Modified PSO based Intrusion Detection in Streaming Network Data

G. Banupriya[1], **M. Lalli**[2]

M. Phil Research Scholar, School of Computer Science and Engineering, Bharathidasan University, Trichy, India [1]

Assistant Professor, School of Computer Science and Engineering, Bharathidasan University, Trichy, India [2]

Abstract: Intrusion Detection is one of the prime functionalities of any Network Based System and it reflects the quality of the network. The existing intrusion detection systems performs in a fairly predictable and reliable manner, however, on the other side, hackers and attackers are always ahead, and they keep conjuring new attacks in a constant and consistent manner. The only way to solve this issue is to detect or predict an attack as soon as possible, in real time and if possible to prevent them from happening all together. Detecting it after it has occurred is useless, as the damage is already done. However, the current systems used for detecting intrusions are not capable of providing real time protection against such attacks. Latencies in the detection structure has become unavoidable due to the hugeness of the data associated with it. This paper presents an Intrusion Detection System based on PSO, a well-known metaheuristic algorithm. Regular PSO is hybridized with Simulated Annealing to provide a more effective and faster detection mechanism. Experiments conducted on PSO-SA prove that the algorithm works effectively on networked data providing high quality results. Further, comparisons with regular PSO also indicate that the proposed technique performs much better than regular PSO in detecting intrusions.

Keywords: PSO; Simulated Annealing; Network Intrusion Detection; KDD.

I. INTRODUCTION

Intrusion is the process of attempting illegal access on a computer system or a network. Existence of intrusions have been recorded ever since the first of the computer networks were created. Since the first intrusion attempt, several mechanisms have been developed to quickly detect and stop these attacks [1]. However, the evolution has been mutual. As the system changes or strengthens its boundaries, the intruders are also becoming more powerful and the attacks metamorphose back more powerful to compromise the system. This has been an unavoidable cycle that had been repeating since the initial attacks [2][3].

Due to the availability of Botnets and other such features, the intensity of intrusions has seen manifold increase. Several online lending mechanisms has also increased the complexity of creating an Intrusion Detection System (IDS). Until recently, attacks and intrusion attempts were performed by a specific few, however now it is available as a service to everyone. Besides there is no specified limit to the level of attack, since these services are created and lent to users on an hourly basis [4][5].

In the current generation of high performance computing, the level of attacks has seen a huge increase. Even though network data is semi-structured, the hugeness of the data and the speed at which the data needs to be processed to detect intrusions makes it a Big Data problem [6]. Hence by the basic definition of Big Data, it becomes clear that intrusion detection, in the current scenario cannot be processed using traditional algorithms [7].

II. RELATED WORKS

An agent based intrusion detection mechanism that uses probability of the attacks is proposed by Gowadia et al [8]. A set of cooperating agents try to resolve the problem by performing intrusion detection. The intrusion detection task is performed in a distributed fashion where each agent is assigned a separate task. The agents are also capable of self-evidential updates; hence continuous monitoring becomes possible. The individual agents depends on an XML based format for reliable communication. They exchange of information and registration requests happens in a similar fashion. The drawback of the approach include too many data transfers and high transfer costs associated with the transfers. Commercially available encryption techniques are used to ensure reliability and security.

Garcia et al [9] had discussed the various challenges, techniques and systems for intrusion detection in detail. The various challenges encountered during the process of intrusion detections is clearly explained along with the technological solutions available to overcome the same. The authors have also provided a taxonomy of the various behavior based intrusion detection techniques in detail.

The agent based system proposed by Dasgupta et al [10], describes effective detection of abnormal network traffic. The proposed work was capable of detecting small deviations and abnormalities in the network traffics and also found to be robust in detecting the various faults and malfunctions. The work provides effective recommendations based on user behavior. The greatest advantage of the above mentioned work is that it is a real

DOI 10.17148/IJARCCE.2016.57128

time system capable of monitoring the network effectively and reporting security threats instantaneously at multiple levels of the network.

The Cougaar framework is used for implementing agents. Each node is provided with four agents. The framework is based on a modular design and incorporates the flexibility to create on the fly action and decision rules. This makes the system dynamic. It uses a swing based GUI, which provides an effective UI for monitoring the network abnormalities. It acts as a robust system for detecting the Denial of Service and probing attacks.

The current requirement is not to just build an effective intrusion detection system, instead to provide an Intrusion Detection System that is fast enough for detecting intrusions in real time. Wang et al [11], Ammann et al [12] and Wing et al [13] proposed a graph based system to predict network intrusions in linear time which is achieved by time memory tradeoff with quadratic space utilization. A Queue Graph integrated with the latest intrusion signatures is effectively used to predict the abnormalities in a fast and effective manner. The processing overload is overcome by discarding the historical and obsolete signatures and replacing them with the latest signatures. Also the proposed work is found to work effectively in detecting the noisy and borderline data points.

Several conventional intrusion detection approaches based on statistical elements [14,15,16] have also been used in the detection process. The advantages of using statistical methods is that the profile size for real-time intrusion detection can be minimized. However, the usage of statistical operators alone cannot provide best results. Their major downside is that false positives cannot be avoided. Furthermore, the statistical methods cannot handle uncommon but periodically occurring activities. An unsupervised clustering based intrusion detection system is proposed by Leonid Portnoy [17]. It is found to detect unknown and new types of attacks effectively in comparison to supervised approaches. introduced a clustering algorithm to detect both known and new intrusion types without the need to label the training data.

III.OUR APPROACH

Network Intrusion Detection in real time is a tedious task requiring faster and more accurate strategies in order to provide effective results. The proposed architecture presents a modified Particle Swarm Optimization based intrusion detection technique for network data.

Particle Swarm Optimization (PSO) [18, 19] is a metaheuristic technique optimizes a given problem by trying to improve a candidate solution iteratively with regard to a given measure provided by the fitness function. The solution is identified by having a set of candidate solutions, referred here as particles and by moving these particles in the search space in search of better solutions. The movement of particles is triggered by the particle's position and the velocity component associated with the particle.

The operation of PSO is carried out in three phases namely, the particle initialization phase, particle movement or acceleration phase and finally the convergence phase.

The search space is initially built using the boundaries defined by the data. The boundaries are defined by the intrusion data and the attributes contained in the data corresponds to the dimensions of the search space. This is followed by the particle initialization phase. The particles are distributed in a uniform manner on defined solutions. These distributions are carried out within search space boundaries. The initial velocities of particles are identified using eq. (1)

$$V_i \sim U(-|b_{up} - b_{lo}|, |b_{up} - b_{lo}|) \qquad (1)$$

where b_{up} and b_{lo} are the upper and lower bounds of the search space.

The next phase defines particle movement. Particle acceleration is initiated using the velocity component assigned in the previous phase. The velocity component contains acceleration distance and direction of movement for every dimension of the search space.

After the initial movement, further movements are determined by two intrinsic components namely, the particle best (pbest) and the global best (gbest) values. Every particle in the solution maintains a pbest containing the best solution visited by it so far. A gbest shared by the entire swarm is also maintained, which maintains the best of the pbest values.

The initial identification of pbest and gbest values mark the beginning of the convergence process. The particles are directed towards the region of pbest and gbest, directed by eq. (2)

$$V_{i,d} \leftarrow \omega V_{i,d} + \varphi_p r_p (P_{i,d} - X_{i,d}) + \varphi_g r_g (g_d - X_{i,d}) \qquad (2)$$

where r_p and r_g are the random numbers, $P_{i,d}$ and g_d are the parameter best and the global best values, $x_{i,d}$ is the value current particle position, and the parameters ω, φ_p, and φ_g are selected by the practitioner.

The process of identifying the global best value is usually carried out as every particle completes its movement and reaches a standard node. This contribution alters that scheme by identifying the global best only after all the particles complete their first hop movement. This provides a set of particle best solutions from which the best solution can be selected to be the global best. The process of selecting the global best is performed using Simulated Annealing [20]. The process begins with a single randomly selected solution from the set of pbest values. It moves in a random manner for a defined number of steps or until a solution with the best result is obtained. This constrains both the time and also makes it certain that the

DOI 10.17148/IJARCCE.2016.57128

operations are carried out in a random that helps us avoid getting struck in the local minima.

Algorithm (PSO embedded with Simulated Annealing):

```
1. Building search space using the base data
2. For each particle i=1…p
   a. Initialize particle, pbest and gbest
   b. Initialize velocity using the search space boundaries ($b_{lo}$,$b_{hi}$)
3. Perform the following until stagnation behaviour is encountered
   a. For each particle i=1…p
      i. Generate $r_p$ and $r_g$ using normal distribution
      ii. Identify the particle velocity
      iii. Discretize the particle movement to a defined node
      iv. Position update for the particle
      v. If pbest< current fitness
         1. Assign current fitness to be the pbest
   b. gbest<- Simulated Annealing (gbest,pbest,p)
4. gbest contains the best found solution
```

Simulated Annealing(gbest,pbest,p)

```
1. Let s = gbest
2. For k = 1 through p :
   a. T ← $pbest_k$
   b. Pick a random pbest (pb), $s_{new}$ ← pb
   c. If P(E(s), E($s_{new}$), T) ≥ random(0, 1), move to the new state:
      • s ← $s_{new}$
3. Output: the final state s
```

P(e,e’,T) was defined as 1 if e’ < e and exp(-(e’-e)/T) otherwise.

This process is repeated until the particles converge into a point and stop providing the same solution. The point of convergence can be used to identify if the packet is legitimate or intrusion. In real time, due to the hugeness of the data, it is not advisable to wait for convergence, hence a time limit is imposed on the system. On reaching this time limit, the process is terminated and the current gbest is taken as the final solution.

IV. RESULTS AND DISCUSSION

PSO and PSO-SA were implemented using C#.NET using Visual Studio 2012. KDD CUP 99 data, the benchmark data is used as the detection process. PSO cannot handle nominal values, while KDD contains nominal data. Hence the data is normalized and passed to PSO and PSO-SA and the detection performance is measured.

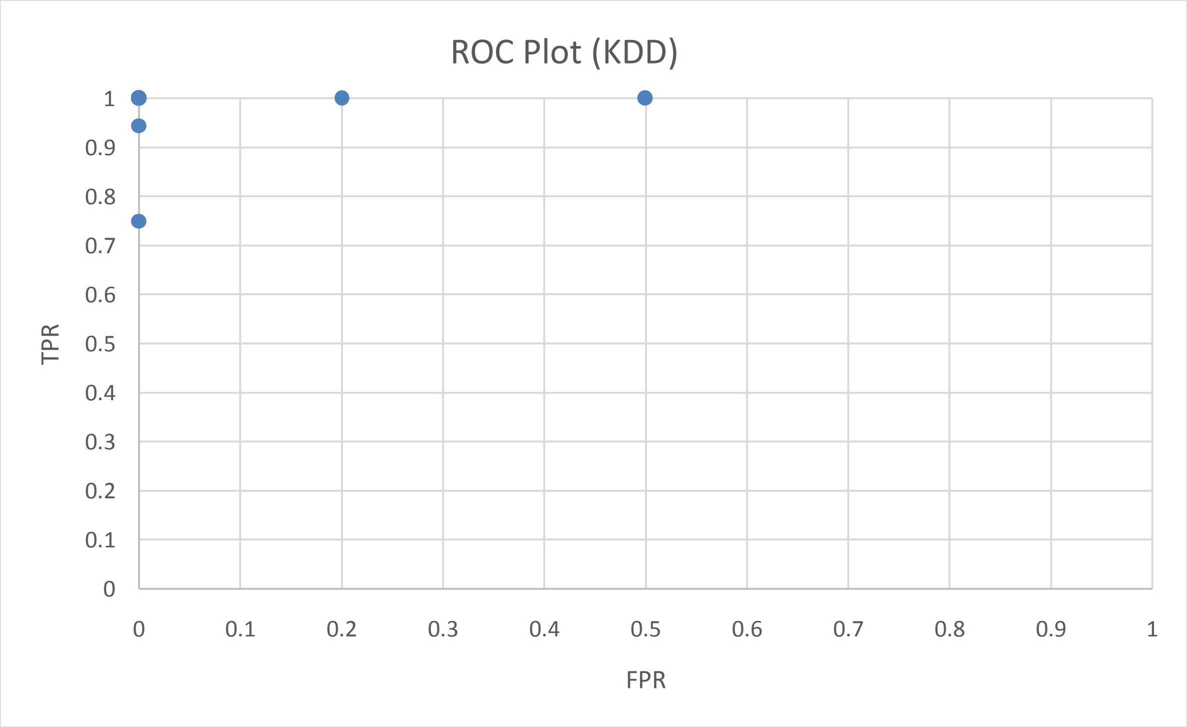

Fig.1 ROC Plot (KDD)

The ROC plot pertaining to the KDD dataset is presented in figure. It could be observed that the point in ROC plot is concentrated in major on the top left. Hence it could be concluded that the KDD dataset, when operated on PSO-SA exhibits very high true positive rates and low to moderate false positive rates. This property makes PSO-SA the best choice for detecting intrusions in a network environment.

DOI 10.17148/IJARCCE.2016.57128

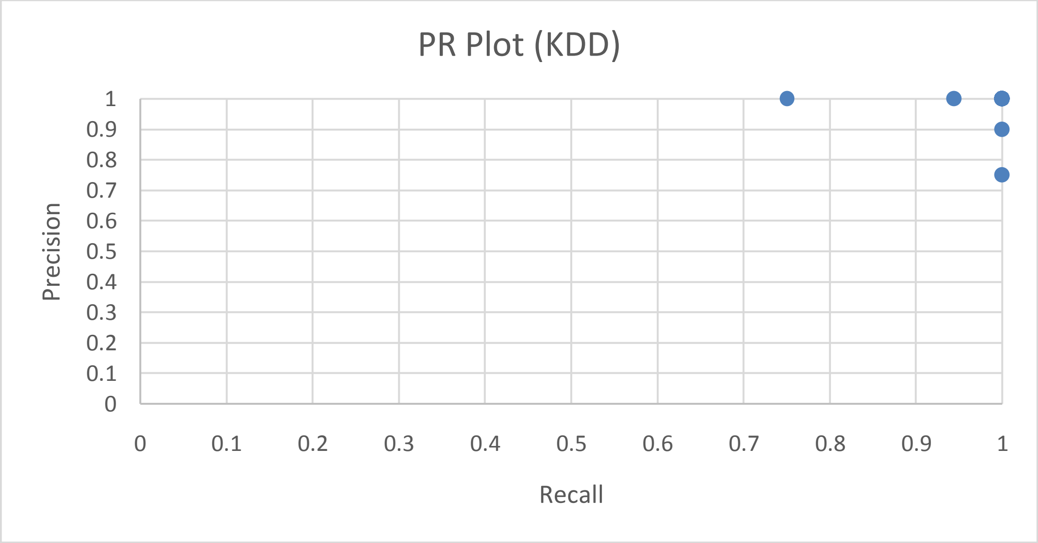

Fig. 2 PR Plot (KDD)

The PR plot pertaining to the KDD dataset is presented in figure. It could be observed that the points in PR plot is concentrated in the top right, exhibiting very high precision and recall levels. Hence it could be concluded that the KDD dataset, when operated on PSO-SA exhibits very high precision and recall rates showing that the algorithm can exhibits very high retrieval rates.

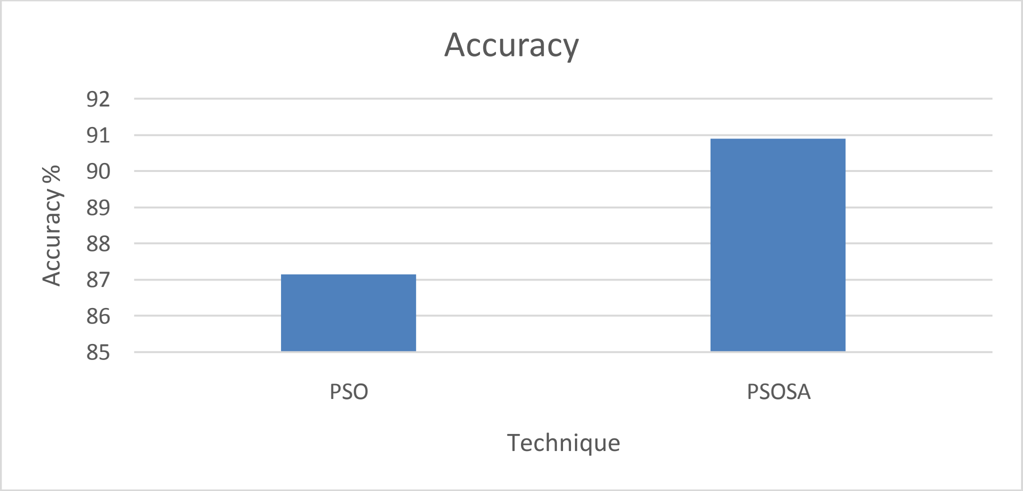

Fig.3 Accuracy

A comparison between the accuracy exhibited by PSO and PSO-SA is presented in Figure. It could be observed that PSO-SA exhibits higher accuracy than normal PSO. This shows the applicability of the algorithm on streaming network data for highly efficient results.

V. CONCLUSION

This contribution presents PSO-SA, a modified form of PSO as a probable solution for identifying intrusion. The optimality of PSO-SA was observed to provide effective solutions both in terms of accuracy and time. The presented technique also incorporates the flexibility of automatically tuning itself to new data. Further, new training data can be dynamically added, which also proved to be another major advantage of PSO. Hence PSO is considered as the most optimal technique for intrusion detection. Even though PSO will work effectively while using preprocessed data, they also provide considerably good results even without data preprocessing. Future contributions will deal with providing an effective integrated pre-processing and feature selection mechanism that can be used to improvise the processing of PSO.

REFERENCES

[1] K. Jackson, “Viewgraphs on Intrusion Detection: User Authentication Profiles at Los Alamos”, Los Alamos National Laboratory, 1989.

[2] H. S. Javitz , A. Valdes , D. E. Denning and P. G. Neumann, “Analytical Techniques Development for a Statistical Intrusion Detection System (SIDS) based on Accounting Records”, SRI International, 1986.

DOI 10.17148/IJARCCE.2016.57128

[3] T. F. Lunt, (1988), “Automated Audit Trail Analysis and Intrusion Detection: A Survey", Proceedings of the 11th National Computer Security Conference, 1988.

[4] H. Tiirmaa-Klaar, J. Gassen, E. Gerhards-Padilla, and P. Martini, “Botnets: how to fight the ever-growing threat on a technical level.” In Botnets (pp. 41-97). Springer London, 2013.

[5] S.Silva, et al., "Botnets: A survey", Computer Networks 57.2: 378-403, 2013.

[6] F.daCosta, and D. Francis, "Small Data, Big Data, and Human Interaction", Rethinking the Internet of Things: A Scalable Approach to Connecting Everything : 77-94, 2013.

[7] S. Feifei Li, "Scalable data summarization on big data", Distributed and Parallel Databases (Impact Factor: 0.81), 32(3). DOI: 10.1007/s10619-014-7145-y. 2014.

[8] V. Gowadia, C. Farkas , M. Valtorta. “Paid: A probabilistic agent-based intrusion detection system.” Computers & Security 24(7):529–545, 2005.

[9] G. Teodoro, et al. “Anomaly-based network intrusion detection: Techniques, systems and challenges.” computers & security 28.1: 18-28, 2009.

[10] D. Dasgupta, et al. 2005. “CIDS: An agent-based intrusion detection system.” Computers & Security 24.5: 387-398, 2005.

[11] W. Lingyu, A. Liu, and S. Jajodia. “Using attack graphs for correlating, hypothesizing, and predicting intrusion alerts.” Computer communications 29.15: 2917-2933,2006.

[12] P. Ammann, D. Wijesekera, S. Kaushik. “Scalable, graph-basednetwork vulnerability analysis,” in: Proceedings of the 9th ACMConference on Computer and Communications Security (CCS’02), pp. 217–224, 2002.

[13] O. Sheyner, J. Haines, S. Jha, R. Lippmann, J.M. Wing. “Automatedgeneration and analysis of attack graphs,” in: Proceedings of the 2002IEEE Symposium on Security and Privacy (S& P’02), pp.273–284, 2002.

[14] H.S.Javitz and A.Valdes, "The NIDES Statistical Component Description and Justification," Annual report, SRI International, 333 Ravenwood Avenue, Menlo Park, CA 94025, March 1994.

[15] A. P. Phillip, and P. G. Neumann, "EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances," 20th NISSC, October 1997.

[16] H.S. Javitz, A. Valdes, "The SRI IDES Statistical Anomaly Detector," In Proc. of the 1991 IEEE Symposium on Research in Security and Privacy, May 1991.

[17] L. Portnoy, E. Eskin, S.Stolfo, “Intrusion Detection with Unlabeled Data Using Clustering”, ACM CSS Workshop on Data Mining Applied to Security, pp. 5–8. ACM Press, Philadelphia, 2001.

[18] J. Kenndy, and R. C. Eberhart, “Particle swarm optimization.” InProceedings of IEEE International Conference on Neural Networks (Vol. 4, pp. 1942-1948) 1995.

[19] Y. Shi, and R. Eberhart, “A modified particle swarm optimizer.” InEvolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Conference on (pp. 69-73),1998.

[20] S. Kirkpatrick, C. Gelatt Jr, and M.P.Vecchi, “Optimization by Simulated Annealing.” Science 220(4598):671–680, 1983.

DOI 10.17148/IJARCCE.2016.57128