ISSN:2320-0790

An International Journal of Advanced Computer Technology

Malware Detection with Different Voting Schemes

Ms. Jyoti Landage, Prof. M.P.Wankhade

Department of Computer Engineering, Sinhgad College of Engineering, Pune, Maharashtra, India

Abstract: A common way of launching the attack in computer system is Malware. It has malicious intent of performing any kind of malicious action to computer system as a result entire system crashes. It comes in different forms like virus, Trojan, Spyware, Scareware, Adware etc. Traditional malware detection techniques viz. signature-based, Heuristic-based and Specification-based detection technique are unable to detect some form of malware and each technique has its own advantages and disadvantages.

A new methodology is proposed for malware detection that is based on data mining and machine learning techniques to detect known as well as unknown instances of malware. The new methodology uses disassemble process and three pre-processing techniques as part of feature extraction process to produce three different data sets with different configurations; feature selection technique is used to achieve consistent, reduced feature dataset. Three classification algorithms are used to generate and train the classifiers named as Ripper, C4.5 and IBk. The ensemble learning algorithm voting is used to improve the accuracy of result. Here majority voting and veto voting is used, the predicted output is decided on the basis of majority voting and veto voting. In veto voting the decision strategy of veto is improved by introducing the trust-based veto voting that definitely helps to improve the detection accuracy.

Keywords: Data mining, Ensemble, Feature Extraction, Feature selection, Machine learning, malware detection, Majority voting, Trust, Veto Voting.

I. INTRODUCTION

In today's world security is major issue in every field of technology. Information security, network security, computer security all are branches of information technology which deal with protection of information on a network or standalone computer. As every organization depends on the computer and technology of security requires constant development. A more recent annual report on the Internet security threat-2013 from Symantec says "Threats to online security have grown and evolved considerably in 2012, In particular, social media and mobile devices have come under increasing attack in 2012."[1]

Malware is a general term for all types of malicious software, which in the context of computer security means: software which is used with the aim of attempting to breach the computer systems security policy with respect to confidentiality, integrity and availability [2]. The main characteristics of the malware are replication, propagation, self–execution and corruption of computer system. It spread over the connected system in the network or internet connection. It infects the system by transferring malware from a polluted device to another uninfected one using local or network file system [3]. Malwares are classified according to their propagation method and they come in the different forms like Virus, worms, Trojan horse, Spyware, scareware, adware, Backdoors, Botnets etc. Malware detection is the key to protect the system from these types of malware. There are two main traditional malware detection techniques: Signature-based detection and Heuristic-based detection. In signature-based technique specific features or unique strings are extracted from binaries, which are later used for detection of malware. However a copy of malware is required to extract and develop a signature for detection purposes. A database of known code signature is updated and refreshed constantly by anti-virus software vendor as a result it detects only known instances of malware accurately. It cannot detect the new, unknown malware as no signature is available in database for such types of malware. Heuristic-based technique detects known as well as unknown instances of malware but level of false positive is high i.e. accuracy is low and it is more time and resource consuming technique, therefore a new malware detection technique named as data mining based detection is proposed.

The aim of this study is to investigate malware detection and enhance the idea of heuristic-based detection method by using machine learning algorithm and data mining technique. The purpose is to detect both known and unknown instances of malware with high accuracy. The proposed system framework consists of data pre-processing techniques, ensemble learning algorithm and evaluation of proposed algorithm for malware detection. The dataset of

benign and malicious executable files is created. These executable files are disassembled to get the sequence of opcode as feature which is part of assembly language instruction. The opcodes are extracted using three pre-processing techniques as part of feature extraction process to produce three different dataset with different configuration. Feature selection method is used to select consistent, relevant features and remove the redundant, irrelevant data. Three classification algorithms Ripper, decision tree C4.5 and K-nearest neighbour are used to generate and train the classifiers. The ensemble learning algorithm voting is used to improve the result accuracy. The output of multiple classifiers is combined and decision is taken on the basis of majority voting, but decision taken by majority voting can be wrong if majority of classifiers classify malware instance as benign. The veto voting is used to overcome the drawback majority voting. The output of multiple classifiers is combined and final prediction is given on the basis of veto voting. The decision strategy of veto is improved by introducing the trust-based veto voting. The results of majority voting, veto voting and trust-based veto voting are compared to determine which voting scheme is best that definitely helps to improve the detection accuracy.

The rest of the paper is organized as follows; section 2 explains the related work in malware detection using data mining and machine learning methods, section 3 explains the proposed system architecture & their modules and section 4 conclude the paper.

II. RELATED WORK

In 2001 Schultz first introduced the idea of applying the data mining and machine learning methods for the detection of malware based on their respective binary codes. Here three different features are extracted using three different techniques. Program header is extracted using libBFD method; strings are extracted using GNU string method, byte sequence (binary n-gram) is extracted by using HEXDUMP command. Three different classification algorithms are used to generate classifiers using different features: Ripper algorithm uses program header information as a feature, naïve bayes uses string, multi-naïve bayes uses the byte sequence as feature and he compared these methods with signature-based method, and the found that detection rate of signature-based method is lower than data mining-based detection method. Highest overall accuracy was achieved by the naïve bayes algorithm with string and highest detection rate was achieved by multi-naïve bayes using byte sequence as feature [4].

In 2010 Yi-Bin Lu et al improved the accuracy of malware detection using the classifier ensembles to replace individual classifier. The combination of multiple classifiers to reach final prediction is called ensemble. Ensemble model performs better than single classifier model; He introduced the different ensemble learning algorithm like bagging, boosting, voting, stacking and grading. The new ensemble learning method SVM-AR was proposed, it combines the SVM and association rules based on hierarchical taxonomy, also proposed the framework for malware detection using machine learning. According to review of related papers on topic of malware detection using machine learning it was found that decision tree, SVM, NB and KNN are most common classification algorithms used by researchers. The overall accuracy of each algorithm was tested using collected dataset. The result showed that NB is the worst classification algorithm. Accuracy improvement is achieved using the multi-classifier as ensemble learning method [5].

In 2010 Raja Khurram Shazhad detected the spyware by using data mining and machine learning methods; he extracted the byte sequence as feature using XXD command which is Unix-based utility for generating the hexadecimal dumps of binary files. From these hexadecimal dumps byte sequences are extracted in term of n-gram of different size where n=4, 5, 6. Two techniques are used to select the consistent features: Common feature-based extraction (CFBE) and frequency-based feature extraction (FBFE) which uses different types of data representation. In CFBE the common n-grams are extracted and in FBFE number of occurrences of some specific n-grams are extracted from a certain class as a result two reduced feature sets are obtained. The classification algorithms ZeroR is used as baseline for comparison, NB, SMO, J48, random forest and Jrip are used. These algorithms are used to generate and train the classifiers that can classify unknown binaries by analysing extracted n-grams. The result showed that feature set generated by CFBE feature selection method generally produced better results with regard to accuracy than feature sets generated by FBFE feature selection method, this methods was successful even though the training data is limited [6]. In 2011 he proposed detection of adware by using same data mining and machine learning approach. Netwide command is used to disassemble the executable file. Opcodes are extracted as feature and it further processed as per selected n-gram size, where n=2,3..8and considered 4 and 5 as intermediary values, then reduced feature set is obtained using text categorization technique TF-IDF. This reduced feature set is again processed with categorical proportional differences (CPD) algorithm to obtain the final dataset. The KNN and SVM were effective when the data are noisy and KNN has an advantage that its performance is superior incrementally when new training samples are introduced [7].

In 2012 Asaf shabtai et al., detected unknown malicious instances by applying the various classification algorithms on opcode patterns i.e opcodes are used as a feature and Evaluated number of experiments & found the setting of opcode 2-gram also called as opcode bi-gram, TF, using 300 features selected by DF measure outperformed. The performance of decision tree & boosted decision tree was very well as compared to NB & boosted NB [8]. By referring different existing work, In 2008 Robiah Yusof et al., has proposed below criteria for malware detection techniques:

1. Capability to reduce false negative rate
2. Capability to reduce false positive rate

3. Capability to detect known attack
4. Capability to detect unknown attack
5. Capability to get accurate results i.e. to improve the accuracy [9]

The proposed malware detection system using data mining tries to fulfil above proposed criteria and successfully detects the known as well as unknown malwares with high accuracy.

III. PROPOSED WORK

The overall process of classifying the unknown files as either benign or malicious using machine learning method is divided into two phase: training phase and testing phase. In training phase training data set of malicious and non-malicious files are prepared. Each file is processed with feature extraction and selection techniques, as a result consistent, reduced feature data set is obtained. The vectors of files in the data set and their known classification are the input for learning algorithm. The learning algorithms process these vectors and generate the trained classifiers. The trained classifiers are used in the proposed classification model. During testing phase, test set collection of new, unknown benign and malicious files which did not appear in the training data set are classified by the classifier that was generated in the training phase. Each file in the test data set is pre-processed as in the training phase. Based on the vectors of files in the test data set the trained classifier will classify the file as either benign or malicious. In the testing phase the performance of the generated classifiers is evaluated by standard accuracy measures. The system architecture for the proposed work is shown in the fig. 1.

A. System Architecture

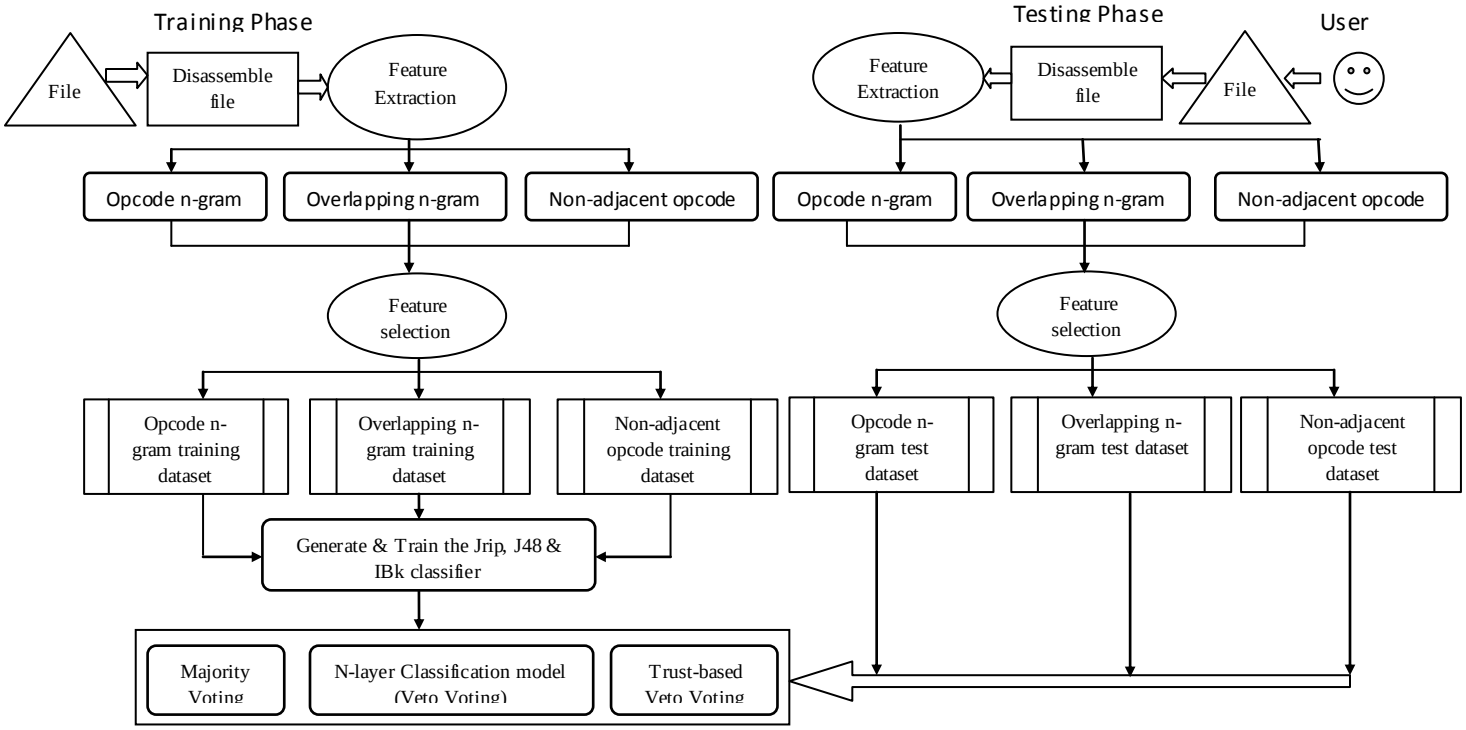

Fig. 1 System Architecture

B. System Module

The experiment that will be performed consists of four modules which are given below with their functionality.

1) Data Set: For malware classification, data sets have been prepared using various representations of files. Features that are commonly extracted from executable files include byte code n-gram, printable strings, instruction sequence, system calls, opcode n-gram. N-gram is sequence of n characters. Here opcode n-gram is used as feature. Executable file is disassembled into assembly language program file. An assembly instruction contains operation code (opcode) and maybe one or more operands for performing the operations. The opcodes are extracted as feature to prepare the dataset.

***2) Feature Extraction*:** Three feature extraction techniques are used to extract the opcode as feature and three correspondence datasets are prepared.

a) Opcode n-gram

Data set is prepared with size of n=2 i.e. opcode bi-gram. To understand this process, assume that a disassembled binary file contains the following given data. A pair of characters represents an opcode.

aa 11 bb 22 cc 33 dd 44 ee 55 ff

The generated bi-grams are aa11 bb22 cc33 dd44 ee55.

b) Overlapping n-gram

This technique is used to extract all possible combination of strings. Two parameters namely size and step are used. The Size parameter defines the size of n-gram to be extracted and step parameter defines the number of opcodes to be skipped before extracting the next n-gram. Following the above example, if size=2 and step =1, the generated string will be aa11 11bb bb22 22cc cc33 33dd dd44 44ee ee55 55ff.

c) Non-adjacent opcode extraction

Some changes are made in overlapping n-gram i.e. size parameter is changed to the start-end size parameter. The start-end size parameter defines the number of adjacent opcodes to be extracted for start and end of n-gram. The step parameter defines the number of opcodes to be skipped for extracting a new n-gram and Gap size parameter specifies the gap between start and end opcode or number of opcodes to be skipped between start and end opcode of n-gram. If start-end size= 1, step =1 and gap=1, the generated output will be aabb 1122 bbcc 2233, ccdd and so on.

These three feature extraction techniques are used to extract no. of possible combination of strings and prepared three correspondence datasets.

3) Feature Selection: It is necessary to remove the irrelevant, redundant, noisy data from the entire large dataset, so we need to select small, relevant, consistent features from the entire large feature set as a result reduced feature dataset will be achieved. Many techniques have been used to select best features like gain ratio, information gain, fisher score, term frequency-inverse document frequency (TF-IDF). In our work TF-IDF is used. It is a text categorization technique. N-gram is analogue to word or term in text document. A vocabulary of words or term is extracted from so called document set. For each word or term (t) in the vocabulary, its frequency (f) in the single document (d) and in the entire set i.e. document set (D) is calculated. Weight is assigned to each word; weight is equal to its frequency (f) in d. such weights are called as term frequency (tf) i.e. frequency of term in document. The frequency (F) of each term is calculated in D, this is called Document Frequency (DF). The normalized TERM FREQUENCY (TF) is calculated by dividing the frequency of term in document (tf) by the frequency of the most frequent term in document [max (tf)] within the range of [0-1] as shown in equation (1):

$$TF = \frac{tf}{\max(tf)} \quad (1)$$

The TF-IDF combines the TF and DF. The equation (2) of TF-IDF is given below:

$$TF - IDF = TF * \log\left(\frac{N}{DF}\right) \quad (2)$$

N: No. of document in the entire data set.

DF: No. of Document d in which term (t) appears.

The reduced feature data set is then converted into attribute relation file format (ARFF). This ARFF file is used as input to three classification algorithm namely Ripper, Decision Tree (C4.5) and K-nearest neighbour (IBk) to generate and train the classifiers. The trained classifiers are used in classification model; this stage is called as training stage.

4) Classification Model: Every classifier has its own decision. In proposed system they are used as committee in classification model. Here we used classifier ensemble which can use method like voting to reach the final prediction. It performs better than single classifier and helps to improve the detection accuracy. Three voting schemes are used viz. majority voting, veto voting and trust-based veto voting.

a) Majority Voting

The decision from more than one expert (Classifier) may be required in certain situation, so committee of experts (Ensemble) is formed as it is expected that a committee always performs better than a single expert. Normally committee uses majority voting for combining the decisions of the experts to reach a final conclusion. The majority voting is considered as simple and effective scheme. This scheme follows democratic rules i.e. the class with highest number of votes is the outcome. The flow diagram for majority voting model is shown in following Fig. 2.

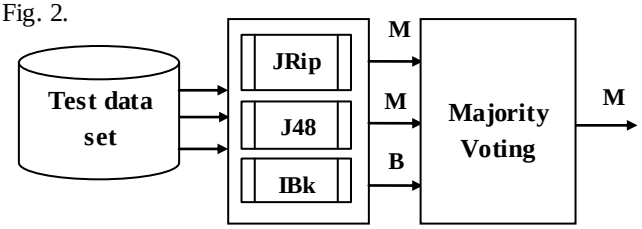

Fig. 2 Majority Voting Model

The test data set is given as input to classifiers. Every classifier has its own decision as whether the file is malicious or benign. The decision of multiple classifiers is combined and final decision is taken on the basis of majority voting. The decision taken by majority voting can be wrong if majority of classifiers will classify malware instance as benign. Also if no. of classifiers in the system is an even then it's difficult to make the decision because equal number of votes are given to both benign and malware classes. Such problems are overcome by using veto voting scheme.

b) Veto Voting

In veto voting the committee may give the right to veto the decision of committee to any member. It is used to give

importance to a single expert (classifier) who predicts against the majority. Any vote indicating an instance as malware, alone can determine the outcome of the classification task regardless of the count of other votes. For veto voting the n-layer classification model is used which is shown in the following Fig. 3.

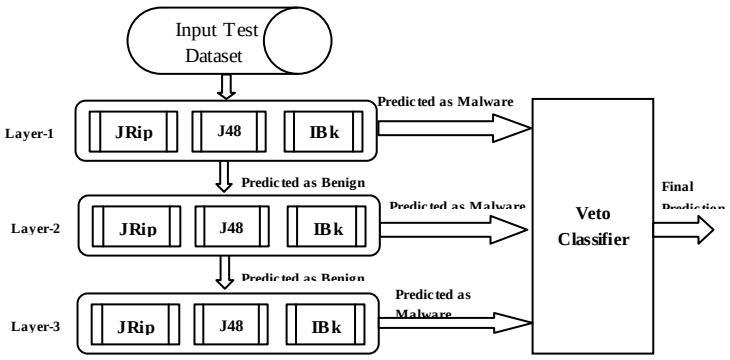

Fig. 3 N-layer classification model

The model consists of number of n layer and number of n classifiers. Here we used 3 layers and 3 classifiers with different configuration. Each layer can be customized with multiple n-gram sizes, multiple feature representations and learning algorithm. Here first layer is customized with classifiers that are trained using opcode n-gram dataset, second layer is customized with classifiers that are trained using overlapping n-gram dataset and third layer is customized with classifiers that are trained using non-adjacent opcode n-gram dataset.

The test dataset of benign and malicious files is given as input to n-layer classification model. From the upper layer, the instances that are declared as benign are given to lower layer for reclassification. If at any layer, an instance is classified as malware, it is not processed to the next layer. In other word any classifier indicating an instance as malware can act as veto to determine the outcome of final classification task, else instance further proceeds for reclassification. The classification results from all layers are given to the veto classifier. The decision strategy of veto is improved by introducing the trust-based veto voting.

c) Trust-based Veto Voting

Trust can be computed as +1 or -1, the increased or decreased value can help in determining the extent of the trust. The trust can be calculated as single trust or group trust. Mostly the group trust is calculated for different computational problems. A set of inference rules are used to value the trust i.e. $0 \le trust \le 1$ and derived value is further used for the decision. In trust-based veto voting three types of trust viz. local trust, recommended trust and global trust are calculated.

i. Local Trust

In local trust each algorithm in the system calculates its trust level for other algorithms in the system which means how much algop trusts the algoq in term of predicting the class of an instance, called as local trust. Local trust of algop on algoq is calculated by comparing the predictions (d) of both algorithms with each other and actual class (C) of instance, so from data set of benign and malicious instances, an instance of benign class is given to both algop and algoq for predicting the class of instance. The possible predictions are, both algorithms may predict correct or both algorithms may predict incorrect or any one of the algorithm may predict the correct class. If both algorithms give the same prediction either correct or incorrect, trust is not affected. If algop predicts the incorrect class and algoq predicts the correct class then algop increases the trust level of algoq with +1. If algop predicts the correct class and algoq predicts the incorrect class then algop increases the distrust level of the algoq with +1. Likewise all the instances in the dataset are given to both algorithms sequentially for the prediction. At the end of process local trust of algop on algoq is calculated by dividing the trust (sat) with the sum of trust (sat) and distrust (unsat). The algorithm for local trust calculation is given below

Algorithm: Local Trust Calculation

Input: Actual class of instance (C), Prediction of $algo_p$ (d_p), Prediction of $algo_q$ (d_q)

Output: Local trust of $algo_p$ on $algo_q$ $\left(t_q : a\lg o_p \to a\lg o_q\right)$

Repeat

If $\left(d_p = d_q\right)$ **then**

Move next

End if

If $\left(d_p \neq d_q\right)$ **then**

If $\left(d_p = C\right)$ **then**

$$Unsat = unsat\left(a\lg o_{p,} a\lg o_q\right) + 1$$

Else $\left(d_q = C\right)$

$$Sat = sat\left(a\lg o_{p,} a\lg o_q\right) + 1$$

End if

End if

Until !EOF

$$\left(t_q : a\lg o_p \to a\lg o_q\right) = \frac{sat\left(a\lg o_p, a\lg o_q\right)}{sat\left(a\lg o_{p,} a\lg o_q\right) + unsat\left(a\lg o_{p,} a\lg o_q\right)}$$

In our system we use three classifiers viz. JRip, J48 and IBk. So we calculate the local trust of JRip on J48 and vice versa, local trust of J48 on IBk and vice versa and local trust of IBk on JRip and vice versa. Local trust shows a unique trust on particular classifier from another classifier. This value varies from classifier to the classifier in the system and can't use as final metric for deciding about the

veto in the system. This local trust is used to calculate the recommended trust.

ii. Recommended Trust

The recommended trust is used to overcome the problem of local trust. The recommended trust is calculated by combining the local trust of all algorithms in the system on that particular algorithm. If the set of all algorithms is S = {algo0, algo1, …..,algon} and two subsets S'={algo0} and S''={algo1, algo2,…algon}. The subset S'' is having all the algorithms in the system as members except the algorithm algo0 for which the RT is calculated. The recommended trust is calculated by using the following formula.

$$RT_q \leftarrow \sum_{n=1}^{n} (t_q : algo_n \rightarrow algo_q) \ \forall algo_n \in S''$$

Using the above formula the recommended trust of JRip, J48 and IBk is calculated. The value of RT varies from algorithm to algorithm. The recommended trust of an algorithm is normalized to obtain the global trust of that particular algorithm.

iii. Global Trust

Basic purpose of normalization is to convert the different values on a notionally standard scale to compare them equally with each other. The normalized global trust value lies in the interval of the [0-1] and it is calculated by using the following equation:

$$GT_q \leftarrow \frac{RT_q}{\sqrt{\sum_{n=1}^{n} RT_n^2}}$$

iv. Veto Decision

The value of Global trust is used for deciding the veto. There are three classifiers in the system i.e. JRip, J48 and IBk. If two classifiers i.e JRip and IBk classify the instance as a benign and only one classifier i.e J48 classifies the instance as a malware then mean of J48 and JRip & IBk is calculated. If the mean of J48 is greater than mean of Jrip &IBk then J48 can veto the decision and outcome will be the prediction of the J48.

$$Veto : GT_{j48} \geq \frac{GT_{Jrip} + GT_{IBk}}{2}$$

In trust based veto voting each classifier evaluates the trust and maintains the trust information locally in a trust table without increasing the processing overhead. The locally stored trust information is used for decision purpose when required.

C. Evaluation Metric

1) *Overall Accuracy:* Measure number of absolutely, correctly classified instances either positive or negative divided by the entire number of instances.

$$OA = \frac{TP + TN}{TP + TN + FP + FN}$$

2) *Recall:* It is also called as True positive rate (TPR). It is the rate of number of positive instances classified correctly.

$$\text{Re}call / TPR = \frac{TP}{TP + FN}$$

3) *False Positive Rate (FPR):* It is number of negative instances misclassified.

$$FPR = \frac{FP}{FP + TN}$$

4) *Precision (P):* It represents the amount of samples classified as malicious that are really malicious.

$$P = \frac{TP}{TP + FP}$$

5) *F1-Measure:* It is the harmonic mean of precision and recall.

$$F1 - measure = \frac{2 * R * P}{R + P}$$

Where,

True Positives (TP): The number of malicious samples classified as malicious

True Negatives (TN): The number of benign samples classified as benign.

False Positives (FP): The number of benign samples classified as malicious.

False Negatives (FN): The number of malicious samples classified as benign.

IV. CONCLUSION

Ensemble based malware detection using different voting schemes can predict known as well as unknown malwares with high accuracy because ensemble model performs better than single classifier model in term of improving the detection accuracy. In proposed system static analysis is used which is safe and fast technique as files are analyses without its execution and it detects the malware accurately. Also Heuristic based malware detection is extended by

using the data mining and machine learning techniques to detect known as well as unknown malwares. Different voting schemes are used to determine which voting scheme is best to detect the known as well as unknown malware with high accuracy.

REFERENCES

[1] Symantec Corporation, Internet security threat report-2013, *Volume 18*

[2] Robin Sharp, An Introduction to Malware, Spring 2012. Retrieved on April, 10, 2013

[3] Imtithal A. Saeed, Ali Selamat, Ali M. A. Abuagoub, A Survey on Malware and Malware Detection Systems, *International Journal of Computer Applications (0975 – 8887) Volume 67– No.16*, April 2013

[4] Matthew G. Schultz, Eleazar Eskin, Erez Zadok, and Salvatore J. Stolfo, Data Mining Methods for Detection of New Malicious Executables, *in Proceedings of the Symposium on Security and Privacy*, 2001, pp. 38-49.

[5] Yi-Bin Lu, Shu-Chang Din, Chao-Fu Zheng, and Bai-Jian Gao, Using Multi-Feature and Classifier Ensembles to Improve Malware Detection, *JOURNAL OF C.C.I.T., VOL.39, NO.2*, NOV., 2010.

[6] R. K. Shahzad, S. I. Haider, and N. Lavesson, Detection of spyware by mining executable files, *in Proceedings of the 5th International Conference on Availability, Reliability, and Security. IEEE Computer Society*, 2010, pp. 295-302.

[7] Raja Khurram Shahzad, Niklas Lavesson, Henric Johnson, Accurate Adware Detection using Opcode Sequence Extraction, *in Proc. of the 6th International Conference on Availability, Reliability and Security (ARES11),Prague, Czech Republic. IEEE*, 2011, pp. 189-195.

[8] Asaf Shabtai, Robert Moskovitch, Clint Feher, Shlomi Dolev and Yuval Elovici, Detecting unknown malicious code by applying classification techniques on OpCode patterns, Security Informatics 2012, 1:1, http://www.securityinformatics.com/content/1/1/1.

[9] Robiah Yusof, Siti Rahayu Selamat, Shahrin Sahib, Intrusion Alert Correlation Technique Analysis for Heterogeneous Log, *IJCNS*, 2008

[10] Jianqiang Shi, Gregor V. Bochhmann, Carlisle Adams, A trust model with statistical foundation, System science, school of information technology and Engineering (SITE), University of Ottawa.

[11] Muazzam Ahmed Siddiqui, *Data Mining Methods For Malware Detection*, B.E. NED University of Engineering and Technology, Doctor of Philosophy in Modeling and Simulation, University of Central Florida, 2008

[12] Pham Van Hung, *An approach to fast malware classification with machine learning technique*, Keio University, 5322 Endo Fujisawa Kanagawa 252-0882 JAPAN, 2011

[13] R. K. Shahzad and N. Lavesson, Detecting scareware by mining variable length instruction sequences, *in Proc. of the 10th Annual Information Security South Africa Conference (ISSA11), Johannesburg, South Africa. IEEE*, August 2011, pp. 1-8.

[14] Raja Khurram Shahzad, Niklas Lavesson, Veto-based Malware Detection, *Seventh International Conference on Availability, Reliability and Security (ARES12), Prague, Czech Republic,IEEE,2012*, pp. 47-54

[15] Raja Khurram Shahzad, Niklas Lavesson, Comparative Analysis of Voting Schemes for Ensemble-based Malware Detection, *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, volume: 4, number: 1*, pp. 98-117.

[16] T.W. A. Grandison, *Trust management for internet applications*, Ph.D. dissertation, Imperial College of Science, Technology and Medicine, University of London, 2003.