A Hybrid Multi-Layer Intrusion Detection System in Cloud

Manickam M. and Rajagopalan S.P.*
1. GKM College of Engineering and Technology, Chennai-63, Tamil Nadu, INDIA
2. Department of Computer Science and Engineering, GKM College of Engineering and Technology, Chennai-63, Tamil Nadu, INDIA
*sasirekaraj@yahoo.co.in

Abstract
The cloud computing is a representation of a technology for the usage of computing infrastructure efficiently and also is a business model for the sale of computing services and resources. This offers a great potential for the improving of the productivity as well as to bring down the costs and simultaneously ensure it handles all risks. The Intrusion Detection Systems (IDS) is widely used for the detection of malicious behavior in the communication of network and its host. The current IDS system has a set of rules that have various patterns of attach that are stored inside databases and the entire network traffic is hereby matched against it in order to avoid any illegal or unauthorized activities.

Here in this work the structure optimized multi-layer Artificial Neural Network (ANN) based IDS in the cloud is presented. The hybrid Glow Swarm Optimization (GSO)-Tabu Search (TS) called GSO-TS is used for the structure optimization and to reduce its convergence time, to solve the similar old problem, premature convergence or trapping at local optima. Results show that the better performance.

Keywords: Intrusion Detection Systems (IDS), Cloud Computing, Artificial Neural Network (ANN), Glow Swarm Optimization (GSO) and Tabu Search (TS).

Introduction

Cloud computing enables ubiquitous, on-demand Internet access to computing resources that can be provisioned with minimal interaction with service providers. The cloud delivers the demand of users for near consistent access to their information, resources and data. Many businesses have already implemented cloud computing given its characteristics such as on-demand self-service, broad network access, resource pooling, rapid elasticity and measured services. These characteristics allow users to focus on their own businesses processes, while the computing resources are managed by a Cloud Service Provider (CSP). The cloud model reduces business costs by simplifying the process of installing hardware and software updates and ensuring availability and adaptability of computing resources [1].

The deployment models, depending on how resources are organized and how they are available to users can be classified as: Public cloud, where access is available to the general public and may be owned and managed by a private, academic, government, or any combination.

Private cloud infrastructure is provisioned for exclusive use by a single organization. Community Cloud, infrastructure is provisioned for exclusive use of a specific user community (organizations with common interests). Hybrid cloud, where the cloud infrastructure is made up of any combination of different infrastructures (private, public or community), becoming a single cloud [2].

Cloud computing also has three service models namely Platform as a Service (PaaS), Infrastructure as a Service (IaaS) and Software as a Service (SaaS) models. PaaS model facilitates users by providing platform on which applications can be developed and run. IaaS deliver services to users by maintaining large infrastructures like hosting servers, managing networks and other resources for clients. SaaS model makes user worry free of installing and running software services on its own machines. Presently, Salesforce.com, Google and Amazon are the leading cloud service providers who extend their services for storage, application and computation on pay as per use basis. Data, application and services non-availability can be imposed through Denial of Service (DOS) or Distributed Denial of Service (DDOS) attacks and both cloud service provider and users become handicap to provide or receive cloud services[3].

Intrusion Detection Systems (IDS) are an essential component of defensive measures protecting computer systems and network against harm abuse. It becomes crucial part in the cloud computing environment. The main aim of IDS is to detect computer attacks and provide the proper response. An IDS is defined as the technique that is used to detect and respond to intrusion activities from malicious host or network. Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions. Intrusions are defined as attempts to compromise the confidentiality, integrity or availability of computer or network or to bypass its security mechanisms [4].

There are two main categories of intrusion detection technique: Misuse Detection model refers to detection of intrusions that follow well-defined intrusion patterns. It is very useful in detecting known attack pattern. Anomaly detection refers to detection performed by detecting changes in the patterns of utilization or behavior of the system. It can be used to detect known and unknown attack. Anomaly Detection identifies abnormal behaviour (anomalies). There are mainly two categories of IDSs, network based and host based. In addition, the IDS can be defined as a defense system, which detects hostile activities in a network.

Cloud computing have two approaches i.e. knowledge-based IDS and behavior-based IDS to detect intrusions in cloud computing. Behavior-based intrusion detection techniques assume that an intrusion can be detected by observing a deviation from normal or expected behavior of the system or the users. The model of normal or valid behavior is extracted from reference information collected by various means. The IDS later compares this model with the current activity. When a deviation is observed, an alarm is generated. In other words, anything that does not correspond to a previously learned behavior is considered intrusive. Therefore, the IDS might be complete (i.e. all attacks should be caught), but its accuracy is a difficult issue (i.e. you get a lot of false alarms) [5].

Knowledge-based intrusion detection techniques apply the knowledge accumulated about specific attacks and system vulnerabilities. The IDS contains information about these vulnerabilities and looks for attempts to exploit these vulnerabilities. When such an attempt is detected, an alarm is triggered. In other words, any action that is not explicitly recognized as an attack is considered acceptable.

There are four types of cloud computing based IDS. Host-based IDS (HIDS) was the first type of intrusion detection software to be designed, with the original target system being the mainframe computer where outside interaction was infrequent. HIDS operate on information collected from within an individual computer system. A HIDS monitors the inbound and outbound packets from the computer system only and would alert the user or administrator if suspicious activity is detected. HIDSs analyze the suspicious activities like system call, processes or thread, asset and configuration access by observing the situation of host. It is especially used to protect valuable and private information on server systems [6].

Networks based IDSs (NIDS) capture the traffic of entire network and analyze it to detect possible intrusions like port scanning, DoS attacks etc. NIDS usually performs intrusion detection by processing the IP and transport layer headers of captured network packets. It utilizes the anomaly based and signature based detection methods to identify intrusions. NIDS collects the network packets and looks for their correlation with signatures of known attacks or compares the users' current behavior with their already known profiles in real-time. Multiple hosts in the network can be secured from attackers by utilizing a few properly deployed NIDSs. If run in stealth mode, the location of NIDS can be hidden from attacker. The NIDS is unable to perform analysis if traffic is encrypted [7].

Hypervisor based IDS, hypervisor provides a level for interaction among VMs. Hypervisor based IDSs is placed at the hypervisor layer. It helps in analyze the available information for detection of anomalous actions of users. The information is based on communication at multiple levels like communication between VMs, VM and hypervisor, and communication within the hypervisor based virtual network [8].

Distributed IDS (DIDS) contains number of IDSs such as NIDS, HIDS which are deployed over the network to analyze the traffic for intrusive detection behavior. Each of these individual IDSs has its two components: detection component and correlation manager. Detection component examine the system's behavior and transmits the collected datain a standard format to the correlation manager. Correlation manager combines data from multiple IDS and generate high level alerts that keep up a correspondence to an attack. Analysis phase makes use of signature based and anomaly based detection techniques so DIDS can detect known as well as unknown attacks.

The ANN as a pattern recognition technique. ANN an information processing model that is inspired by the biological nervous systems, such as brain, process information. ANN is the network of individual neurons. Each neuron is a Neural Network (NN) acts as an independent processing element. Each processing element (neuron) is fundamentally a summing element followed by an activation function. The output of each neuron (after applying the weight parameter associated with the connection) is fed as the input to all of the neurons in the next layer. Like human or other brain, NNs also learn by example or training, they cannot define or program to perform a specific task [9].

For solving constrained optimization design problem, most of traditional algorithms are based on the concept of gradient, they request that the objective function and constraint conditions should be differentiable, and the obtained solution is mostly local optimal solution. Penalty function methods are simple, convenient and don't strictly require problem itself, but how to determine the suitable penalty factors is more difficult. In addition, in view of the deficiencies of the low accuracy and the poor stability for solving constrained optimization problems, this work a hybrid GSO-TS algorithm is proposed. In the evolutionary process, GSO-TS uses the constraint processing technology based on feasibility rules to update the optimal location of the population, which makes the population rapidly convergence to feasible regions and find better feasible solution. Besides, to avoid premature, GSO-TS adopts the local search strategy based on TS to optimize the local optimal value [10].

In this work, propose an ANN, GSO and hybrid GSO-TS in IDS based cloud computing. Section 2 reviews related work in literature. Section 3 describes methods used and Section 4 discusses experiments results. Section 5 concludes the work.

Related Works

Ghosh et al., [11] proposed an efficient, fast and secure IDS with the collaboration of multi-threaded NIDS and HIDS. In the existing system, Cloud-IDS capture packets from Network, analyze them and send reports to the cloud administrator on the basis of analysis. Analysis of packets is done using K-Nearest Neighbor and Neural Network (KNN-NN) hybrid classifier. For training and testing purpose here the author have used NSL-KDD dataset. After getting the report from the cloud-IDS, Cloud Service Provider (CSP) will generate an alert for the user as well as maintain a loglist for storing the malicious IP addresses.

The proposed model handles large flow of data packets, analyze them and generate reports efficiently integrating anomaly and misuse detection.

Pandeeswari & Kumar [12] proposed an anomaly detection system at the hypervisor layer named hypervisor detector that uses a hybrid algorithm which is a mixture of Fuzzy C-Means (FCM) clustering algorithm and ANN (FCM-ANN) to improve the accuracy of the detection system. The proposed system is implemented and compared with naïve bayes classifier and classic ANN algorithm. The DARPA's KDD cup dataset 1999 is used for experiments. Based on extensive theoretical and performance analysis, it is evident that the proposed system is able to detect the anomalies with high detection accuracy and low false alarm rate even for low frequent attacks thereby outperforming naïve bayes classifier and classic ANN.

Baig et al., [13] presented a Cascade of ensemble-based ANN for multi-class Intrusion Detection (CANID) in computer network traffic. The proposed system learns a number of neural-networks connected as a cascade with each network trained using a small sample of training examples. The proposed cascade structure uses the trained NN as a filter to partition the training data and hence a relatively small sample of training examples are used along with a boosting-based learning algorithm to learn an optimal set of NN parameters for each successive partition. Experimental results show that the proposed approach can efficiently detect various types of cyber-attacks in computer networks.

Rajendran [14] proposed a HIDS algorithm for private cloud environment which is efficient in terms of security and performance. The existing Intrusion detection systems, which gives a clear view that existing system cannot detect intrusion effectively. Artificial Intelligence (AI) has been incorporated in the research work to detect any type of intrusion in private cloud environment. Incorporating AI technique resulted in self-adaptive IDS, which has been tested using the real time data collected using the network speed. The proposed algorithm can be implemented for highly secured private cloud which are built for Military purpose and Banking sector to monitor the activities of the network efficiently.

Various meta-heuristic algorithms are applied to deal with the problem of scheduling, which is an NP-hard problem. Masdari et al., [15] presented an in-depth analysis of the Particle Swarm Optimization (PSO)-based task and workflow scheduling schemes proposed for the cloud environment in the literature. Moreover, it provides a classification of the proposed scheduling schemes based on the type of the PSO algorithms which have been applied in these schemes and illuminates their objectives, properties and limitations. Finally, the critical future research directions are outlined.

Cao et al., [16] proposed a parallel design and realization method for a PSO-optimized Back-Propagation (BP) NN based on MapReduce on the Hadoop platform using both the PSO algorithm and a parallel design. The PSO algorithm was used to optimize the BP NN's initial weights and thresholds and improve the accuracy of the classification algorithm. The MapReduce parallel programming model was utilized to achieve parallel processing of the BP algorithm, thereby solving the problems of hardware and communication overhead when the BP NN addresses big data. The algorithm proposed in this work demonstrated both higher classification accuracy and improved time efficiency, which represents a significant improvement obtained from applying parallel processing to an intelligent algorithm on big data.

Hyper-heuristic algorithm finds better scheduling solutions for cloud computing systems and to further improve the scheduling results in terms of make span. Kumari et al., [17] proposed a novel Multi-Objective PSO (MOPSO) and Genetic Algorithm (GA) based hyper-heuristic resource scheduling algorithm as the hybrid algorithm. Performance of the proposed algorithm has also been evaluated through the Cloud Sim toolkit. The author have compared the hybrid scheduling algorithm with existing common heuristic-based scheduling algorithms. The results thus obtained have shown a better performance by the algorithm than the existing algorithms, in terms of giving reduce cost and improve makespan. The proposed model shows the improved resource utilization, makespan, and throughput.

Methodology

In this section, the structure optimized NN, GSO, GSO-ANN and hybrid GSO-TS methods are discussed.

Structure Optimized Neural Network Using IDS: ANN based IDS is an efficient solution for structured network data. The intrusion detection accuracy of this approach is based on number of hidden layers and training phase of ANN. However, it requires more training samples and time for effective learning of ANN. Use of only ANN based IDS cannot be an efficient solution to detect intrusions for cloud as it requires quick intrusion detection mechanism. It uses ANN based anomaly detection technique for cloud environment, which requires more training samples as well as more time for detecting intrusions effectively [18].

During training, the NN parameters are optimized to associate outputs (each output represents a class of computer network connections, like normal and attack) with corresponding input patterns (every input pattern is represented by a feature vector extracted from the characteristics of the network connection record). When the NN is used, it identifies the input pattern and tries to output the corresponding class. The most commonly reported application of NNs in IDSs is to train the neural net on a sequence of information units, each of which may be an audit record or a sequence of commands [19].

A Multi-Layer Perceptron (MLP) is used for intrusion detection. MLP is a layered feed forward ANN networks typically trained with BP. These networks have found their way into countless applications requiring static pattern classification. Their main advantage is that they are easy to use, and that they can approximate any input/output map. The proposed system detects the attacks and classifies them in six groups with the hidden layers of neurons in the NN.

The first advantage in the utilization of a NN in the detection of the network intrusion would be the flexibility that the network would provide. ANNs are a uniquely power tool in multiple class classification, especially when used in applications where formal analysis would be very difficult or even impossible, such as pattern recognition and nonlinear system identification. NNs are able to work imprecise and incomplete data. It means that they can recognize also patterns not presented during a leaning phase. In that case, traditional IDS, based on the signatures of attacks or expert rules, may not be able to detect the new version of this attack [20].

The inherent speed of NNs is another benefit of this approach. Because the protection of computing resources requires the timely identification of attacks, the processing speed of the NN could enable intrusion responses to be conducted before irreparable damage occurs to the system.

The most important advantage of NNs in misuse detection is the ability of the NN to "learn" the characteristics of misuse attacks and identify instances that are unlike any which have been observed before by the network. A NN might be trained to recognize known suspicious events with a high degree of accuracy. While this would be a very valuable ability, since attackers often emulate the "successes" of others, the network would also gain the ability to apply this knowledge to identify instances of attacks which did not match the exact characteristics of previous intrusions.

Glow-worm Swarm Optimization (GSO) Algorithm: In GSO, each glow-worm distributes in the objective function definition space. These glow-worms carry own luciferin respectively, and has the respective field of vision scope called local-decision range. Their brightness concerns with in the position of objective function value. The brighter the glow, the better is the position, namely has the good target value. The glow seeks for the neighbor set in the local-decision range, in the set, a brighter glow has a higher attraction to attract this glow toward this traverse, and the flight direction each time different will change along with the choice neighbor. Moreover, the local-decision range size will be influenced by the neighbor quantity, when the neighbor density will be low, glow's policy-making radius will enlarge favors seeks for more neighbors, otherwise, the policy-making radius reduces. Finally, the majority of glow-worm return gathers at the multiple optima of the given objective function [21].

Each glow-worm i encodes the object function value $J(x_i(t))$ at its current location $x_i(t)$ into a luciferin value l_i and broadcasts the same within its neighbourhood. The set of neighbours $N_i(t)$ of glow-worm i consists of those glow-worms that have a relatively higher luciferin value and that are located within a dynamic decision domain, and updating by formula (1) at each iteration.

Local-decision range update in (1):

$$r_d^i(t+1) = \min\{r_s, \max\{0, r_d^i(t) + \beta(n_t - |N_i(t)|)\}\} \tag{1}$$

And $r_d^i(t+1)$ is the glow-worm i's local-decision range at the t +1iteration, r_s is the sensor range, n_t is the neighbourhood threshold, the parameter β affects the rate of change of the neighbourhood range.

The number of glow in local-decision range in (2):

$$N_i(t) = \{j : \| x_j(t) - x_i(t) \| < r_d^i ; l_i(t) < l_j(t)\} \tag{2}$$

and, $x_j(t)$ is the glow-worm i's position at the t iteration, $l_j(t)$ is the glow-worm i 's luciferin at the t iteration.; the set of neighbours of glow-worm i consists of those glow-worms that have a relatively higher luciferin value and that are located within a dynamic decision domain whose range I d r is bounded above by a circular sensor range $r_s (0 < r_d^i < r_s)$. Each glow-worm i selects a neighbour j with a probability p_{ij} (t) and moves toward it. These movements that are based only on local information, enable the glow-worms to partition into disjoint subgroups, exhibit a simultaneous taxis-behaviour toward and eventually co-locate at the multiple optima of the given objective function [22].

Probability distribution used to select a neighbour in (3):

$$p_{ij}(t) = \frac{l_j(t) - l_i(t)}{\sum_{k \in N_i(t)} l_k(t) - l_i(t)} \tag{3}$$

Movement update in (4):

$$x_i(t+1) = x_i(t) + s\left(\frac{x_j(t) - x_i(t)}{\| x_j(t) - x_i(t) \|}\right) \quad (4)$$

Luciferin-update in (5):

$$l_i(t) = (1-\rho)l_i(t-1) + \gamma J(x_i(t)) \quad (5)$$

and $l_i(t)$ is a luciferin value of glow-worm i at the t iteration, ρ ∈(0,1) leads to the reflection of the cumulative goodness of the path followed by the glow-worms in their current luciferin values, the parameter γ only scales the function fitness values, $J(x_i(t))$ is the value of test function [23].

Each glow-worm i selects a neighbour j with a probability $p_{ij}(t)$ and moves toward it. These movements that are based only on local information, enable the glow-worms to partition into disjoint subgroups, exhibit a simultaneous taxis-behaviour toward and eventually co-locate at the multiple optima of the given objective function.

Glow Swam Optimization Artificial Neural Network (GSO-ANN): The GSO-ANN divides the data of training into many subsets by using GSO. Later it trains the ANN by using various subsets and then determines the grades of the membership of the subsets and combines them through a new ANN to get the final results. This being a typical framework of machine learning it takes the GSO-ANN and incorporates the testing and the training phase. The training phase includes three stages:

Stage I: In an arbitrary Data Set (DS), Training Sets (TR) and Testing Sets (TS) are divided. After this different subset of training $TR_1,\ TR_2,\ ..\ TR_k$ are made from the TR with the module of Glow Swarm Optimization.

Stage II: In each subset of training $TR_i\left(i=1,\ 2,\ ...k\right)$, the model of ANN, $ANN_i\left(i=1,\ 2,\ ...k\right)$ which is the training by means of specific learning algorithm for formulating k which are different base models of ANN.

Stage III: To reduce errors for each ANN_i that simulated the ANN_i by using the entire set of training TR and gets results. Later the membership grades are used that are generated by the module of Glow Swarm optimization to combine these results. Later it train a new ANN model by using the results that are combined.

The stages in the GSO-ANN framework bring up three important issues:
(1) How a k different training subset can be created from TR the original training dataset;
(2) How a different base model ANN_i that has different training subsets can be created;
(3) How to aggregate different results that are produced by different base models ANN_i

Hybrid GSO-TS Algorithm: In 1986, Glover and Laguna first developed a renowned meta-heuristic algorithm called TS. TS is an iterative procedure designed for exploring in the solution space to find the near optimal solution. TS starts with a random solution or a solution obtained by a constructive and deterministic method and evaluates the fitness function. Then all possible neighbors of the given solution are generated and evaluated. A neighbor is a solution which can be reached from the current solution by a simple move. New solution is generated from the neighbors of the current one. To avoid retracing the used steps, the method records recent moves in a tabu list. The tabu list keeps track of previously explored solutions and forbids the search from returning to a previously visited solution. If the best of these neighbors is not in the tabu list, pick it to be the new current solution. One of the most important features of TS is that a new solution may be accepted even if the best neighbour solution is worse than the current one. In this way it is possible to overcome trapping in local minima. TS algorithm has been successfully used to lots of optimization problems [24].

However, in the TS algorithm, if a neighboring solution is not in the tabu list, TS sets it as the new current solution, but this solution is commonly worse than the current best solution. TS typically finds local minima and so do not change the best solution for many iterations; therefore, reaching a near-global minimum takes a long time and its convergence speed is low. To overcome this shortcoming of the TS algorithm; to reduce its convergence time, to solve the similar old problem, premature convergence or trapping at local optima.

In hybrid GSO-TS algorithm, during initialization, GSO a solution space of ‘population size’ is generated randomly within the limits of cutting speed, feed rate and depth of cut and TS sets it as the new current solution. Iteration consists of luciferin update phase, glow-worms’ movement phase and local decision range update phase and TS take each iteration is 30. The luciferin update phase was influenced by the function value at the glow-worm location. The value of the function will be altered due to the function value at the present position although the glow-worm has the same value of luciferin at the beginning. During the movement phase, each glow-worm uses a probabilistic mechanism to move toward their neighbor that hashigher intensity of luciferin then its own. Neighborhood range update phase is used to detect the multiple peaks in a multimodal function landscape [25].

The hybrid GSO-TS concept can be used to construct a more powerful neighborhood structure by hybrid optimization concepts. Implementation at the GSO at the individual agent level gives rise to two major phases at the group level: Formation of dynamic networks that results in splitting of the swarm into sub-swarms and local convergence of glow-worms in each subgroup to the peak locations. Figure 1 shows the flowchart for hybrid GSO-TS.

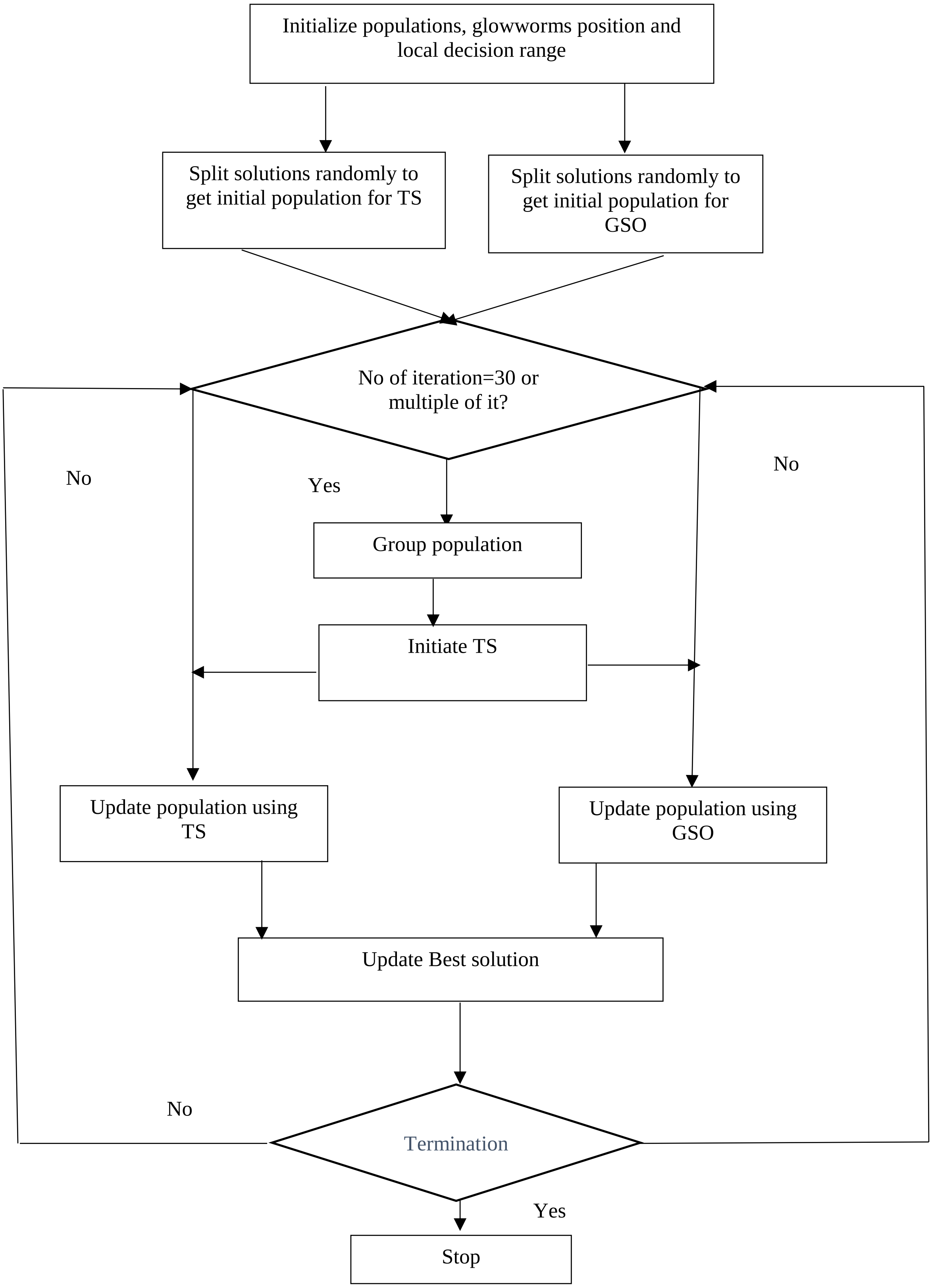
Initialize populations, glowworms position and local decision range
Split solutions randomly to get initial population for TS
Split solutions randomly to get initial population for GSO
No of iteration=30 or multiple of it?
No
Yes
No
Group population
Initiate TS
Update population using TS
Update population using GSO
Update Best solution
No
Termination
Yes
Stop

Figure 1: Flowchart for Hybrid GSO-TS

Results and Discussion

In this section, the ANN with three hidden layers and GSO-Tabu-ANN with three hidden layers are used. Figure 1 & 2 shows the detection rate using ANN and GSO-Tabu ANN.

From the figure 2, it can be observed that the ANN - one hidden layer has lower average detection rate by 0.39% for ANN - two hidden layer and by 0.73% for ANN - three hidden layer.

From the figure 3, it can be observed that the GSO-Tabu-ANN - one hidden layer has lower average detection rate by 0.65% for GSO-Tabu- ANN - two hidden layer and by 0.22% for GSO-Tabu-ANN - three hidden layer.

Conclusion

The cloud computing aims at providing a convenient, and on-demand network access for a shared pool of computing resources that are rapidly provisioned and also released with a minimal effort of management or the service provider and its interactions. The ANN algorithms offer an advantage of dealing with the requirements of the computation. For the structure optimization of the GSO-TS a proposal is made of optimization of structure.

GSO-TS uses the constraint processing technology based on feasibility rules to update the optimal location of the population, which makes the population rapidly convergence to feasible regions and find better feasible solution. The results show that theANN - one hidden layer has lower average detection rate by 0.39% for ANN - two hidden layer and by 0.73% for ANN - three hidden layer. The GSO-Tabu-ANN - one hidden layer has lower average detection rate by 0.65% for GSO-Tabu- ANN - two hidden layer and by 0.22% for GSO-Tabu-ANN - three hidden layer.

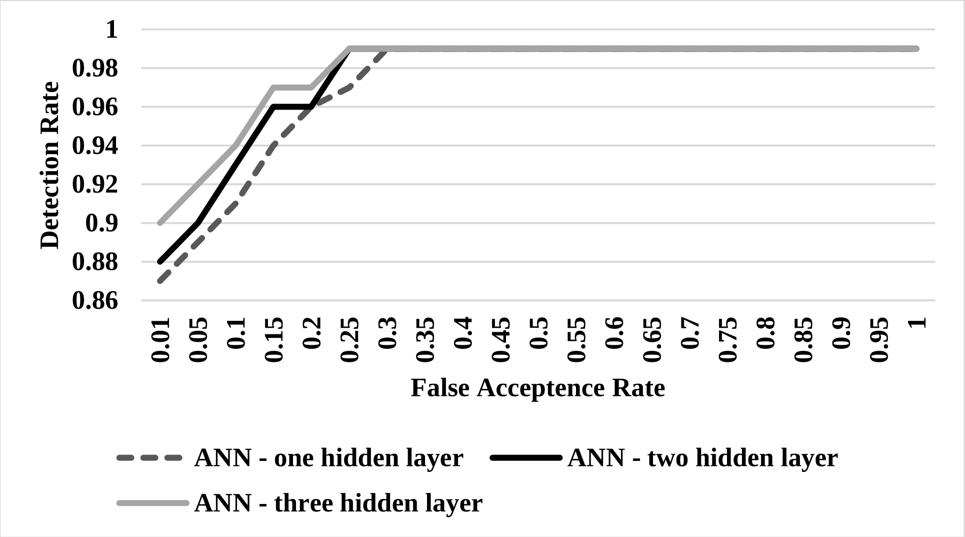

Figure 2: Detection Rate Using ANN

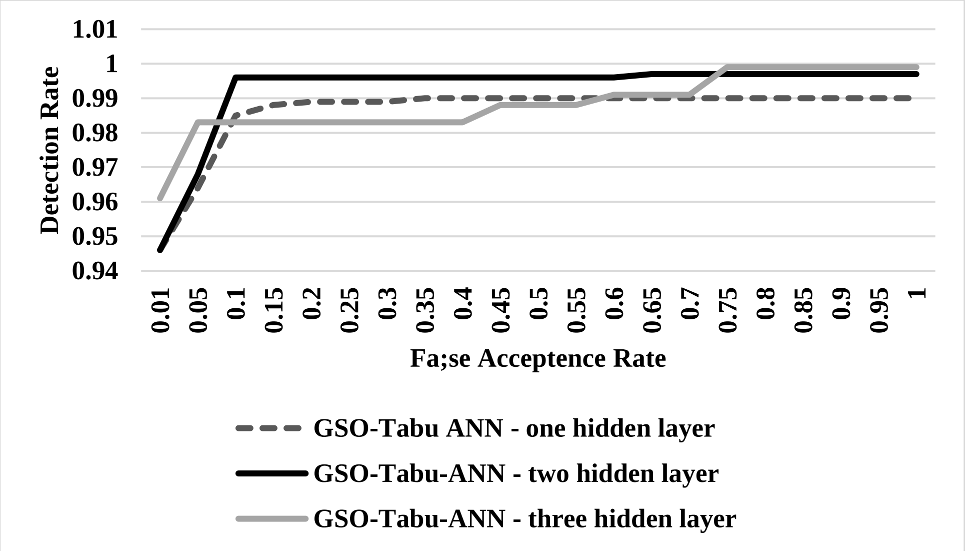

Figure 3: Detection Rate Using GSO-Tabu ANN

References

1. Carlin, A., Hammoudeh, M., & Aldabbas, O. **(2015)**. Defence for Distributed Denial of Service Attacks in Cloud Computing. Procedia Computer Science, **73**, 490-497.

2. Goyal, S. **(2014)**. Public vs private vs hybrid vs community-cloud computing: A critical review. International Journal of Computer Network and Information Security, 6(3), **20**.

3. Shelke, M. P. K., Sontakke, M. S., & Gawande, A. D. **(2012)**. Intrusion detection system for cloud computing. International Journal of Scientific & Technology Research, 1(4), **67-71**.

4. Mohod, A. G., &Alaspurkar, S. J. Analysis of IDS for Cloud Computing. International Journal of Application or Innovation in Engineering & Management (IJAIEM) Vol, 2, **344-349**.

5. Narwane, S. V., &Vaikol, S. L. **(2012)**. Intrusion Detection System in Cloud Computing Environment. In InInternational Conference on Advances in Communication and Computing Technologies (ICACACT).

6. Kumbhare, M. A., & Chaudhari, M. M. **(2014)**. IDS: Survey on Intrusion Detection System in Cloud Computing, International Journal of Computer Science and Mobile Computing, 3 (4), **497-502**.

7. Mehmood, Y., Habiba, U., Shibli, M. A., & Masood, R. (**2013**, December). Intrusion detection system in cloud computing: Challenges and opportunities. In Information Assurance (NCIA), 2013 2nd National Conference on (pp. **59-66**). IEEE.

8. Kene, S. G., & Theng, D. P. (**2015**, February). A review on intrusion detection techniques for cloud computing and security challenges. In Electronics and Communication Systems (ICECS), 2015 2nd International Conference on (pp. **227-232**). IEEE.

9. Subba, B., Biswas, S., &Karmakar, S. (**2016**, March). A Neural Network based system for Intrusion Detection and attack classification. In Communication (NCC), 2016 Twenty Second National Conference on (pp. **1-6**). IEEE.

10. Zhou, Y., Zhou, G., & Zhang, J. **(2013)**. A hybrid glowworm swarm optimization algorithm for constrained engineering design problems. Appl. Math. Inf. Sci, 7(1), **379-388**.

11. Ghosh, P., Mandal, A. K., & Kumar, R. **(2015)**. An Efficient Cloud Network Intrusion Detection System. In Information Systems Design and Intelligent Applications (pp. **91-99**). Springer India.

12. Pandeeswari, N., & Kumar, G. **(2016)**. Anomaly detection system in cloud environment using fuzzy clustering based ANN. Mobile Networks and Applications, 21(3), **494-505**.

13. Baig, M. M., Awais, M. M., & El-Alfy, E. S. M. **(2017)**. A multiclass cascade of artificial neural network for network intrusion detection. Journal of Intelligent & Fuzzy Systems, 32(4), **2875-2883**.

14. Rajendran, P. K. **(2015)**. Hybrid intrusion detection algorithm for private cloud. Indian Journal of Science and Technology, 8(35).

15. Masdari, M., Salehi, F., Jalali, M., &Bidaki, M. **(2016)**. A Survey of PSO-Based Scheduling Algorithms in Cloud Computing. Journal of Network and Systems Management, **1-37**.

16. Cao, J., Cui, H., Shi, H., & Jiao, L. **(2016)**. Big Data: A Parallel Particle Swarm Optimization-Back-Propagation Neural Network Algorithm Based on MapReduce. PLoS One, 11(6), e0157551.

17. Kumari, K. R., Sengottuvelan, P., &Shanthini, J. **(2017)**. A Hybrid Approach of Genetic Algorithm and Multi Objective PSO Task Scheduling in Cloud Computing. Asian Journal of Research in Social Sciences and Humanities, 7(3), **1260-1271**.

18. Modi, C., Patel, D., Borisaniya, B., Patel, H., Patel, A., &Rajarajan, M. **(2013)**. A survey of intrusion detection techniques in cloud. Journal of Network and Computer Applications, 36(1), **42-57**.

19. Devikrishna, K. S., & Ramakrishna, B. B. **(2013)**. An artificial neural network based intrusion detection system and classification of attacks. International Journal of Engineering Research and Applications (IJERA), 3 (4), **1959-1964**.

20. Minal, Z., Pooja, D., Snehal, P., Poonam, P., & Priyanka, P. **(2014)**. Intrusion Detection System Using Artificial Neural Network. International Journal of Emerging Engineering Research and Technology, 2(6), **146-149**.

21. Dogra, R., & Gupta, N. **(2014)**. Glowworm Swarm Optimization Technique for Optimal Power Flow, Advance in Electronic and Electric Engineering, 4 (2), **155-160**

22. Liu, J., Zhou, Y., Huang, K., Ouyang, Z., & Wang, Y. **(2011)**. A glowworm swarm optimization algorithm based on definite updating search domains. Journal of Computational Information Systems, 7(10), **3698-3705**.

23. Zhou, Y., Luo, Q., & Liu, J. **(2014)**. Glowworm swarm optimization for dispatching system of public transit vehicles. Neural processing letters, 40(1), **25-33**.

24. Lee, C. W., & Lin, B. Y. **(2016)**. Application of Hybrid Quantum Tabu Search with Support Vector Regression (SVR) for Load Forecasting. Energies, 9(11), **873**.

25. Zainal, N., Zain, A. M., Radzi, N. H. M., & Othman, M. R. **(2016)**. Glowworm swarm optimization (GSO) for optimization of machining parameters. Journal of Intelligent Manufacturing, 27(4), **797-804**.