Accuracy Improvement in Intruder Detection using Optimally Selected Features

[1]Aastha, [2]Shivani Gaba

[1]M.Tech Scholar, Computer Science and Engineering, N.C. College of Engineering, Panipat

[2]Assistant Professor, Computer Science and Engineering, N.C. College of Engineering, Panipat

Abstract: In this work we use a MANET collection of mobile nodes. It reduce intrusion but cannot eliminate them. Intrusion detection in MANET is the task which can be related to the machine learning field. This work mainly feature reduction to get maximum accuracy and reduce the time overhead machine learning algorithms. The intrusion dataset is taken from standard NSL-KDD dataset. The genetic algorithm (GA) is replaced by Gravitational search algorithm (GSA) in this work which provide highest accuracy and consume less time in training the model.

I. INTRODUCTION

MANET is a very dynamic and continuously changing ad-hoc network, so to have a centralise monitoring on it is not possible. VANET is like MANET in which vehicles keeps on communicating with nearby vehicle and road side unit. It is highly dynamic in nature. To detect the intruder in it is very challenging task. An intrusion detection system (IDS) monitors network traffic and alerts the system or network administrator. IDS may also respond to anomalous traffic by blocking the user or source IP address from accessing the network. [1]

Some environments (such as the military tactical operations) have very stringent requirements on security, which make the deployment of security-related technologies necessary. Intrusion prevention measures, such as encryption and authentication, can be used in MANETs to reduce intrusions, but cannot eliminate them. For example, a physically captured node that carries the private keys may allow the defeat of the authentication safeguards. The history of security research has demonstrated that no matter how many intrusion prevention measures are used, there are always some weak points in the system [1][4]. In a network with high security requirements, it is necessary to deploy intrusion detection techniques. MANET IDSs, serving as the second wall of defence to protect MANETs, should operate together with prevention mechanisms (authentication, encryption etc.) to guarantee an environment with highsecure requirements. They should complement and integrate with other MANET security measures to provide a high-survivability network. However, most of today's Intrusion Detection Systems (IDSs) focus on wired networks. The dramatic differences between MANETs and wired networks make it inapplicable to apply traditional wired ID technologies directly to MANETs. MANET does not have a fixed infrastructure. While most of today's wired IDSs, which rely on real-time traffic parse, filter, format and analysis, usually monitor the traffic at switches, routers, and gateways. The lack of such traffic concentration point makes traditional wired IDSs inapplicable on MANET platforms. Each node can only use the partial and localized communication activities as the available audit traces. There are also some characteristics in MANET such as disconnected operations, which seldom exist in wired networks. What's more, each mobile node has limited resources (such as limited wireless bandwidth, computation ability and energy supply, etc.), which means MANET IDSs should have the property to be lightweight. All of these imply the inapplicability of wired IDSs on the MANET platform. Furthermore, in MANETs, it is very difficult for IDSs to tell the validity of some operations. For example, the reason that one node sends out falsified routing information could be because this node is compromised, or because the link is broken due to the physical movement of the node. All these suggest that an IDS of a different architecture needs to be developed to be applicable on the MANET platform.

Intrusion Detection in MANET's or adhoc networks is the task which can be related to machine learning field. The data set is available with NSL-KDD data for intrusion detection. On the basis of which forthcoming intruder whether that can be selfish node in the network, any malicious node or any Sybil node, can be detected as anomaly node. The dataset consists of previous history of intruders which are field names and their numerical values. This dataset is very large so dimensionality reduction has to be performed to select the best suitable features which gives highest accuracy and also consumes less time in training the model.

II. DATASET DESCRIPTION

The NSL-KDD data set is the improvement over KDD'99 data set. The NSS-KDD data set has 42 attributes that are used in empirical study. In NSS-KDD data set duplicate instances were removed to get rid of biased classification results. Many numbers of versions are available of this dataset, out of which 20% of the training data is used. Training data is identified as KDDTrain+_20Percent with a total number of 125973 instances. The test data is identified as KDDTest+ and it has a total of 22543 instances. The number of attributes in each is 42. With variations in number of instances, different types of configurations of the data set are available. Labelled attribute 42 is the 'class' attribute which indicates that the given instance is normal connection instance or an attack. In table 1 the description of KDD dataset attributes with class labels are shown. Out of 42, 41 attributes can be classified into four different classes as discussed below:

- Basic (B) Features are the attributes of individual TCP connections.
- Content (C) features are the attributes within a connection suggested by the domain knowledge
- Traffic (T) features are the attributes computed using a two-second time window
- Host (H) features are the attributes designed to assess attacks which last for more than two seconds.

The dataset description table is provided in to appendix A.1.

III. OPTIMIZED FEATURES SELECTION FROM DATASET

The dataset has 42 attributes for each sample and not all samples contribute to accuracy due to either all zeros in the attribute or not defined values in that. Also some non relevant attributes are also provided with the data which may reduce the accuracy. So we need to find out them and remove those to improve the accuracy. For this purpose we rely upon gravitational search algorithm (GSA) which is a global meta heuristic optimization method based on movements of planets in an orbit. Each planet attracts other with its gravitational force and force of attraction will be more for which mass is higher. Here the mass represents the accuracy in intruder detection for a particular set of attributes. As the time increases, the orbit gets longer and gravitational constant which is to be considered a constant so far will decay with time as in equation 3.1.

$$G(t) = G_0 e^{-\alpha t/T} \qquad (3.1).$$

G0 and α are initialized at the beginning and will be reduced with time to control the search accuracy. *T* is the total number of iterations. These iterations are defined by user and maximum iterations changes from application to application. Within these iterations, the maximum accuracy for a chosen set of attributes must be attained and no more increase in accuracy should be there. In other words a saturation state of accuracy in these iterations should be obtained which means the system has attained and located the optimal set of attributes for which accuracy is maximum. GSA method does this task iteratively. The positions of agents in GSA are represented by binary digits. The 1 represents the attribute is chosen and 0 represents that attribute is discarded. A total of 41 0's and 1's series represent the agent's positions. In an iteration, there are 20 agents which have 20 set of 0's and 1's series. We calculate the accuracy for all those 20 different attributes combinations using SVM (Support vector machine) classification method and a matrix is generated for these 20 values. All agent's positions are updated by following equations:

$$x_i^d(t+1) = v_i^d(t+1) + x_i^d(t) \quad \dots (3.2)$$

$$v_i^d(t+1) = rand_i x v_i^d(t) + a_i^d(t) \quad \dots (3.3)$$

where $x_i^d(t+1)$ is the new agent's position for the next iteration and $x_i^d(t)$ is the present position. $v_i^d(t+1)$ is the new velocity of movement and $a_i^d(t)$ is the present acceleration. This can be further calculated as:

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \dots\dots (3.4)$$

$F_i^d(t)$is the total force acting on ith agent calculated as:

$$F_i^d(t) = \sum_{j \in kbest} rand_j F_{ij}^d(t) \quad \dots\dots (3.5)$$

$F_{ij}^d(t)$Can be computed as:

$$F_{ij}^d(t) = G(t).\left(M_{pi}(t) \times \frac{M_{ai}(t)}{R_{ij}(t)} + \varepsilon\right).\left(x_j^d(t) - x_i^d(t)\right) \quad \dots\dots (3.6)$$

$M_{ai}(t)$ is the mass of an agent which is normalised accuracy value for each agent. It is formulated as:

$$m_i(t) = \frac{fit(t) - worst(t)}{best(t) - worst(t)} \qquad (3.7)$$

where $fit(t)$ is the fitness value of each agent, $worst(t)$ is the minimum accuracy value among all present agents and $best(t)$ is the maximum accuracy value.

The new updated position of all 20 agents as in equation 3.2 is used to check the accuracy for them. Another matrix is generated which stores the accuracy for all those new set of attributes. This process continues till all iterations are not finished. Finally the maximum accuracy indexed agent's position is considered as the optimal set of attributes. For these attributes only accuracy obtained considered as maximum accuracy. This way it reduces the overhead in classification and improves the time and test accuracy. Overflow pipeline for the work is shown in appendix A.2.

IV. RESULTS

In our work we have proposed a comprehensive study on the application GSA (Gravitational Search Algorithm)optimization algorithm for feature reduction for making a better intrusion detection system (IDS). The proposed work is implemented in MATLAB R 2017a. A lot of inbuilt functions in MATLAB makes the use easier and saves our time to build our code from scratch, so we can use that time in problem solution of research. Dataset contains 125973 training data set and 22543 testing data set with total 41 features having 3 symbolic features and output is categorised as types of attacks. We applied GSA to reduce number of features from feature table of training and testing dataset. Now using SVM multi class classifier training data is used to create a training model and testing data is tested for predicted output labels using this trained model.

We have divided our results in four cases depending upon type of attack. There are four major types of attacks i.e. Denial-of-service (DoS), User to Root (U2R), Probing, and Remote to User (R2L) attacks.

Case-1 Denial of Service (DOS) Attack

In NSL-KDD dataset DOS attack is further categorised in six subtypes which are back, land, Neptune, smurf, pod and teardrop. We have compared performance of GSA based feature reduction with GA based feature reduction. Figure 4.1 shows the GSA iteration curve which is settled at maximum accuracy after 40 iterations and this one is its convergence point.

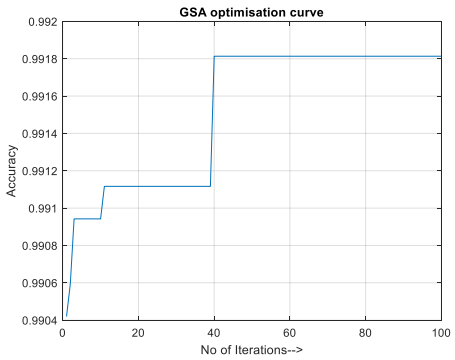

Figure 4.1: Iteration curve for DOS attack

It is observed that proposed method has more accuracy, precision, recall, F measure, sensitivity and specificity for DOS type of attack. For some attack such as pod type DOS attack proposed method gives non-zero value as against standard GA method.

Case-2 Probe Attack

Probe attack is further subcategorized as 4 types namely ipsweep, nmap, portsweep and satan. The accuracy and F-measure curve are shown in figure 4.2 and 4.3.

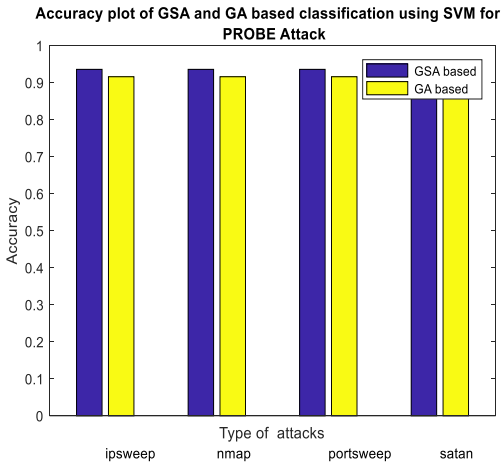

Figure4.2 : Accuracy comparison between GA and GSA selected features for PROBE attack

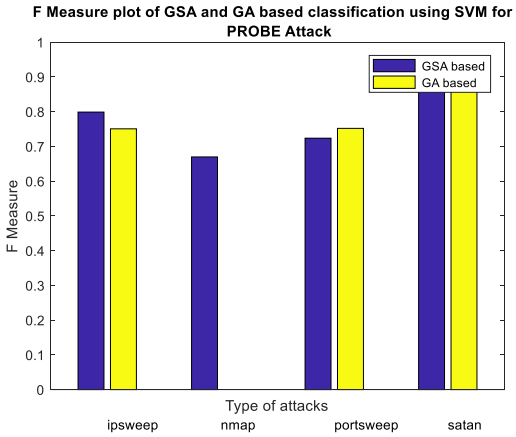

Figure 4.3: F-measure comparison between GA and GSA selected features for PROBE attack

It can be clearly checked that accuracy and F-measure are higher than GA for GSA.

Case-3User to Root (U2R) Attack

U2R attack is further subcategorized as 4 types namely bufferoverflow, loadmodule, perl and roortkit.

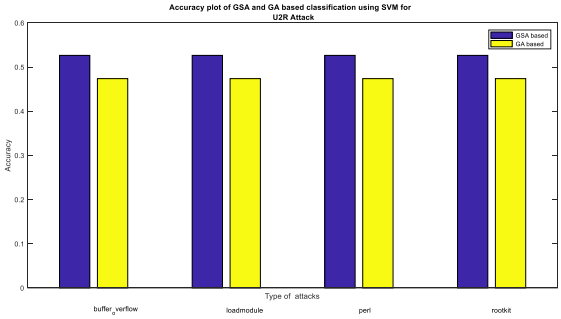

Figure 4.4: Accuracy Comparison for U2R attack

Case-4Remote to User (R2L)Attack

R2L attack is further subcategorized as 8 types namely ftpwrite, guesspasword, imap, multihop, phf, spy, warezclient, warezmaster.

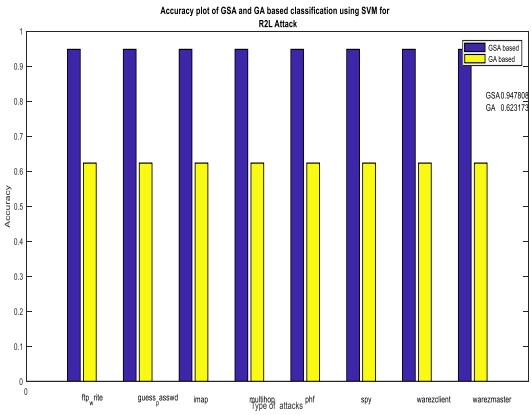

Figure 4.5: Accuracy Comparison for r2L attack.

In this case the proposed optimization gives better accuracy than all other cases. The improvement is higher.

V. CONCLUSION

MANET provide safety from the attacks. Machine learning field provide the solution of this problem. This work used public available NSL-KDD dataset which is modified and filtered version of KDD cup dataset. The work provided great accuracy and consume less time as compared to previous one. GSA is used only for those features which actually takes part in attack detection .The older one use the GA(genetic algorithm) but in this we use GSA(gravitational search algorithm) which results 97% accuracy in these attacks. GSA provide 53% more accuracy than the GA.

REFERENCES

[1] Z. Ullah, M. S. Khan, I. Ahmed, N. Javaid and M. I. Khan, "Fuzzy-Based Trust Model for Detection of Selfish Nodes in MANETs," *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, Crans-Montana, 2016, pp. 965-972.

[2] M. A. Abdelshafy and P. J. B. King, "Dynamic source routing under attacks," *2015 7th International Workshop on Reliable Networks Design and Modeling (RNDM)*, Munich, 2015, pp. 174-180.

[3] C. Alocious, H. Xiao and B. Christianson, "Analysis of DoS attacks at MAC Layer in mobile adhoc networks," *2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*, Dubrovnik, 2015, pp. 811-816.

[4] A. Quyoom, R. Ali, D. N. Gouttam and H. Sharma, "A novel mechanism of detection of denial of service attack (DoS) in VANET using Malicious and Irrelevant Packet Detection Algorithm (MIPDA)," *International Conference on Computing, Communication & Automation*, Noida, 2015, pp. 414-419.

[5] A. M. Shabut, K. P. Dahal, S. K. Bista and I. U. Awan, "Recommendation Based Trust Model with an Effective Defence Scheme for MANETs," in *IEEE Transactions on Mobile Computing*, vol. 14, no. 10, pp. 2101-2115, Oct. 1 2015.

[6] A. Menaka Pushpa and K. Kathiravan, "Resilient PUMA (Protocol for Unified Multicasting through Announcement) against internal attacks in Mobile Ad hoc Networks," *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Mysore, 2013, pp. 1906-1912.

[7] M. A. Abdelshafy and P. J. B. King, "Analysis of security attacks on AODV routing," *8th International Conference for Internet Technology and Secured Transactions (ICITST-2013)*, London, 2013, pp. 290-295.

[8] A. M. Kurkure and B. Chaudhari, "Analysing credit based ARAN to detect selfish nodes in MANET," *2014 International Conference on Advances in Engineering & Technology Research (ICAETR - 2014)*, Unnao, 2014, pp. 1-5.

[9] S. Biswas, P. Dey and S. Neogy, "Trusted checkpointing based on ant colony optimization in MANET," *2012 Third International Conference on Emerging Applications of Information Technology*, Kolkata, 2012, pp. 433-438.

[10] D. Das, K. Majumder and A. Dasgupta, "A game-theory based secure routing mechanism in mobile ad hoc network," *2016 International Conference on Computing, Communication and Automation (ICCCA)*, Noida, 2016, pp. 437-442.

[11] T. Poongothai and K. Duraiswamy, "Intrusion detection in mobile AdHoc networks using machine learning approach," *International Conference on Information Communication and Embedded Systems (ICICES2014)*, Chennai, 2014, pp. 1-5.

[12] D. A. Varma and M. Narayanan, "Identifying malicious nodes in Mobile Ad-Hoc Networks using polynomial reduction algorithm," *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, Chennai, 2016, pp. 1179-1184.

[13] Bandana Mahapatraa and Prof.(Dr) Srikanta Patnaik, "Self Adaptive Intrusion Detection Technique Using Data Mining concept in an Ad-Hoc Network," 2nd International Conference on Intelligent Computing, Communication & Convergence(ICCC-2016)

[14] Manjula C. Belavagi and BalachandraMuniyal, "Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection," 12th International Multi-Conference on Information Processing-2016 (IMCIP-2016).

[15] PreetiAggarwala and Sudhir Kumar Sharmab, "Analysis of KDD Dataset Attributes - Class wise For Intrusion Detection," 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015).

[16] Ciza Thomas, Vishwas Sharma and N. Balakrishnan, "Usefulness of DARPA Dataset for Intrusion Detection System Evaluation

[17] P.Natesan and P.Balasubramanie, "Multi Stage Filter Using Enhanced Adaboost for Network Intrusion Detection," International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.3, May 2012

[18] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Ottawa, ON, 2009, pp. 1-6.

Appendix A.1.

Table A.1. Classwise detail of KDD data set attributes

Serial .no.	Attribute name	Serial no.	Attribute name	Serial no.	Attribute name	Serial no.	Attribute name
1	Duration	11	Num_failed_logins	21	Is_hot_login	31	Srv_diff_host_rate
2	Protocol_type	12	Logged_in	22	Is_guest_login	32	Dst_host_count
3	Service	13	Num_compromised	23	Count	33	Dst_host_srv_count
4	Src_bytes	14	Root_shell	24	Serror_rate	34	Dst_host_same_srv_rate
5	Dst_bytes	15	Su_attempted	25	Rerror_rate	35	Dst_host_diff_srv_rate
6	Flag	16	Num_root	26	Same_srv_rate	36	Dst_host_same_srv_port _rate
7	Land	17	Num-file_creations	27	Diff_srv_rate	37	Dst_host_diff_srv_host_ rate
8	Wrong_fragm ent	18	Num_shells	28	Srv_count	38	Dst_host_serror_rate
9	Urgent	19	Num_access_files	29	Srv_serror_rate	39	Dst_host_srv_serror_rate
10	Hot	20	Num_outbound_cmd	30	Srv_rerror_rate	40	Dst_host_rerror_rate

Appendix A.2.

Complete pipeline for the work progress

41	Dst_host_srv_rerror_rate
42	Class