Detection of Cyber Attack Using Support Vector Machine and Directed Acyclic Graph

Prerna Sharma
M.Tech Scholar, Department of CSE
PCST, Bhopal INDIA
E-mail- teena.prerna@gmail.com

Saurabh Mandloi
Assistant Professor
Department of CSE, PCST, Bhopal
E-mail:- saurabh.mandloi@patelcollege.com

ABSTRACT

The current decade of internet based communication faced a problem of cyber attack. Cyber attack disclosure the credential of information. For the minimization of cyber attack used various algorithms for the minimization of attack possibility. In this paper proposed feature based classification technique for the processing of cyber attack categorization. The proposed technique used support vector machine and graph based technique for classification process. Our empirical evaluation shows that better result in compression of pervious method.

Keyword: - Firewall, IDS, Neural Network, Data Mining, Worms.

INTRODUCTION

Malware includes viruses, worms, Trojan horses, spy-ware, and adware. A virus is a computer program that attaches itself to a host (e.g., a program file or a hard disk boot record) and spreads when the infected host is moved to a different computer. A worm is a computer program that can replicate itself and spread across a network. A Trojan horse appears to be a legitimate computer program but has malicious code hiding inside which runs when activated. Spy-ware is malware that collects and sends data copied from the victim's computer, such as financial data, personal data, passwords, etc [7]. Adware, or advertising- supported software, is a computer program that automatically displays ads. Soft computing embraces several computational intelligence methodologies, including artificial neural networks, fuzzy logic, evolutionary computation, probabilistic computing, and recently it is extended towards artificial immune systems, belief networks, etc. These members neither are independent of one another nor compete with one another. Rather, they work in a cooperative and complementary way. There are various soft computing and machine learning techniques which are used in malware detection. Malware is a program that has malicious intention [12]. Whereas has defined it as a generic term that encompasses viruses, Trojans, spywares and other intrusive codes. Malware is not a "bug" or a defect in a legitimate software program, even if it has destructive consequences. The malware implies malice of forethought by malware inventor and its intention is to disrupt or damage a system.

WORMS

Worms are malicious software applications designed to spread via computer net- works. There are two types of worms: scanning worms and email worms.

Scanning worms exploit a software vulnerability to gain access/control of an end-host and require no human intervention to propagate. An infected end-host scans (dispatches suitably crafted packets often to randomly chosen IPv4 addresses of) potential victim end-systems. If the scanned end-system is susceptible to the exploit, it is subsequently infected and begins scanning (spreading the worm) in turn [3].

Email worms are installed when an end-host user inadvertently opens an email attachment containing malicious executables/scripts. Once installed, email worms harvest for email addresses from the infected host, craft new emails, attach the executables/scripts to the email and sends it.

WORM DETECTION TECHNIQUES

We now examine each worm detection technique individually, as they are applied in various detection systems. After describing each technique, we briefly analyze its strengths and weaknesses towards worm detection. There have been a variety of worm detection system proposed, using a wide range of techniques. We make the distinction here between a detection system, a relatively complete structure for detecting a worm which is typically the subject of one or more research publications; and a detection technique, which is a specific low-level means of detecting one aspect of a worm. Worm detection systems typically employ multiple techniques [9]. Looking directly at the techniques allows us to consider their strengths and weaknesses beyond the constraints of the system they are implemented in. To examine worm detection techniques, we first broadly categorize the detection techniques into one of four categories: host-based, honey-pot based, content-based, or behavior-based.

Host-based: Host based detection is characterized by the fact that it uses information only available at the end-host. It must be installed on each host that is to be protected by it. Modifications may be required to the operating system or the software that runs on it to give the detection software access to the internals of the execution environment. Host-based techniques include: buffer overflow detection, correlating network data to memory errors, and looking for patterns in system calls.

Honey-pot based: Honey-pot based worm detection is closely related to host-based detection, but differs in that host-based detection is deployed to live servers whereas honey-pot by design serve no function beyond worm detection. All host-based worm detection methods could be deployed to the software running on a honey-pot, but this is not generally necessary as all connections to a honey-pot are already considered to be suspicious.

Content-based: Content-based systems observe the contents of network traffic looking for byte patterns that match the signature of a worm. The signatures are either generated on the fly by the worm detector, or developed manually from deconstruction of a worm instance. They rely on the fact that some aspect of the data that is sent to take advantage of vulnerability is never sent as part of legitimate traffic, and can therefore be used to accurately identify worm traffic. Content-based techniques include static signatures, dynamic signatures, and advanced signatures.

Behavior-based: In contrast to content-based worm detection techniques, behavior-based worm detection attempts to detect the presence of a worm by monitoring the network without examining the payload of transmitted packets. Instead, these techniques rely solely on patterns of network activity that are characteristic of worm-specific behavior, such as the aggressive scanning a worm might rely on in searching for vulnerable targets. Like content-based systems, behavior-based systems are typically easily deployable as they are often amenable to being installed at a network gateway. They offer an additional advantage in that they are robust against content polymorphism, thus potentially providing better overall coverage. On the other hand, behavior-based systems typically produce less information gain. They can detect the presence of the worm, and may be able to identify which host is infected, but they typically cannot generate a signature that could be used to block individual worm connections.

Section-II gives the information about Data mining approach. In section III discuss the proposed method. In section IV discuss comparative result finally, in section-V conclusion and future scope.

II ARTIFICIAL NEURAL NETWORKS

Artificial neural network is an information processing model that is inspired by the biological nervous systems, such as brain, process information. It tries to represent the physical brain and thinking process through electronic circuit or software. Artificial neural network is the network of individual neurons. Each neuron is a neural network 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, neural networks also learn by example or training, they cannot define or program to perform a specific task. Neural networks perform very successfully for recognizing and matching complicated or incomplete patterns. The most successful application of neural network is classification or categorization and pattern recognition. The learning process is essentially an optimization process in which the parameters of the best set of connection coefficients (weighs) for solving a problem are found and includes the following basic steps:-

- ❖ Present the neural network with a number of inputs.
- ❖ Check how closely the actual output generated for a specific input matches the desired output.
- ❖ Change the neural network parameters to better approximate the outputs.

There are two different learning methods for the neural networks: supervised and unsupervised. In supervised learning method, the network learns the desired output for a given input or pattern. The well known architecture of supervised neural network is the Multi-Level Perceptron (MLP); the MLP is employed for Pattern Recognition problems. On the other hand, in unsupervised learning method, the network learns without specifying desired output. Self-Organizing Maps (SOM) are popular unsupervised training algorithms; a SOM tries to find a topological mapping from the input space to clusters.

The main merit of the artificial neural networks includes the ability of faster information processing, the ability of classification and the ability of learning and self-organization. Because of these abilities of the artificial neural networks, the network intrusion detection system can analyze the network captured packets and detect whether it would be an intrusion or not.

HIDDEN MARKOV MODELS

Hidden Markov models (HMMs) are well suited for statistical pattern analysis. Since their initial application to speech recognition problems in the early 1970’s, HMMs have been applied to many other areas including biological sequence analysis. An HMM is a state machine where the transitions between states have fixed probabilities. Each state in an HMM is associated with a probability distribution for observing a set of observation symbols. We can “train” an HMM to represent a set of data, which is usually in the form of observation sequences. The states in the trained HMM then represent the features of the input data, while the transition and the observation probabilities represent the statistical properties of these features. Given any observation sequence, we can match it against a trained HMM to determine the probability of seeing such a sequence. The probability will be high if the sequence is “similar” to the training sequences. In protein modelling, HMMs are used to model a given family of proteins. The states correspond to the sequence of positions in space while the observations correspond to the probability distribution of the 20 amino acids that can occur in each position. A model for a protein family assigns high probabilities to sequences belonging to that family. A trained HMM can then be used to discriminate family members from non-members. Metamorphic viruses form families of viruses. Even though members in the same family mutate and change their appearances, some similarities must exist for the variants to maintain the same functionality. Detecting virus variants thus reduces to finding ways to detect these similarities. Hidden Markov models provide a means to describe sequence variations statistically. We propose to use HMMs similar to those used in protein sequence analysis to model virus families [13]. In virus modelling, the states correspond to the features of the virus code, while the observations are instructions or op codes making up the program. A trained model should then be able to assign high probabilities to and thus identify viruses belonging to the same family as the viruses in the training set.

DATA MINING APPROACH

Data mining methods are often used to detect patterns in a large set of data. These patterns are then used to identify future instances in a similar type of data. The experimented with a number of data mining techniques to identify new malicious binaries. Here three learning algorithms to train a set of classifiers on some publicly available malicious and benign executables. They compared their algorithms to a traditional signature-based method and reported a higher detection rate for each of their algorithms. However, their algorithms also resulted in higher false positive rates when compared to signature-based method. The key to any data mining framework is the extraction of features, which are properties extracted from examples in the dataset. Schultz et al. extracted some static properties of the binaries as features. These include system resource information (the list of DLLs, the list of DLL function calls, and the number of different function calls within each DLL) obtained from the program header, and consecutive printable characters found in the files [21]. The most informative feature they used was byte sequences, which were short sequences of machine code instructions generated by the hex dump tool. The features were used in three different training algorithms. There was an inductive rule-based learner that generated Boolean rules to learn what a malicious executable was; a probabilistic method that applied Bayes rule to compute the likelihood of a particular program being malicious, given its set of features; and a multi-classifier system that combined the output of other classifiers to give the most likely prediction.

III PROPOSED WORK

Malware classification and detection process is very complex process in network security. In current network security scenario various types of malware family are available some are known family and some are unknown family. The family of know malware detection used some well know technique such as signature based technique and rule based technique. in case of unknown malware family of attack detection is various challenging task. In current trend of malware detection used some data mining technique such as classification and clustering. The process of classification improves the process of detection of malware. The continuity of chapter discusses feature extraction process of malware data, directed acyclic graph technique, support vector machine and proposed methodology.

Step1: Initially input Malware data passes through preprocessing function and extracted feature part of Malware data in form of traffic type.

Step2: the extracted traffic feature data converted into feature vector.

Step 3: In phase of feature mapping in feature space of DAG create a fixed class according to the group of data.

Step 4: steps of processing of DAG.

1. Initialize Gaussian hyper plane margin.
2. Choose a random vector from training data and present it to the DAG.
3. The weight of the plane support vector is estimated. The size of the vector decreases with each iteration.
4. Each vector in the SV’s neighborhood has its weights adjusted to become more like the SV. Vector closest to the SV are altered more than the vector furthest away in the neighborhood.
5. Repeat from step 2 for enough iteration for convergence.
6. Calculating the SV is done according to the Euclidean distance among the node’s weights (W_1, W_2, … , W_n) and the input vector’s values (V_1, V_2, … , V_n).
7. The new weight for a node is the old weight, plus a fraction (L) of the difference between the old weight and the input vector… adjusted (theta) based on distance from the SV.

Step 5: After processing of support vector finally malware data are classified.

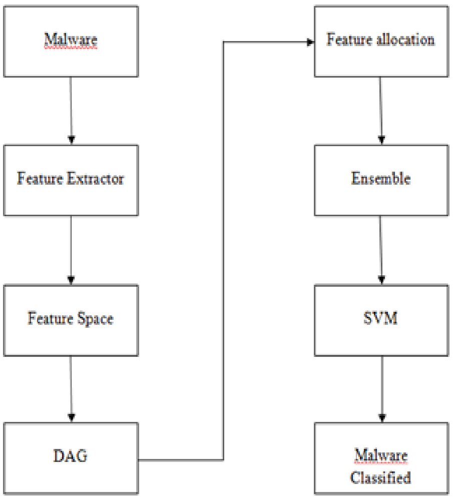
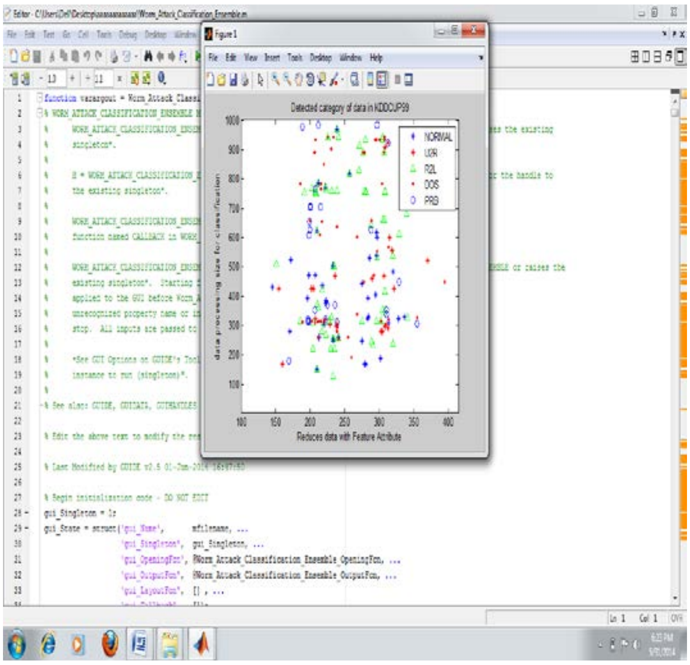


Figure 1: Proposed Model for Malware Classification.

IV EXPERIMENTAL RESULT AND ANALYSIS

In this paper, we perform the experimental process of proposed improved ensemble for Malware detection. The proposed method is implemented in Matlab 7.14.0 and tested with very reputed dataset from the UCI machine learning research center. In the research work, I have measured detection accuracy, true positive rate, false positive rate, true negative rate and finally the false negative rate error of the classification ensemble method. To evaluate these performance parameters I have used KDDCUP99 datasets from the UCI machine learning repository namely Worm detection dataset.


Editor - C:\Users\Dell\Desktop\aaaaaaaaaaa\Worm_Attack_Classification_Ensemble.m
Figure 1
File Edit View Insert Tools Desktop Window Help
Detected category of data in KDDCUP99
NORMAL
U2R
R2L
DOS
PRB
data processing size for classification
Reduces data with Feature Attribute
100
150
200
250
300
350
400
1000
900
800
700
600
500
400
300
200
100
Ln 1 Col 1 OVR

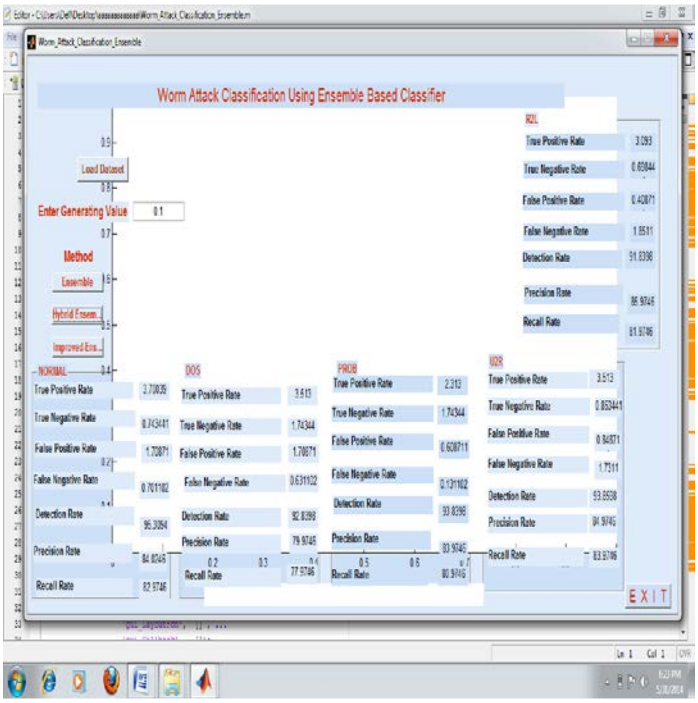
```
function varargout = Worm_Attack_Classi
% WORM_ATTACK_CLASSIFICATION_ENSEMBLE M
%      WORM_ATTACK_CLASSIFICATION_ENSEM                          es the existing
%      singleton*.
%
%      H = WORM_ATTACK_CLASSIFICATION_E                          r the handle to
%      the existing singleton*.
%
%      WORM_ATTACK_CLASSIFICATION_ENSEM
%      function named CALLBACK in WORM_
%
%      WORM_ATTACK_CLASSIFICATION_ENSEM                          EMBLE or raises the
%      existing singleton*.  Starting f
%      applied to the GUI before Worm_A
%      unrecognized property name or in
%      stop.  All inputs are passed to
%
%      *See GUI Options on GUIDE's Tool
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the res

% Last Modified by GUIDE v2.5 01-Jun-2014 16:07:50

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Worm_Attack_Classification_Ensemble_OpeningFcn, ...
                   'gui_OutputFcn',  @Worm_Attack_Classification_Ensemble_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
```

Figure 2: Shows that the window for detection the Worm with Hybrid ensemble method with using the generating value is 0.1.


Editor - C:\Users\Dell\Desktop\aaaaaaaaaaa\Worm_Attack_Classification_Ensemble.m
Worm_Attack_Classification_Ensemble
Load Dataset
Method
Ensemble
Hybrid Ensem...
Improved Ens...
EXIT
0.9
0.8
0.7
0.6
0.5
0.4
0.2
0.3
0.5
0.6
Ln 1 Col 1 OVR

Worm Attack Classification Using Ensemble Based Classifier

Enter Generating Value 0.1

R2L
True Positive Rate 3.093
True Negative Rate 0.65844
False Positive Rate 0.40871
False Negative Rate 1.8511
Detection Rate 91.8398
Precision Rate 86.9746
Recall Rate 81.9746

NORMAL
True Positive Rate 3.70039
True Negative Rate 0.743441
False Positive Rate 1.70871
False Negative Rate 0.701102
Detection Rate 95.3094
Precision Rate 84.8246
Recall Rate 82.9746

DOS
True Positive Rate 3.513
True Negative Rate 1.74344
False Positive Rate 1.70871
False Negative Rate 0.631102
Detection Rate 92.8398
Precision Rate 79.9746
Recall Rate 77.9746

PROB
True Positive Rate 2.313
True Negative Rate 1.74344
False Positive Rate 0.608711
False Negative Rate 0.131102
Detection Rate 90.8398
Precision Rate 83.9745
Recall Rate 80.9746

U2R
True Positive Rate 3.513
True Negative Rate 0.853441
False Positive Rate 0.84871
False Negative Rate 1.7311
Detection Rate 93.8598
Precision Rate 84.9746
Recall Rate 83.9746

Figure 3: Shows that the windows for detection the Worm with Hybrid ensemble method with using the generating value is 0.1, and find the parameters value FPR, FNR, TPR, TNR, Recall, precision and detection rate.

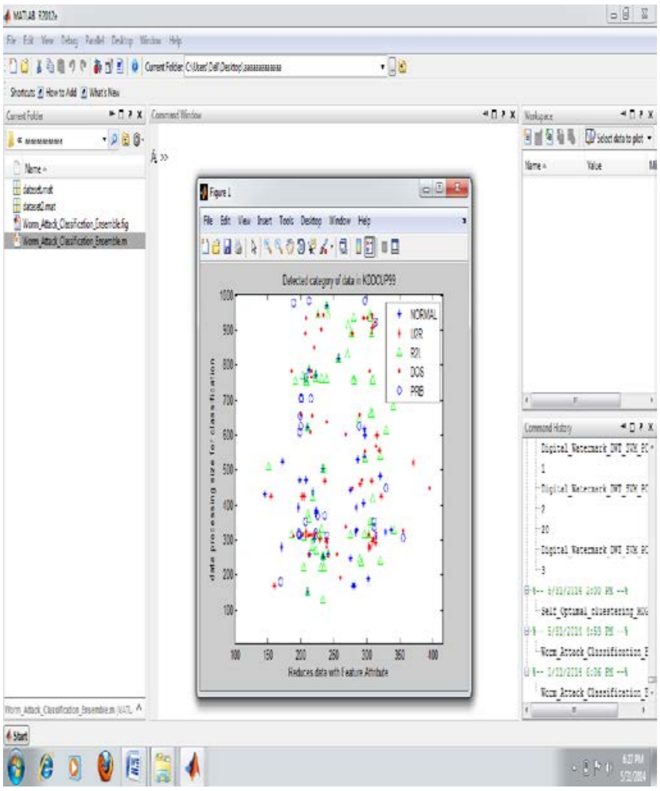

MATLAB R2012a
File Edit View Debug Parallel Desktop Window Help
Current Folder: C:\Users\Dell\Desktop\aaaaaaaaaaa
Shortcuts How to Add What's New
Current Folder
Command Window
Workspace
Name
Value
dataset.mat
dataset2.mat
Worm_Attack_Classification_Ensemble.fig
Worm_Attack_Classification_Ensemble.m
Figure 1
File Edit View Insert Tools Desktop Window Help
Detected category of data in KDDCUP99
NORMAL
U2R
R2L
DOS
PRB
data processing size for classification
Reduces data with Feature Attribute
100
150
200
250
300
350
400
1000
900
800
700
600
500
400
300
200
100
Command History
Start

Figure 4: Shows that the window for detection the Worm with improved ensemble method with using the generating value is 0.1.

Method Name	Value	TYPES OF ATTACK	TPR	TNR	FPR	FNR	DETECTION RATE	PRECISION RATE	RECALL RATE
ENSEMBLE	0.1	NORMAL	4.273	0.703	1.568	0.691	89.79	81.93	80.93
		DOS	4.373	0.296	0.568	0.308	88.79	80.83	78.46
		PROBE	4.483	1.703	0.191	0.351	86.79	79.56	81.93
		U2R	5.273	0.407	1.431	0.308	85.79	82.93	81.93
		R2L	3.473	1.592	0.568	0.168	86.79	84.43	79.93

Table 1: Shows that the performance evaluation of TPR, TNR, FPR, FNR, Detection rate, Precision rate and Recall rate for Ensemble method, and the input value is 0.1.

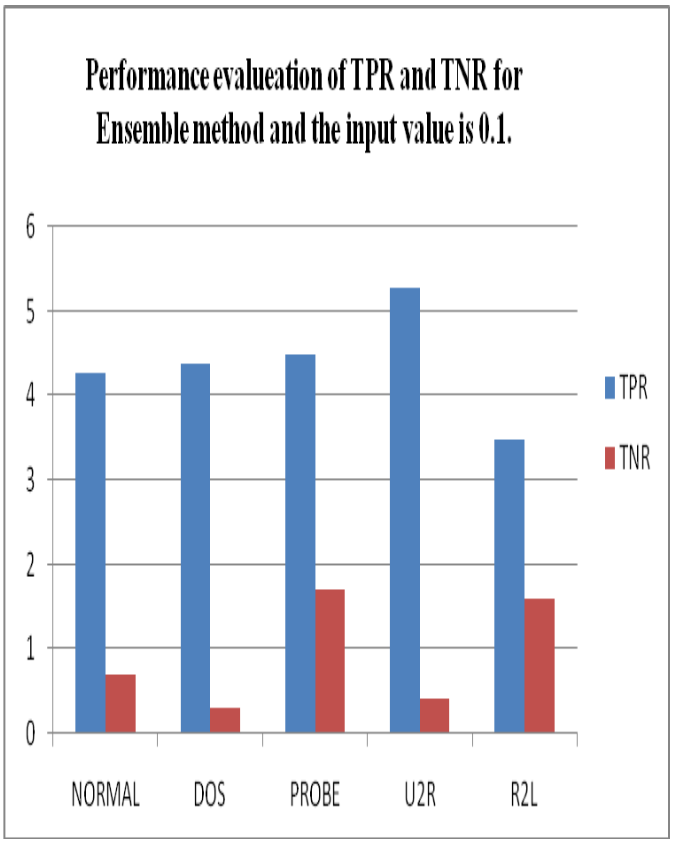


Figure 5: Shows that the performance evalueation of TPR and TNR for the ensemble method and the input value is 0.1.

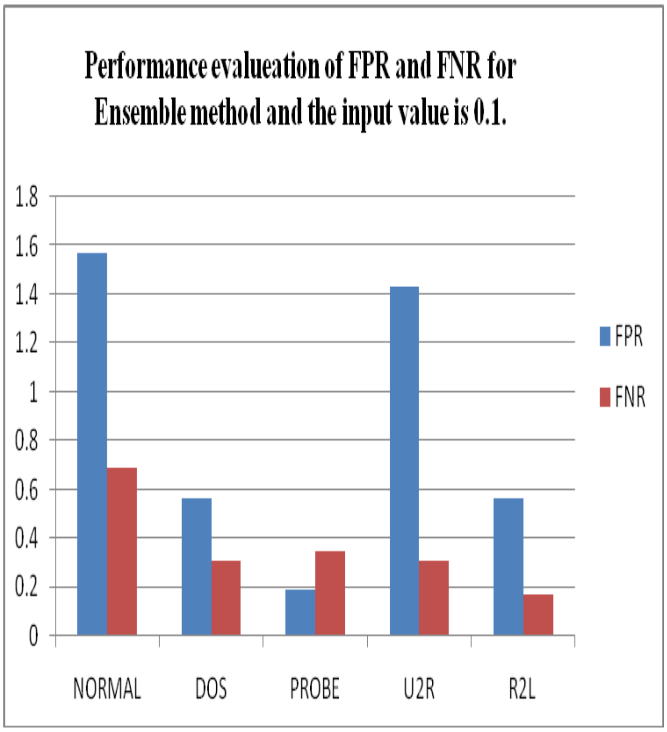


Figure 6: Shows that the performance evalueation of FPR and FNR for the ensemble method and the input value is 0.1.

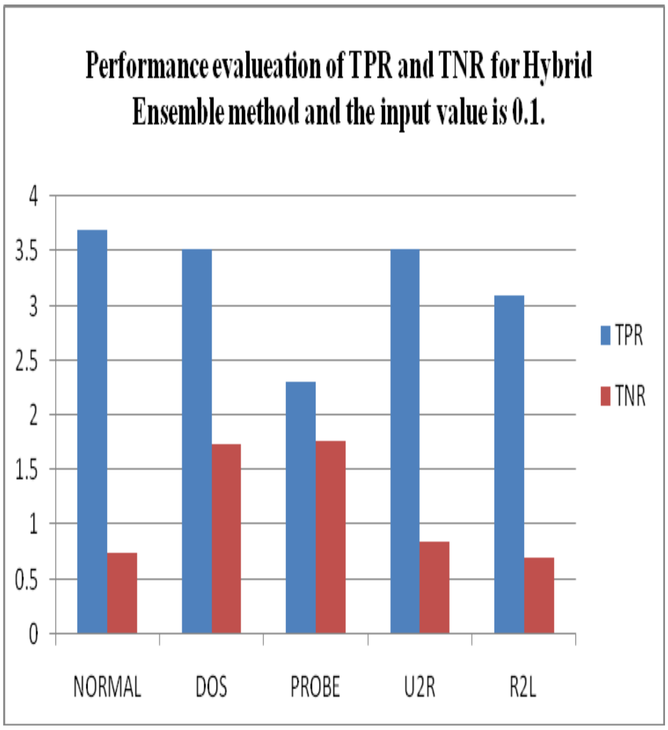


Figure 7: Shows that the performance evalueation of TPR and TNR for the Hybrid ensemble method and the input value is 0.1.

V CONCLUSION AND FUTURE WORK

In this paper, we have proposed a novel hybrid method, based on DAG and Gaussian Support Vector Machines, for malware classification. Experiments with the KDD Cup 1999 Data show that SVM-DAG can provide good generalization ability and effectively classified malware data. Moreover, the modified algorithms proposed in this desecration outperform conventional CIMDS and ISMCS in terms of precision and recall. Specifically, accuracy of the modified algorithms can be increase due to feature allocation of DAG, and reduces feature sub set increase the accuracy of classification. From our experiments, the DAG-SVM can detect known attack types with high accuracy and low false positive rate which is less than 1%.

The proposed method classified attack and normal data of KDDCUP99 is very accurately. The proposed method work in process of making group of attack very accurately, the learning process SVM training process makes very efficient classification rate of Malware data. Our empirical result shows better performance in compression of ISMCS and another data mining technique for malware detection.

In this paper we proposed a hybrid method for Malware classification. Our experimental result shows that better result in compression of old and traditional method of malware classification. But the computational time of process is increase. In future we reduce the iteration process of DAG-SVM neural network for Speed classification and detection of Malware. In the collection of feature attribute of malware data, some suspicious features are not collected, so in future used heuristic function for better selection of features.

REFERENCES:-

[1] Tawfeeq S. Barhoom, Hanaa A. Qeshta "Adaptive Worm Detection Model Based on Multi classifiers" 2013 Palestinian International Conference on Information and Communication Technology, IEEE 2013. Pp 58-67.

[2] Ibrahim Aljarah, Simone A. Ludwig "Map Reduce Intrusion Detection System based on a Particle Swarm Optimization Clustering Algorithm" IEEE Congress on Evolutionary Computation, 2013. Pp 955-963.

[3] Kai Huang, Yanfang Ye, Qinshan Jiang "ISMCS: An Intelligent Instruction Sequence based Malware Categorization System" IEEE 2010. Pp 658-662.

[4] Jonghoon Kwon, Heejo Lee "Bin Graph: Discovering Mutant Malware using Hierarchical Semantic Signatures" IEEE, 2012. Pp 104-112.

[5] P.R.Lakshmi Eswari, N.Sarat Chandra Babu "A Practical Business Security Framework to Combat Malware Threat" World Congress on Internet Security, IEEE 2012. Pp 77-81.

[6] Ahmed F.Shosha, Chen-Ching Liu, Pavel Gladyshev, Marcus Matten "Evasion-Resistant Malware Signature Based on Profiling Kernel Data Structure Objects" 7th International Conference on Risks and Security of Internet and Systems, 2012. Pp 451-459.

[7] Hira Agrawal, Lisa Bahler, Josephine Micallef, Shane Snyder, Alexandr Virodov "Detection of Global, Metamorphic Malware Variants Using Control and Data Flow Analysis" IEEE, 2013. Pp 1-6.

[8] Vinod P., V.Laxmi, M.S.Gaur, Grijesh Chauhan "MOMENTUM: MetamOrphic Malware Exploration Techniques Using MSA signatures" International Conference on Innovations in Information Technology, IEEE 2012. Pp 232-238.

[9] Robiah Y, Siti Rahayu S., Mohd Zaki M, Shahrin S., Faizal M. A., Marliza R. "A New Generic Taxonomy on Hybrid Malware Detection Technique" International Journal of Computer Science and Information Security, Vol-5, 2009. Pp 56-61.

[10] anfang Ye, Tao Li, Qingshan Jiang, Youyu Wang "CIMDS: Adapting Postprocessing Techniques of Associative Classification for Malware Detection" IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, IEEE Vol-40, 2010. Pp 298-307.

[11] Raman Singh, Harish Kumar, R.K. Singla "Review of Soft Computing in Malware Detection" IJCA, 2013. Pp 55-60.

[12] Mihai Christodorescu, Somesh Jha, Sanjit A. Seshia, Dawn Song, Randal E. Bryant "Semantics-Aware Malware Detection"

[13] Sarnsuwan N.; Wattanapongsakorn N.; and Charnsripinyo Ch."A New Approach for Internet Worm Detection and Classification" etworked Computing (INC), 6th International Conference, 2010. Pp 546-552.

[14] Wang X.; Yu W.; Champion A.; Fu X.; and Xuan D "Detecting Worms via Mining Dynamic Program Execution" Authorized licensed use limited to: The Ohio State University, 2008. Pp 696-702.

[15] Z. Gao, T. Li, J. Zhang, C. Zhao, and Z. Wang "A parallel method for unpacking original high speed rail data based on map reduce" Springer Berlin Heidelberg, vol. 124, 2012. Pp 59–68.

[16] W. Zhu, N. Zeng, and N. Wang "Sensitivity, specificity, accuracy associated confidence interval and roc analysis with practical SAS implementations" in In Proceedings of the NorthEast SAS Users Group Conference NESUG10, 2010.

[17] I. Aljarah and S. A. Ludwig "Parallel particle swarm optimization clustering algorithm based on map reduce methodology" in Proceedings of the Fourth World Congress on Nature and Biologically Inspired Computing (NaBIC'12), Mexico City, Mexico, November 2012, Pp 104–111.

[18] J. Mazel, P. Casas, Y. Labit, and P. Owezarski "Subspace clustering, inter-clustering results association & anomaly correlation for unsupervised network anomaly detection" in Proceedings of the 7th International Conference on Network and Services Management, Paris, France, 2011, Pp 73–80.

[19] Z. Li, Y. Li, and L. Xu "Anomaly intrusion detection method based on k-means clustering algorithm with particle swarm optimization" in Proceedings of the 2011 International Conference of Information Technology, Computer Engineering and Management Sciences. Washington, DC, USA: IEEE Computer Society, 2011, Pp 157–161.

[20] Y. Ye, D.Wang, T. Li, and D. Ye "IMDS: Intelligent malware detection system" In Proccedings of ACM International conference on Knowledge Discovery and Data Mining, 2007, Pp 1043-1047.

[21] Y. Ye, D.Wang, T. Li, D. Ye and Q. Jiang "An intelligent PE malware detection system based on association mining" Journal in Computer Viorology, 2008. Pp323-334.

[22] L.Jing, M.K.Ng, J.Z.Huang "An Entropy Weighting k-Means Algorithm for Subspace Clustering of High-Dimensional Sparse Data " IEEE Transactions on Knowledge and Data Engineering, 2007, Pp 1-16.

[23] P. Ferguson "Observations on emerging thrests" in USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET), Apr. 2012.

[24] K. Thomas and D. Nicol, “The koobface botnet and the rise of social malware” in IEEE Int. Conf. Malicious and Unwanted Software (Malware 10), 2010, Pp 63–70.

[25] L. Martignoni, E. Stinson, M. Fredrikson, S. Jha, J. Mitchell “A layered architecture for detecting malicious behaviors” in Symposiumon Recent Advances in Intrusion Detection (RAID), 2008.