International Journal of Soft Computing 1(2) : 108-110, 2006

Neural Network Approach for Anomaly Intrusion Detection in Adhoc Networks Using Agents

S. Bose, P. Yogesh and A. Kannan
Department of Computer Science and Engineering,
College of Engineering, Guindy, Anna University, Chennai-25, India

Abstract: This study proposes a distributed intrusion detection system for adhoc wireless networks using self organizing maps and mobile agents. In this research, we efficiently use log file data obtained from the local host for training the neural network, to analyze the adhoc wireless network for detecting intrusions. Security agents are used to monitor multiple clients of the wireless network to determine the correlation among the observed anomalous patterns and to report such abnormal behavior to the administrator and the user in order to take possible actions. From the system developed in this research, we obtained high intrusion-detection rates (99.2%) and low false-alarm rates. The main contribution of this paper is the provision of an agent based framework that is capable of detecting intruders and to forecast the anomalies using the neural classifier, self organizing maps.

Key words: Intrusion detection system, mobile agent, adhoc network, SOM

INTRODUCTION

Intrusion detection is an essential component of network security in wireless adhoc networks. It acts as an additional layer of defense against computer misuse after physical, authentication and access control[1]. A mobile ad hoc network is a collection of wireless mobile hosts forming a temporary network without the aid of any established infrastructure or centralized administration. The flexibility in space and time induces new challenges towards the security infrastructure in such adhoc networks[2]. In this research, we used intelligent mobile agents for detecting and reporting anomalies obtained from the neural network classifier.

An agent is a small intelligent active object that accomplishes their essential tasks with minimal code. They are dynamically updatable and upgradeable, smaller, simpler and faster. Mobile agents are a special type of agents defined as processes capable of roaming through large networks such as the adhoc wireless network, interacting with machines, collecting information and returning after executing the tasks adjusted by the user[3].

In this study, we propose an efficient framework that targets intrusion at multiple levels of a distributed wireless adhoc network using mobile agents with less false alarm rates. In order to achieve this, the intrusion detection system uses Self Organizing Maps (SOM) for prediction. In this research, SOM neural network classifier has been chosen due to its ability to classify patterns into categories without human supervision. Since SOM is an unsupervised neural network that uses competitive learning, it is possible to provide automatic classification. In our SOM network, learning is accomplished by the application of log file data as input and hence no expected output data has been used to train the network. In this type of learning called unsupervised learning, all the data are unlabeled and all the units in the neighborhood that receive positive feedback from the winning unit participate in the learning process. Even if a neighboring unit's weight is orthogonal to the input vector, it will change its response to the input vector. In this research, first we obtain the log data and then classify it using neural networks and finally agents are used to monitor user behavior, address probing and take corrective actions. Comparing with the works present in the literature, our work is different in many ways. First, we provide a new framework that uses an integrated form of agents and neural networks[9]. Second we use it for adhoc networks. Third, the data files used in our work are organized in relational format so that they can be stored in relational database for effective retrieval. Finally, we used log file dataset generated from actual adhoc networks. The remainder of this paper is organized as follows: Section 2 provides the architectural framework and the system descriptions that form the core part of this work. Section 3 discusses the results obtained from our experiments. Section 4 gives the conclusions on this work and suggests some possible future enhancements.

Corresponding Author: S. Bose, Department of Computer Science and Engineering, College of Engineering, Guindy, Anna University, Chennai-25, India

SYSTEM ARCHITECTURE

Overview: The overall architectural framework of the system developed in this research is illustrated in Fig. 1. It has been implemented by using Aglets Software Development Kit (ASDK)[5]. Three major characteristics have influenced the design of the system using ASDK. They are dispatching aglets to perform a specific task on behalf of the user at a remote computer, without user interaction, saving the aglets to run at a later time and to maintain the state information about the environment using the aglets. The functions and relationships among the agents in the system are as follows:

Address probing agent: Address Probing agent is a static agent that provides a user friendly human-computer interface for the client user. Whenever a client is connected to the network, its address is informed to the server. These addresses are useful to the server agent to distribute the monitor agents to all connected client systems.

Server agent: Server agent is another static agent. It is always kept in the listen mode. This agent provides the friendly human-computer interface to the administrator. The server agent creates and sends the monitor agents to all the client nodes at particular intervals of time to detect the intrusions in the client nodes. Intrusion attempts are informed to the administrator, by analyzing the log information. The administrator monitors the entire network and removes suspicious client nodes.

Monitor agent: Monitor Agent are created and sent to each client node for monitoring. After reaching the client, the monitor agent creates forecasting agents and sends them to the forecasting engine for prediction. After prediction, the monitor agent will read the predicted files and finds the number of abnormal activities happened. Now it takes two actions: 1. Corrective Action for abnormal activity and 2. Information Action for the normal activities.

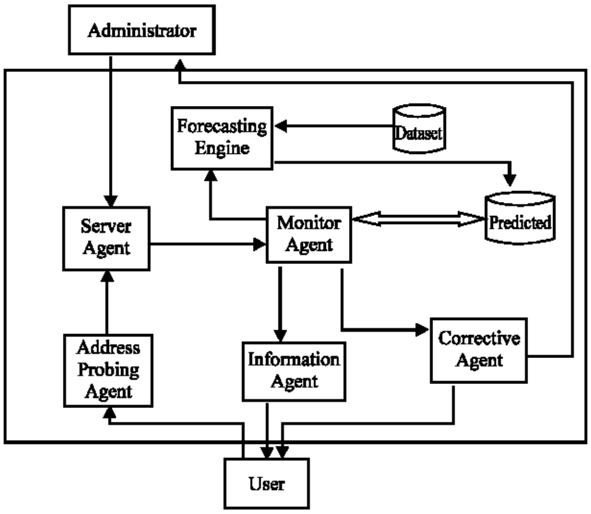

Fig. 1: System architecture

Action agent: If the monitor agent finds any single activity that is noted as abnormal, then it creates a corrective agent. Otherwise, it creates an information agent. Both action agents are used to visualize the predicted files and their contents and also the network details. The corrective agent is capable of removing the abnormal activity records from the predicted files and it also sends a report to the administrator about the intrusions happened.

Communicator agent: In this system, the basic mode of communication between any two agents is through message passing. In this framework, no two main agents are allowed to communicate directly with each other. However, they are intended to accomplish a particular task. Hence, they need to be static after they have been created or dispatched. In order to provide a communication between the client and the server, main agents create communicator agents to communicate.

Forecasting engine: The forecasting engine is on of the main component of our intrusion detection system. The incoming packets and user level information are stored in the corresponding data files. The forecasting engine takes the data from the files and use them as test data and also the offline dataset as training instances. Using SOM, the forecasting engine predicts the normal and abnormal instances and stores the results in the predicted files.

EXPERIMENTAL RESULTS

The SOM trains system with different number of instances and has been tested with various number of instances as given in Table 1. From this, we observe that increasing the number instances improves the prediction accuracy. Moreover, the increase in the number of instances, reduces false alarms. The result is shown in Table 1.

Table 1: Training results

No. of Train Instances	Training Percentage	Accuracy (%)	False Alarms	Time Taken (sec)
164328	100	99.2006	4	3.70
115252	70	98.6609	6	2.72
54352	33	98.2609	6	0.78
26516	16	77.971	36	0.36

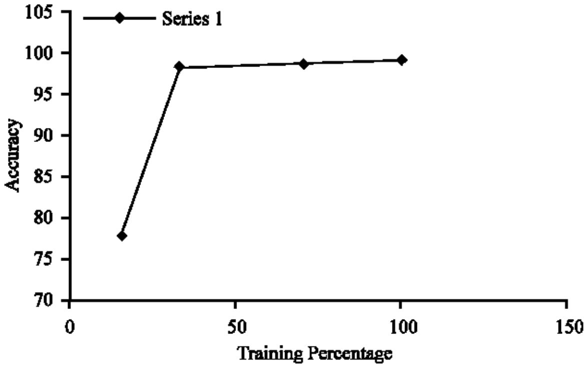

Fig. 2: Accuracy vs training percentage

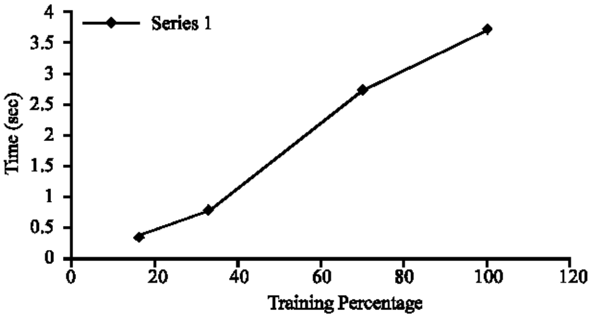

Fig. 3: Time vs training percentage

The graph, shown in Fig. 2 depicts the increase in accuracy with respect to the number of training instances. Figure 3 shows the increase in time which is an overhead with respect to training instances.

CONCLUSION

The developed system works efficiently and detects the intrusions at multi levels, namely, user and packet level. The SOM neural classifier predicts the accuracy rate of the system and is found as 99.2% in our work and the rate achieved for false alarms are comparatively less. Evidently, the experiment shows that training with more instances improves the prediction rate.

In future, we are focusing an improving the efficiency of the system further in order to reduce the time, even if it takes a large training dataset. To achieve this, it is necessary to select essential fields from the dataset using multi-level feature selection algorithms.

REFERENCES

1. Tim, Corthers, 2003. Implementing Intrusion Detection System. First Edition, pp: 17-156.
2. Ping, Yi, Yao Yan, Hou Yafei, Zhong Yiping and Zhang Shiyong, 2004. Securing Ad Hoc Networks through Mobile Agent. Proceedings of the 3rd ACM International Conference on Information Security, pp: 125-129.
3. Danny, B. and Lange Mitsuru Oshima, 1999. Mobile Agents with Java: The Aglet API. Addison-Wesley, Reading, MA.
4. Simon, G., A. Lendasse, M. Cottrell, J.C. Fort, M. Verleysen, 2005. Time Series Forecasting: Obtaining Long Term Trends with Self organizing maps, Pattern Recognition Lett., 26: 1795-1808.
5. IBM Tokyo Research Laboratory http://www.trl.ibm.co.jp/aglets.
6. Zhang and W. Lee, 2000. Intrusion Detection in Wireless Ad-Hoc Networks. Proceedings of the 6th Annual international Conference on Mobile Computing and Networking, MobiCom, pp: 275-283.