in a hidden layer. By applying the improved genetic algorithm, for different types of attacks, the optimal number of hidden layers and neurons in a hidden layer can be iteratively generated, and the network complexity can be reduced as much as possible while ensuring the detection rate. The solution of these two problems of deep network can make the intrusion detection system have a greater improvement in performance. Therefore, after the number of hidden layers and the number of neurons in each layer in DTS are determined, the DTS with the obtained most optimal network structure will be used for intrusion detection.

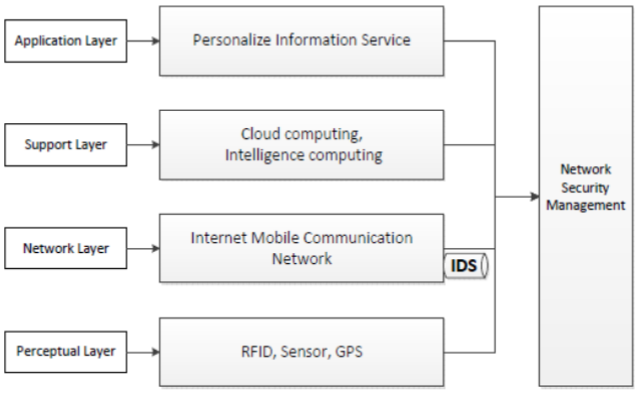

Fig. 1 IoT Architecture

II. RELATED WORK

The intrusion detection technology based on machine learning method can be divided into two major categories: intrusion detection based on artificial neural networks and intrusion detection based on deep learning [15].

Intrusion detection based on artificial neural network is generally divided into three sub-categories of neural networks: supervised, unsupervised and hybrid. The main type of supervised neural networks are multilayer feed-forward (MLFF) neural networks. Ryan et al. [16] used MLFF neural network to detect anomaly based on user behavior. However, supervised neural networks depend on training of a large number of data sets. Sometimes the distribution of training data sets is not balanced, which makes the MLFF neural network easily reach the local minimum value, and thus the stability is low. Detection rate of low-frequency attack is a key factor in judging the quality of the detection model. The detection accuracy of MLFF neural network is low for low-frequency attacks.

The main advantage of the unsupervised artificial neural networks is that new data can be analyzed without tagging data in advance. Yu et al. [17] introduced a theoretical foundation for combining individual detectors with Bayesian classifier combination. This ensemble is fully unsupervised and does not require labelled training data, which in most practical situations is hard to obtain.

The Self-Organizing Feature Map (SOM) used in [18] is an unsupervised learning method that extracts features from normal system activity and identifies statistical changes from normal trends. However, for low-frequency attacks, the detection accuracy of unsupervised neural network is also low.

The third category is the hybrid neural network, e.g., FC-ANN proposed in [19] is such a model. The FC-ANN method introduces fuzzy clustering techniques into general artificial neural networks. Using fuzzy clustering techniques, the entire training set can be divided into small, low-complex subsets. Therefore, based on these subsets, the stability of the individual neural network can be improved and the detection accuracy can be improved as well, especially for the detection of low-frequency attacks. Ma et al. [20] proposed a novel approach called SCDNN, which combines spectral clustering (SC) and deep neural network (DNN) algorithms. It provides an effective tool of study and analysis of intrusion detection in large networks. Chiba et al. [21] proposed a cooperative and hybrid network intrusion detection system (CH-NIDS) to detect the attacks by sensing the network traffic.

In [22], based on Back Propagation neural networks (BPNN), a discussion was made on the selection of the number of hidden layers. It is believed that the training set must be analyzed before the design of the neural network to correctly estimate the similarity between the number of neurons and the number of hidden layers.

At present, there are many intrusion detection technologies based on deep learning. Yin et al. [23] proposed a deep learning approach for intrusion detection using recurrent neural networks (RNN-IDS) which is Suitable for high-precision classification model modelling. Abolhasanzadeh [24] proposed a method for detecting attacks in big data using Deep Auto-Encoder. Gao et al. [25] trained the deep Trust System (DTS) as a classifier to detect intrusions. Similarly, Alom et al. also utilized the capabilities of DTS to detect intrusions through a series of experiments.

Compared with traditional neural networks, DTS has the advantages of multi-layer structure and pre-training with fine-tuning learning methods. These advantages enable DTS to extract the deep attributes of training set, thus the problems existing in the traditional neural network intrusion detection methods are solved, such as low training efficiency, easy to fall into the local optimum and the need of large amount of tag data

Genetic Algorithms is a method to search for an optimal solution by simulating natural evolution processes, but is often neglected when choosing the optimal network structure. In this paper, in order to solve the low detection rate and weak stability of the detection model caused by low-frequency attacks, we propose an intrusion detection model based on an improved Genetic Algorithm (GA) and Deep Trust System, for Different training types including low-frequency attacks and other types of attacks. The corresponding different network structures are obtained by iterative evolution through GA, thereby detection rate is improved

III. RESTRICTED BOLTZMANN MACHINES

Deep Trust System (DTS) is a kind of deep learning structure. As shown in Fig. 2, it is composed of multiple Restricted Boltzmann Machines (RBMs), mainly executing unsupervised learning of Pre-processed data, processing and abstracting the high-dimensional data. [33]

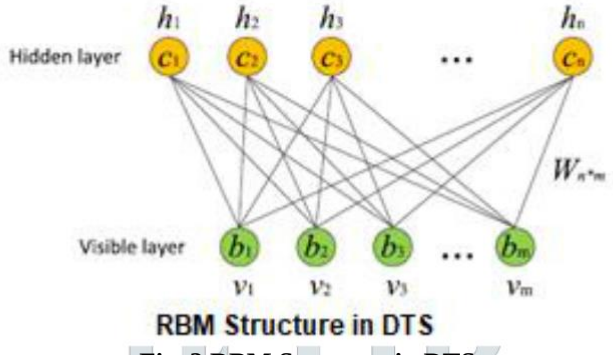

Fig. 2 RBM Structure in DTS

IV. GA OPTIMIZED DTS MODEL

This paper presents an intrusion detection model by a combined GA and DTS. Through multiple iterations of the GA, an optimal network structure is produced. The network structure contains the number of hidden layers and the number of neurons in each layer. This structure is then applied to deep Trust System for intrusion detection.

A. Improved genetic algorithm

Genetic Algorithm is known to be an ideal technique for finding optimal solutions to various problems.

1） Population initialization

The purpose of initialization is to generate an initial population randomly for subsequent genetic manipulation. For a simple training set, up to three hidden layers are enough to get a good detection rate. Binary coding is the most common coding method in genetic algorithm, so we encode the number of nodes in the three hidden layers directly in the binary chromosome. The length of chromosome is 18 bits: the first 6 bits are reserved for the first hidden layer, the subsequent 7-12 bits and 13-18 bits are for the second and the third hidden layers respectively

2） Improved selection

The selection operation is to select excellent chromosomes from the current population and prepare for crossover and Mutation. As the fitness of candidate individuals increases, the probability of being selected increases. In general, a method of roulette wheel selection based on proportional fitness assignment (also known as Monte Carlo method) is used however; one drawback of this method is that the selection based on the generated random number that may lead to some individuals with high fitness is eliminated. Therefore, we made an improvement: Firstly, we will select the individuals with the greatest fitness value to ensure that they can enter the next stage, and then select the remaining individuals according to the method of roulette. This improvement ensures that the best individuals will not be eliminated

3） Improved crossover

Crossover using partially matched crossover (PMC), the traditional method is to exchange randomly selected segments from two adjacent chromosomes. However, the two adjacent chromosomes, selected by roulette, are sometimes the same, So two chromosomes remain unchanged after the crossover operation, and thus this crossover operation has no effect. So We take the interval crossover.

4） Mutation

The mutation operation is to change a certain bit in the chromosome. It can use the random search ability of mutation operator. When the operation result is close to the optimal solution neighborhood, it can quickly converge to the optimal solution.

V. DTS FOR INTRUSION DETECTION

DTS module is mainly divided into two steps in the training phase.Each RBM is trained separately, characterized by unsupervised and independent; to ensure that feature information is retained as much as possible when mapping feature vectors into different feature spaces.

Once an RBM is trained, another RBM is "stacked" atop it, taking its input from the final trained layer. The new visible layer is initialized to a training vector, and values for the units in the already-trained layers are assigned using the current weights and biases. The new RBM is then trained with the procedure above. This whole process is repeated until the desired stopping criterion is met [34]. Finally, this process is repeated until to the last layer. This is a Deep Learning method. The last layer of the DTS is the BP neural network.

The feature vector of upper RBM is used as an input vector to train an entity classifier under supervision. Since the RBM of each layer can only ensure its own weight corresponding to the feature vector is optimal after the first step training, our ultimate goal is to make the overall weight corresponding to the feature vector as optimal. So according to the characteristics of the BP neural network, the BP neural network can propagate error information from the top layer to the bottom layer of RBM. If fine-tune the DTS network is finely tuned, a global optimization could be achieved. The number of hidden layers and the number of neurons in each layer in the deep Trust System are determined by the algorithm model we constructed earlier.

Algorithm

The algorithm flow is summarized as:

Step1: Initialize the population and generate different number of hidden layers and the number of neurons in each layer randomly;

Step2: Calculate the fitness value according to Equation below, chosen by the roulette method, and keep the optimal individual in the present; interval crossover; variation;

$$f = w_1 \times p + w_2 \times l + w_3 \times (1 - \sigma^*)$$

Step3: "Elite" retains, retaining individuals with the greatest fitness value during evolution;

Step4: Determine if the maximum number of iterations has been reached. If reached, the generated network structure is retained, otherwise iterate Step2- Step3 again;

Step5: Use the optimal network structure for the deep Trust System and train the intrusion detection model.

Step6: Classify the testing set by the trained DTS module, and finally match the classification result with the category information of the testing set to check the accuracy of the classification.

VI. EXPERIMENTAL SIMULATION

A. Experimental data

KDDCUP99 [35] and NSL-KDD are the most commonly used datasets in the intrusion detection research. We used NSL-KDD intrusion dataset which is available in csv format for model validation and evaluations. The dataset composes of the attacks shown in Table 1, and identified as a key attack in IoT computing. Sherasiya and Upadhyay (2016) point out that IoT objects are also exposed to such types of attacks. Furthermore, Sherasiya and Upadhyay (2016) point out that the data that IoT objects exchange are of the same value and importance, or occasionally more important than a non-IoT counterpart [36].

According to the analysis of KDDCUP99 and its latter version NSL-KDD, malicious behaviors (attacks) in network-based intrusions can be classified into the following four main categories:

Table 1:

Main class	Sub class (attacks) in train set	New sub class (attacks) in test set
DoS	Back, land, Neptune, Smurf, pod, teardrop	Apache2, Mailbomb, Processtable
Probe	Imap, multihop, phf, spy, warezclient, warezmaster, ftp write, guess passwd	Mscan, Saint
U2R	Buffer overflow, perl, loadmodule, rootkit	Httptunnel, Ps, Sqlattack, Xterm
R2L	Ipsweep, nmap, portsweep, satan	Sendmail, Named, Snmpgetattack, Snmp guess, Xlock, Xsnoop, Worm

Probe: when an attacker seeks to only gain information about the target network through network and host scanning activities.

DoS (Denial of Service): when an attacker interrupts legitimate users' access to the given service or machine.

U2R (User to Root): when an attacker attempts to escalate a limited user' privilege to a super user or root access (e.g. via malware infection or stolen credentials).

R2L (Remote to Local): when an attacker gains remote access to a victim machine imitating existing local users.

In order to make the classification result more accurate and meet the standard conditions of the DTS's input data set, the data set needs to be normalized. Normalization techniques are necessary for data reduction since it is quiet complex to process huge amount of network traffic data with all features to detect intruders in real time and to provide prevention methods. The method used in this paper is the Min-Max normalization method, also known as deviation standardization, which is a linear change to the original data, mapping the resulting value to [0, 1], the conversion function is as follows:

$$X^* = \frac{X - Min}{Max - Min}$$

where Max is the maximum value of the sample data, and Min is the minimum value of the sample data. Below is a summary of the metrics we adopted to evaluate the detection method:

	Predicted: normal	Predicted: attack
Actual：normal	TN	FP
Actual：attack	FN	TP

where, accuracy (ACC) is the percentage of true detection over total data instances; detection rate (DR) represents ratio of intrusion instances; false alarm rate (FAR) represents the ratio of misclassified normal instances;

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

$$DR = \frac{TP}{TP + FN}$$

$$FAR = \frac{FP}{TN + FP}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{(TP + FN)}$$

Precision represents how many of the returned attacks are correct; Recall represents how many of the attacks does the model return. FP: false positive, TP: true positive, TN: true negative, FN: false negative.

VII. SIMULATION ENVIRONMENT

The experiment was conducted using NS2 running on a personal computer (PC). GA optimized DTS model is trained with the training sets and then evaluated using the test set.

Simulation Results

First, we need to set the number of generations of the genetic algorithm.

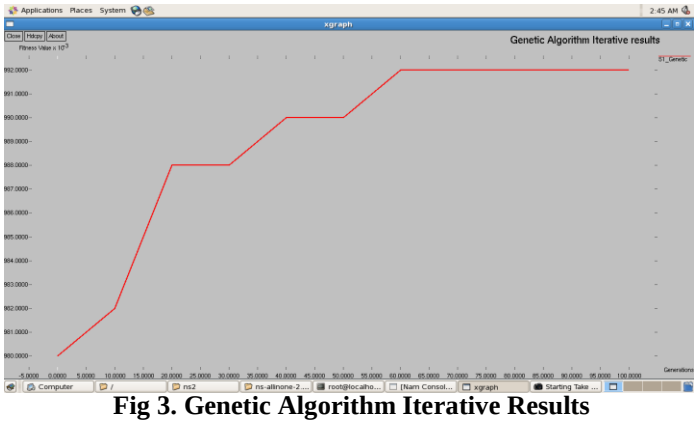

Fig 3. Genetic Algorithm Iterative Results

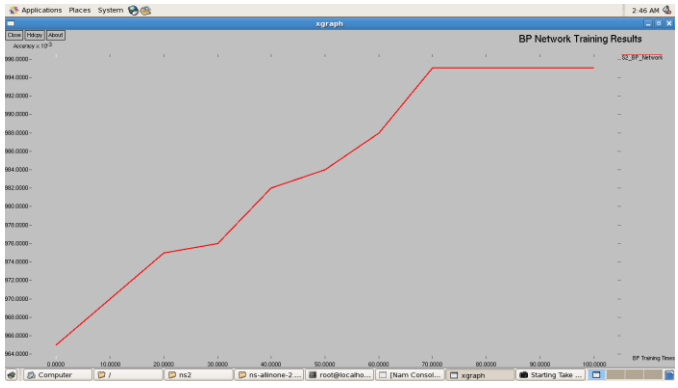

Fig. 4. BP Network Training Results

It can be seen from Fig. 3 that as the number of iterations increases, the fitness value increases, and when the number of iterations exceeds 50, the curve tends to be stable, and the fitness value no longer increases with the number of iterations. Therefore, we set the genetic algebra of the genetic algorithm to 50 generations.

From Fig. 4, we can see that when the number of training exceed 80 times, the curve is basically stable, and with the increase in the number of training, the classification accuracy rate no longer increases significantly and wasted training time in vain, so we set the BP network training epochs to 80.

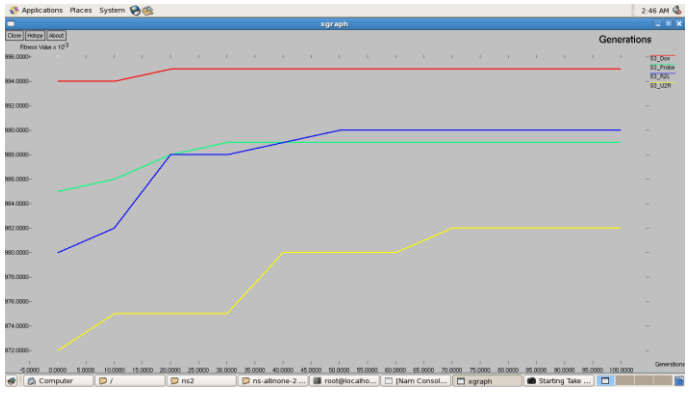

Fig. 5. Generations

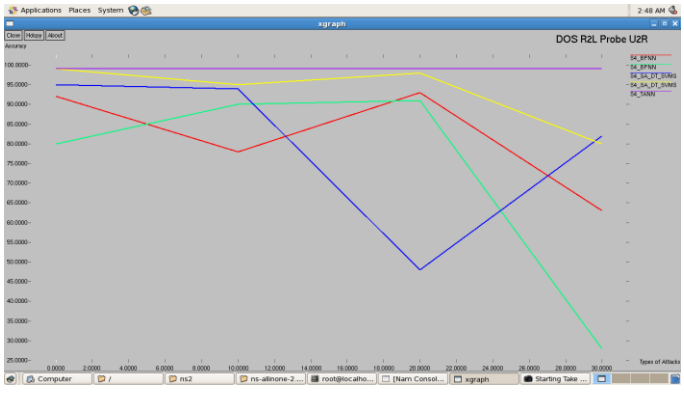

Fig. 6. Types of Attacks

The DoS detection rate of network structure ‘A’ generated by DoS as a training set is significantly higher than that of other structures; the R2L detection rate of network structure ‘B’’ generated by R2L as a training set also significantly higher than that of other structures. The classification accuracy of Probe and U2R is relatively high under all the four network structures, so the comparison results are not very significant. It can be seen that the network structure adaptively generated by the genetic algorithm has a higher detection rate than other network structures. It can be seen from Figs 5 and 6 that the proposed GA-DBN method has reached a very high level for the detection of four types of attacks. The classification accuracy of DoS is higher than 99%, and the classification accuracy of R2L, Probe and U2R is also significantly higher than other methods.

CONCLUSION

Through GA, the optimal individuals can be generated. DTS can effectively process high complex and high dimensional data, and the classification results are very good. So in this paper, the improved genetic algorithm combined with a deep Trust Systems, GA performs multiple iterations to produce an optimal network structure, DTS then uses the obtained network structure as an intrusion detection model to classify the attacks. In this way, facing different attacks, the problem of how to select an appropriate network structure when using deep learning methods for intrusion detection is solved, and thus it improves the classification accuracy and generalization of the model, and reduces the complexity of network structure. Algorithm is also significantly higher than other methods. In addition, as the model complexity is reduced, the training time of DTS can be reduced without affecting the accuracy of model classification.

In addition, the algorithm combining GA and DTS model not only can be used in intrusion detection in the IoT, also can be applied to other situations, such as classification and recognition.

For different training sets, an optimal network structure is adaptively generated for classification. Moreover, for small training sets, high classification accuracy can also be achieved, which helps to find low-frequency attacks in intrusion detection systems. In the future, we will consider to optimize the other parameters of the deep network, reduce the training time and improving the accuracy.

REFERENCES

[1] Q. Jing, A. V. Vasilakos, J. Wan, J. Lu, and D. Qiu, "Security of the Internet of Things: perspectives and challenges," Wireless Networks, vol. 20, pp. 2481–2501, 2014.

[2] A. Abduvaliyev, A.K Pathan, J. Zhou, R. Roman and W. Wong, "On the Vital Areas of Intrusion Detection Systems in Wireless Sensor Networks", Communications Surveys & Tutorials, IEEE vol. 15, pp. 1223-1237, 2013.

[3] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security,privacy and trust in internet of things: The road ahead," Computer Networks, vol. 76, pp. 146–164, 2015.

[4] M. Farooq , M. Waseem , A. Khairi , and S. Mazhar , "A Critical Analysis on the Security Concerns of Internet of Things (IoT)," Perception, vol. 111, pp. 1-6, 2015.

[5] Tuhin Borgohain, Uday Kumar, Sugata Sanyal, "Survey of Operating Systems for the IoT Environment", arXiv preprint arXiv:1504.02517, vol. 6, pp. 2479-2483, 2015

[6] H HaddadPajouh, A Dehghantanha, R Khayami, KK Choo, "A deep Recurrent Neural Network based approach for Internet of Things malware threat hunting", Future Generation Computer Systems, vol. 85, pp. 88–96, 2018.

[7] M Conti, A Dehghantanha, K Franke, S Watson, "Internet of Things Security and Forensics: Challenges and Opportunities", Elsevier Future

[8] H. H. Pajouh, R. Javidan, R. Khayami, D. Ali, and K. K. R. Choo, "A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in iot backbone networks," IEEE Transactions on Emerging Topics in Computing, vol. PP, no. 99, pp. 1–1, 2016.

[9] W. Lee and S. J. Stolfo. Data mining approaches for intrusion detection. In Proceedings of the 7th USENIX Security Symposium, San Antonio, TX, pp. 120–132, January 1998.

[10] L. Khan, M. Awad, and B. M. Thuraisingham. A new intrusion detection system using support vector machines and hierarchical clustering. VLDB J., 16(4):507–521, 2007.

[11] E. Hodo, X. Bellekens, A. Hamilton, P. L. Dubouilh, E. Iorkyase, C. Tachtatzis, and R. Atkinson, "Threat analysis of iot networks using artificial neural network intrusion detection system," in 2016 International Symposium on Networks, Computers and Communications (ISNCC), May 2016, pp. 1–6.

[12] A. Diro and N. Chilamkurti, ''Distributed attack detection scheme using deep learning approach for Internet of Things,'' Future Generat. Comput. Syst., vol. 282, pp. 761–768, May 2017.

[13] R. Beghdad, "Critical study of neural networks in detecting intrusions," Computers and Security, vol. 27, pp. 168-175, 2008.

[14] S. Mukkamala, G. Janoski, and A. Sung. "Intrusion detection using neural networks and support vector machines," Proceedings of the International Joint Conference on Neural Networks (IJCNN'02), Honolulu, HI, USA, pp. 1702–1707, 2002.

[15] E. Hodo, X. J. A. Bellekens, A. Hamilton, C. Tachtatzis, and R. C. Atkinson, "Shallow and deep networks intrusion detection system: A taxonomy and survey," CoRR, vol. abs/1701.02145, 2017.

[16] J. Ryan, M. Lin, and R. Miikkulainen, "Intrusion detection with neural networks," in Proc. Advances NIPS 10, Denver, CO, pp. 943–949, 1997.

[17] E. Yu, and P. Parekh, "A Bayesian Ensemble for Unsupervised Anomaly Detection," in arXiv preprint arXiv:1610.07677, pp. 1–5, 2016.

[18] A Saraswati, M Hagenbuchner, Zhi Quan Zhou, "High Resolution SOM Approach to Improving Anomaly Detection in Intrusion Detection

[19] G. Wang, J. Hao, J. Ma, L. Huang, "A new approach to intrusion detection using artificial neural networks and fuzzy clustering", Exp. Syst. Appl. pp. 6225–6232, 2010.

[20] T Ma, F Wang, J Cheng, Y Yu, and X Chen, "A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks," in Sensors vol. 16, no. 10, 2016.

[21] Z Chiba, N Abghour, K Moussaid, and M Rida, "A cooperative and hybrid network intrusion detection framework in cloud computing based on snort and optimized back propagation neural network," in Procedia Computer Science vol. 83 pp. 1200-1206, 2016.

[22] S. Karsoliya. "Approximating number of hidden layer neurons in multiple hidden layer BPNN architecture," International Journal of Engineering Trends and Technology, vol.3, pp. 713–717, 2012.

[23] C Yin, Y Zhu, J Fei, and X He, "A deep learning approach for intrusion detection using recurrent neural networks," in IEEE Access vol. 5 pp. 21954-21961, 2017.

[24] B. Abolhasanzadeh, "Nonlinear dimensionality reduction for intrusion detection using auto-encoder bottleneck features," in 2015 7th Conference on Information and Knowledge Technology (IKT), Urmia, Iran, pp. 1–5, 2015.

[25] N. Gao, L. Gao, Q. Gao, and H. Wang, "An Intrusion Detection Model Based on Deep Belief Networks," in 2014 Second International Conference on Advanced Cloud and Big Data, Huangshan, China, pp. 247–252, 2014.

[26] M. Z. Alom, V. Bontupalli, and T. M. Taha, "Intrusion detection using deep belief networks," in 2015 National Aerospace and Electronics

[27] Q. Tan, W. Huang, and Q. Li, "An intrusion detection method based on DBN in ad hoc networks," in International Conference on Wireless Communication and Sensor Network, Wuhan, China, pp. 477–485, 2016.

[28] Y. Liu, J. A. Starzyk, and Z. Zhu, "Optimizing number of hidden neurons in neural networks," in Proceedings of the IASTED International Conference on Artificial Intelligence and Applications (AIA '07), Innsbruck, Austria, pp. 121–126, February 2007.

[29] I. Rivals I. and L. Personnaz, "A statistical procedure for determining the optimal number of hidden neurons of a neural model," in Proceedings of the Second International Symposium on Neural Computation (NC'2000), Berlin, May, 23-26, 2000.

[30] K. Z. Mao and G. B. Huang, "Neuron selection for RBF neural network classifier based on data structure preserving criterion," IEEE Transactions on Neural Networks, vol. 16, no. 6, pp. 1531–1540, 2005.

[31] C. A. Doukim, J. A. Dargham, and A. Chekima, ``Finding the number of hidden neurons for an MLP neural network using coarse to ne search technique," in Proc. 10th Int. Conf. Inf. Sci., Signal Process. Appl. (ISSPA), May 2010, pp. 606609.

[32] K. G. Sheela and S. N. Deepa, ``Reviewon methods to x number of hidden neurons in neural networks," Math. Problems Eng., vol. 2013, May 2013, Art. no. 425740.

[33] G. E. Hinton, S. Osindero, and Y.-W. Teh, ``A fast learning algorithm for deep belief nets," Neural Comput., vol. 18, no. 7, pp. 15271554, 2006.

[34] Y. Bengio, ``Learning deep architectures for AI," Found. Trends Mach. Learn., vol. 2, no. 1, pp. 1127, 2009.

[35] M. Tavallaee, E. Bagheri,W. Lu, and A. Ghorbani, ``A detailed analysis of the KDD CUP 99 data set," Proc. 2nd IEEE Symp. Comput. Intell. Secur. Defense Appl. (CISDA), Ottawa, ON, Canada, Jul. 2009, pp. 16.

[36] A, Alghuried, ``Amodel for anomalies detection in Internet of Things (IoT) using inverse weight clustering and decision tree," M.S. thesis, School Comput., Dublin Inst. Technol., Dublin, Republic of Ireland, 2017.