Analysis of markov model and Guassian mixture model to malicious code detection

Dr. Chetan Shelke[1], Dr. Rahul Ghongade[2]
[12]P.R.Pote College of Engg & Mgt

Abstract- In the last decade, a more number of machine learning and data mining based approaches have been used in the areas of intrusion detection, malware detection & classification and also traffic analysis. In the area of malware analysis, static binary analysis techniques have become more difficult with the code obfuscation methods and code packing employed while writing the malware. The behavior-based analysis techniques are being used in large malware analysis systems because of that reason. In prior art, a number of clustering and classification techniques have been used to classify the malwares into families and also identify new malware families, from the behavior reports.

In this paper, we have analyzed in detail about the Profile Hidden Markov models for the problem of malware classification and clustering. The advantage of building accurate model with limited examples is very helpful in early detection and modeling of malware families. The paper also revisits the learning setting of an Intrusion Detection System that employs machine learning for identifying attacks and traffic. It substantiates the suitability of incremental learning setting(or stream based learning setting) for the problem for learning attack patterns in IDS, when large volume of data arrive in a stream. Related to the above problem, the survey of the IDS that use data mining and machine learning was done.

Keywords- Intrusion detection, malware detection & classification, Profile Hidden Markov models, IDS.

I. INTRODUCTION

In current years, intrusion detection technologies are indispensable for network and computer security as the threat becomes a serious issue year by year. Therefore Intrusion Detection Systems (IDSs) inspect all inbound and outbound network activities & identify suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise the system [1]. IDSystems are categorized into misuse detection and anomaly detection systems [2]. Misuse detection systems detect known attacks by the help of predefined attack patterns and signatures, e.g., a hacker attempting to break into an email server in a way that IDS has been already trained. Anomaly detection systems detect attacks by observing deviations from the normal behavior of the system & works by comparing network traffic, system call sequences, or other features of known attack patterns.

The Profile Hidden Markov Model is a probabilistic approach that was developed specifically for modeling sequence similarity occurring in biological sequences such as proteins and DNA[3][4]. It is also a faster alternative to the traditional deterministic approaches used in sequence matching [4]. It is a modified implementation of HMM, which is basically a generative model and constructs a probabilistic finite state machines. For behavior based analysis, we again assume that there is a sequence of operations common for a virus family and for a presented new sequence we would like to find the best known match from the database.

In general, intrusion detection using Hierarchical Gaussian Mixture Model (HGMM) is the process of finding the abnormal packets in the network. There are two phase in the process of HGMM. The first is the training phase in that reference templates are generated by HGMM. So, the attacks have to be trained to the system. The features were extracted from the sample data that is provided by traffic, in order to obtain the data for statistical modeling. Next phase is the detection phase. During the detection phase, the input packet's deviation from the stored reference models is calculated and recognition decision is made as to which model suits that packet.

II. MALWARE ANALYSIS

The malware analysis that Antivirus companies do, can be classified broadly into two categories; the static analysis techniques and the dynamic analysis techniques. The static techniques involve looking into the binaries directly or the reverse engineering the code for patterns in the same. The dynamic analysis techniques involve capturing the behavior of the malware sample by executing it in a sandboxed environment or by program analysis methods and then use that for extracting patterns for each family of virus. Examples for these are systems like Anubis, [5] and CWSandbox. Of late static binary analysis techniques are becoming increasingly difficult with the code obfuscation methods and code packing employed when writing the malware.

III. Profile Hidden Markov Models

The main reason for us to choose this approach for solving the problem of finding malware similarity is because the behavior of malware program has variablility, yet has a characteristic signature reected in the sequence of system calls. For example if we look at the CWSandbox reports for two malware programs from same family, we notice that a sequence of malicious actions is preserved, interspersed with some other actions introduced to confuse the malware detection system. A hidden markov model (HMM) is very suitable for probabilistic modelling of such sequences, which is evident from past works. Thus it can be used for modelling difierent classes of malware. But as we have discussed above , there might be additions, deletions or changes to the system calls for difierent programs within same malware family. The profile HMM is exactly designed to model this kind of problem, because it also has non-emitting states or the delete states.

IV. HIDDEN MARKOV MODELS

A hidden markov model(HMM) is a statistical tool which captures the features of one or more sequences of observable symbols by constructing a probabilistic finite state machine with some hidden states that are emitting the observed symbols [6]. When the state machine is trained, its graph and the transition probabilities are computed such that they best produce the training sequences. When we test with a new sequence, the HMM gives a score for how best the sequence matched with the known state machine .In our case, the observed symbols are the codes for each unique system call in the behavior report of the malware program(MIST codes).

An HMM is specified by the following parameters.

- the alphabet of symbols ∑
- the hidden state set Z
- the emission probability matrix $E_{|Z|z|\sum|}$
- the state transmission matrix $A_{|Z|z|Z|}$
- the initial state distribution π

Thus the HMM λ can be written as λ= (∑,Z,A,E,π). This model can thus be used to assign a probability to an observed sequence X as follows

$$P(X|\lambda) = \Sigma_z \Pi_k Azk, zk + 1Ezk, Xk$$

This probability as indicated by the formula, is that of emitting the observation sequence X after all possible state transitions(i.estate transmission sequences). of the model λ.The model λ has to be learnt from training data consisting of independent and identically distributed sequences. This can be done by maximizing the probability P(T|λ)where T is a training sequence. There is no analytical solution to this, however this can be done by using an iterative procedure that uses E-M (Expectation-Maximization) algorithm[6].Given a sequence X, the Viterbi algorithm[7] can be used to compute the hidden state Z, so as to maximise P(Z|X) i.e determine most probable sequence of hidden states that produced the observed sequence. Equation 1 can then be evaluated using the likelihood and P(X) got using the forward and backward procedures [6].A PHMM is a specific formulation of a standard HMM that makes explicit use of positional information contained in the observation sequences[8]. PHMM is a strongly linear left-right model while HMM is not[6]. A PHMM model allows null transitions, so that it can match sequences that differ by point insertions and deletions happening by chance mutations. They were specifically formulated for use in bioinformatics, where such insertions and deletions to DNA sequences were natural during evolution. Thus PHMMs can be seen effective in modeling metamorphic malware, that also go through similar kind of evolution, both at binary level and at a behavioral level. Furthermore, HMM state transition matrices are essentially sparser than those of HMM, allowing quicker inference

The Viterbi algorithm, forward backward procedure and Expectation-Maximization are naturally extended to PHMMs. In PHMM, the emission probabilities are position dependent unlike in standard HMM. Learning a profile HMM from data involves computing the emission probability matrix E and the state transition probability matrix A using the multiple sequence alignment data. These are given by g(x|μi , Σi), i = 1, . . . , M, are the component Gaussian densities. with mean vector μi and covariance matrix Σi . The mixture weights satisfy the constraint that PM i=1 wi = 1. The complete Gaussian mixture model is parameterized by the mean

$$\mathrm{A_{uv}} = \frac{N_{uv}^{A}}{\Sigma_v N_{uv}^{A}}$$

$$\mathrm{A_{uv}} = \frac{N_{ut}^{E}}{\Sigma_t N_{ut}^{E}}$$

Where represents the number of transitions from the state u to v and , the number of emissions of t given a state u.[8] After the model λ has been learnt from the training multiple alignment data, the problem of identifying the family that a new sequence X belongs to, is decided by the rule

y(X) = argmaxkP(X|λk)

V. OVERVIEW OF GAUSSIAN MIXTURE MODEL

A Gaussian mixture density is a weighted sum of *M* component densities, as is depicted in Fig. 1 and given by [10].

$$p(x|\lambda) = \sum_{i=1}^{M} \mathrm{wi\ g(x|\mu i\ ,\Sigma i)},$$

where x is a D-dimensional continuous-valued data vector (i.e. measurement or features), wi , i = 1, . . . , M, are the mixture weights,and g(x|μi , Σi), i = 1, . . . , M, are the component Gaussian densities. with mean vector μi and covariance matrix

Σi . The mixture weights satisfy the constraint that PM i=1 wi = 1. The complete Gaussian mixture model is parameterized by the mean vectors, covariance matrices and mixture weights from all component densities. These parameters are collectively represented by the notation,

$$\lambda = \{wi , \mu i , \Sigma i\}\ i = 1, . . . , M.$$

There are several variants on the GMM shown in Equation . The covariance matrices, Σi , can be full rank or constrained to be diagonal. Additionally, parameters can be shared, or tied, among the Gaussian components, such as having a common covariance matrix for all components, The choice of model configuration (number of components, full or diagonal covariance matrices, and parameter tying) is often determined by the amount of data available for estimating the GMM parameters and how the GMM is used in a particular biometric application. It is also important to note that because the component Gaussian are acting together to model the overall feature density, full covariance matrices are not necessary even if the features are not statistically independent. The linear combination of diagonal covariance basis Gaussians is capable of modeling the correlations between feature vector elements. The effect of using a set of M full covariance matrix Gaussians can be equally obtained by using a larger set of diagonal covariance Gaussian.[10].

VI. CONCLUSION

In this way we are compared markov model and Gaussian mixture model to malicious code detection ,In this case HMM is more useful than GMM. Also GMMs are universal approximators of densities (provided sufficient no. of mixtures are used); true for diagonal GMMs as well A hidden markov model(HMM) is a statistical tool which captures the features of one or more sequences of observable symbols by constructing a probabilistic finite state machine with some hidden states that are emitting the observed symbols.

VII. REFERENCES

[1]. K. Rieck, P. Trinius, C. Willems, and T. Holz: Automatic Analysis of Malware Behavior using Machine Learning, In Journal of Computer Security 2011.

[2]. K. Rieck, T. Holz, C. Willems, P. Dussel, and P. Laskov, Learning and classification of malware behavior, in Proc. of DIMVA 2008, pp. 108125.

[3]. S. R. Eddy. Profile hidden Markov models. Bioinformatics,14(9):755763, 1998

[4]. S. R. Eddy. HMMER: Sequence analysis using profile hidden Markov models.

[5]. R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. Biological Sequence Analysis: Probabilistic models of proteins and nucleic acids. Cambridge University Press,1998.

[6]. International Secure Systems Lab, Anubis: Analyzing unknown binaries, http://anubis.iseclab.org/

[7]. S. Attaluri, S. McGhee and M. Stump: Profile hidden Markov models and metamorphic virus detection, In Journal of Computer Virology(2009) 5:151-169

[8]. Malheur Dataset: http://pi1.informatik.uni-mannheim.de/malheur/#dldata

[9]. Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE, 77 (2), 257-286.

[10]. D. Reynolds. Gaussian Mixture Models. *Encyclopedia of Biometric Recognition*, *Springer*, Feb. 2008.