INTRUSION DETECTION SYSTEM USING GINI INDEX BASED TECHNIQUE

[1]G. Meghana Madhuri, [2]Dr. D. Vivekananda Reddy

[1]Master of Technology, S V University, Tirupati, [2]Assistant Professor, Department of CSE, S V University, Tirupati,
[1]Department of CSE, S V University College of Engineering, Tirupati,
[1]Department of CSE, S V University, Tirupati, India.

Abstract: An Intrusion Detection System is a module of the software that monitors the activities occurring in a computer system or network. An intruder tries to modify the packets or fabricate the malicious packets and insert them into a set of network flows on the network. It has the capability of detecting attacks which compromise the confidentiality, integrity or availability of network resources. The proposed method uses Supervised Learning in Quest (SLIQ), a fast scalable decision tree classifier for intrusion detection. NSL-KDD Dataset is used to study effectiveness of the classification method. Accuracy metric is considered as performance metric for the proposed method. The objective of this paper is to inspect the packets in the network flow using Supervised Learning in Quest and to predict whether the flow is malicious or normal flow.

Index Terms - **Intrusion Detection System (IDS), Supervised Learning in Quest (SLIQ), Gini Index, NSL KDD Dataset, Netbeans IDE.**

I. INTRODUCTION

The Confidentiality, Integrity, Availability of data on internet is a challenging task due to the increasing threats of malicious attacks. These attacks break the security strategies when a legitimate user accesses or transfers the data to other users. Therefore, a security system is needed that can detect these types of activities and alert the user. This type of system is called an Intrusion Detection System (IDS). An Intrusion Detection System monitors a standalone computer system or the network system for the activities and sends the report to the user or administrator informing whether these activities are normal or suspicious (intrusive) ones.

The Intrusion Detection Systems are of two types, namely, Host Based Intrusion Detection System (HIDS) and Network Based Intrusion Detection System (NIDS). The former type monitors an individual computer system and the later type monitors the network system. The Host Based Intrusion Detection System uses the rule based pattern matching system that compares the predefined file to the recent generated files. If the current file records depart from the predefined sets of file, then intrusion occurs and it is detected by the anomaly detection system.

In Network Intrusion Detection System (NIDS), when a message is transferred from source to destination application, it is transformed into the packets and again into the fragments. An intruder tries to modify the data, then he will change it in the inter network. While the fragments are about to be entered into the firewall, the NIDS at the firewall will check for the malicious data fragments, if any malicious data is found then the total flow is rejected and requested again by the destination application NIDS. In a network, huge amount of data packets move from a user's system to server and vice versa. Therefore, detection of intruders in a large dataset using a Supervised Learning in Quest (SLIQ) algorithm is very important. A Supervised Learning in Quest (SLIQ) algorithm uses the Pre-sorting technique while retaining a suitably high accuracy.

II. LITERATURE SURVEY

There are lists of data mining techniques such as frequent pattern mining, regression, clustering, association rule mining and many more, but out of these classification is frequently used at the most. In the Classification, the model is trained which describes and differentiate different data classes to predict the classes whose labels are not known. The Classification can be performed with the different algorithms such as the neural networks, decision trees, regression, etc. Due to the significant importance of the decision tree for large datasets, in this paper decision tree approach has been used.

Generally, the Classification is the sequence of the operations as follows:

- Prepare the training dataset using the pre-processing on the raw data.
- Class attribute and the classes are identified.
- Identify useful attributes for classification (Relevance analysis).
- Learn a model using training examples in Training Dataset.
- Use the model to classify the unknown data samples.

A decision tree represents the sequence of rules to form the class. It is a flow chart like tree structure. The decision tree consists of three fundamentals, root node, internal node and leaf node. Top most fundamental is the root node. Leaf node is the terminal fundamental of the structure and the nodes in between is called the internal node. Each of the internal node denotes test on the attribute, each branch represents an outcome of the test, and each leaf node holds a class label. Various decision tree algorithms are used in the classification like ID 3, CART, C5.0, SLIQ, SPRINT, Random forests, Random Trees.

In a network, huge amount of data packets move from a user's system to server and vice-versa. Therefore, detection of intruders in a large dataset using a Supervised Learning in Quest (SLIQ) algorithm is very important. A Supervised Learning in Quest (SLIQ) algorithm uses pre-sorting technique in the tree growing phase and an inexpensive pruning algorithm which retains a suitably high accuracy in representing the original features. It provides data pattern knowledge and visualizes the process. This provides data reduction, limiting storage requirements, and perhaps helping in reducing the costs. This has simplicity, possibility of using simpler model and gaining speed. This model defines the expected accuracy of a predefined learning algorithm. There have been done several comprehensive studies on feature selection and classification methods to select the best subset of features to improve the accuracy of classification methods.

III. EXISTING SYSTEM

The Existing System is based on the Feature Selection Algorithm which used the entropy information theory. This algorithm finds the minimum feature subset while retaining a suitably high accuracy in representing the original features. It calculates the entropy of each of the features after classification of features and the classification of features is based on the feature's attributes.

In this algorithm, first the entropy of the total dataset is calculated. The dataset is then split on the different attributes. The entropy for each branch is calculated. Then it is added proportionally, to get the total entropy for the split. The result is the Information Gain or decrease in entropy. The attribute that yields the largest Information Gain is chosen for the decision node.

Disadvantages:

- In this system, data may be over-fitted, if a small sample is tested.
- The detection and accuracy rate of this algorithm is less.
- The computational cost of this algorithm is very high.
- The gaining speed is also very less and hence the efficiency rates are also very less.

IV. PROPOSED SYSTEM

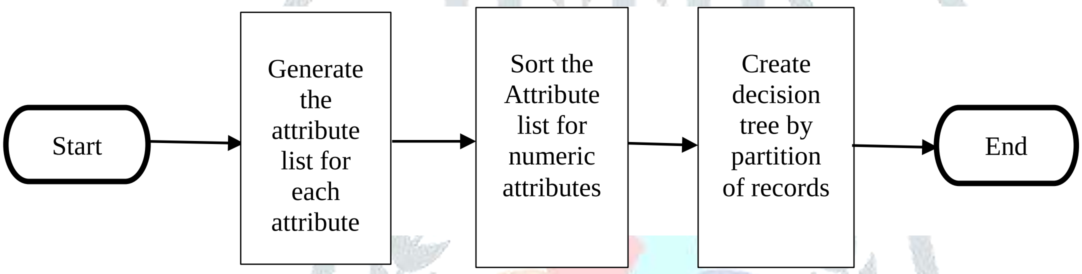
Start
Generate the attribute list for each attribute
Sort the Attribute list for numeric attributes
Create decision tree by partition of records
End

Fig 1: SLIQ Methodology

The Proposed System is the Supervised Learning in Quest (SLIQ) using Gini Index. It is a decision tree classifier. It is applied to both the numerical and categorical attributes. This builds the compact and accurate trees. This algorithm uses the pre-sorting technique in the tree growing phase and an inexpensive pruning algorithm. It is suitable for classification of large disk, resident datasets and independent of number of classes, attributes and records. This algorithm uses the Gini Index for the splitting criteria instead of information gain.

The Supervised Learning in Quest Algorithm involves the following steps:

1. Start Pre-sorting of the samples.
   - For each attribute, create an attribute list with columns for the value, sample -id and class.
   - Create a class list with columns for sample-id class and the leaf node.
   - Iterate over all training samples for each attribute.
   - Insert its attribute values, sample-id, the class and the leaf node into the class list.
2. As long as the stop criterion has not been reached.
   - For every attribute, place all the nodes in the class histogram and start evaluation of the splits.
   - Choose a split.
   - Update the decision tree for each node and update its class list.
     - Traverse attribute list of the attribute used in the node.
     - For each entry (value, id), find the matching entry (id, class, node) in class list.
     - Apply split criterion emitting a new node.
     - Replace the corresponding class list entry with (id, class, new node).

The Proposed System provides many advantages such as:

- It provides high accuracy, data reduction, limiting storage requirements and perhaps helping in reducing costs.
- It provides data pattern knowledge and visualizes the process.
- It has simplicity, possibility of using simpler model and gaining speed.

V. DATASET

NSL-KDD Dataset is used to study the effectiveness of the classification method. It is the refined version of the predecessor KDD 99 dataset. It is used to train and test various existing and new attacks. It contains the essential records of the complete KDD dataset. It does not include redundant records in the train set, so the classifiers will not be biased towards more frequent records. There are no duplicate records in the proposed test sets. So, the performance of learners is not biased. Number of selected records from each difficulty level group is inversely proportional to the percentage of records in the original KDD dataset. The number of train and test sets is reasonable which make it affordable to run the experiments on the complete set without need to randomly select a small portion. So, evaluation of results of different research works will be consistent and comparable. The training dataset is used to build up a model and the testing dataset is used to validate the model built.

VI. HARD WARE AND SOFTWARE

Net Beans IDE:

Net Beans IDE is an open source integrated development environment for application development environment on windows, Mac, Linux and Solaris operating systems. Netbeans IDE supports all java application types such as Java SE, Java ME, web and mobile applications). All the functions of the IDE are provided by modules. Each module provides a well-defined function such as support for the java language, editing or support for the CVS versioning system and SVN.

Net Beans contains all the modules needed for java development in a single download and allows the user to start working immediately. Modules also allow Net Beans to be extended, new features such as support for other programming languages can be added by installing additional modules. For instance, Sun Studio, Sun Java Studio Enterprise from Sun Microsystems are all based on the Net Beans IDE.

Java Technology:

For the software part, we used Java as a programming software since it is a high-level programming language with dynamic semantics. Java is an interpreted, object oriented programming language with a clear syntax and readability as well as it is supported by almost all the operating systems like Windows, Linux, Solaris and Mac OS. Java is portable, distributed, multithreaded, robust, secure and architectural neutral language.

The Java Programming language is unusual in that a program is both compiled and interpreted. With the compiler, first translate into an intermediate language called the Java bytes code where the platform independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once and the interpretation occurs each time the program is executed.

Java platform has two components. They are Java Virtual Machine (Java VM) and Java Application Programming Interface (Java API). Java Virtual Machine is the base for the Java platform and is ported onto the various hardware-based platforms. Java Application Programming Interface is a large collection of ready-made software components that provide many useful capabilities such as Graphical User Interface widgets. The Java API is grouped into libraries of related classes and the interfaces, these libraries are known as packages.

VII. CONCLUSION

Intrusion Detection System uses Supervised Learning in Quest (SLIQ) Algorithm for detecting the attacks. This system is capable of detecting the attacks and prevents the manipulations of data which is done by the intruders. It is developed to detect the intrusion in a system at the firewall or at the router. If any malicious data is found, then the total flow is rejected and requested again by the destination application Network Intrusion Detection System. Therefore, the detection and accuracy rate is higher than the existing system.

We can extend this application of Intrusion Detection System in a situation where the packets in the network flow are to be inspected to check whether they are malicious or normal. NSL KDD dataset is used to study the effectiveness of the classification method. This dataset consists of network flows which are split into the test dataset and train dataset. Accuracy and Time metrics are considered as performance metrics for the proposed method.

VIII. FUTURE SCOPE

In future, it is possible to provide extensions or modifications to the proposed classification algorithms using intelligent agents to archive further increased performance. These experiments and their results provide reliable guidelines for future research in applying supervised classifiers for field of intrusion detection and expose some new avenues of research. Many improvements can be added to the intrusion detection system developed in this thesis.

REFERENCES

[1] Elder J. F. and King M. A. Evaluation of Fourteen Desktop Data Mining Tools, in proceedings of the IEEE International Conference on Systems, Man and Cybernatics,1998.

[2] Juhua Chen, Wei Peng and Halping Zhou, An Implementation of ID3: Decision Tree Learning Algorithm Project of Comp 9417: Machine Learning University of New South Wales, School of Computer Science and Engineering, Sydney, NSW 2032.

[3] C4.5 Algorithm, Wikipedia, The Free Encyclopedia, Wikipedia Foundation, 28-Jan-15

[4] Breiman L., Random forests, Mach. Learn., vol. 45, no.1, pp.532, 2001

[5] Random Tree, Wikipedia, The Free Encyclopedia, Wikipedia Foundation, 13-Jul-14.

[6] Charles J. Stone, Jerome H. Friedman, Leo Breiman and Richard A. Olshen Classification and Regression Trees. Wadsworth International Group, Belmont, California, 1984.

[7] Gordan. V. Kass, An Exploratory Technique for investigation large quantities of categorical data Applied Statics, vol 29, No.2, pp. 119-127.1980.

[8] Manish Mehta, Rakesh Agarwal and Jorma Rissanen, SLIQ; A Scalable Parallel Classifier for Data Mining IBM Almaden Research Center, CA 95120.

[9] L. Hall, N. Chawla, and K. Bowyer, Combining Decision Trees Learned Parallel, Working Notes of the KDD-97 Workshop on Distributed Data Mining, pp. 1015, 1998.

[10] A. Andrzejak, F. Langer and S. Zabala, Interpretable models from distributed data via merging of decision trees, 2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), Apr. 2013.

[11] Lin, Shih-Wei et al, “An Intelligent algorithm with feature selection and decision rules applied to Anomaly Intrusion Detection “ Applied Soft Computing, vol. 12, no. IO pp. 3285-3290, 2012.

[12] Travallace, Mahbod, et ai, “A Detailed Analysis of the KDD CUP ((dataset”, proceedings of the second IEEE Symposium on Computation Intelligence for Security and Defence Applications 2009.

[13] Parsazad, Shafigh, Ehsan Saboori, Amin Allahyar,” Fast Feature Reduction in Intrusion Detection Datasets”, MIPRO, 2012 Proceedings of the International Convention, IEEE, 2012.

[14] Navaz, AS Syed, V. Sangeetha and C. Prabhadevi,” Entropy Based Anomaly Detection System to prevent DDOS attacks in cloud,” International Journal of Computer Applications (0975-8887) vol. 62, no. IS, 2013.

[15] F. Provost and D. Hennessy, Scaling up: Distributed Machine Learning with cooperation, in Proceedings of the National Conference on Artificial Intelligence, pp. 7479, 1996.