Virtual Machine Failure Prediction Method Based on AdaBoost-Hidden Markov Model

Zhixin Li[1, 2*], Lei Liu[1], Degang Kong [2]
1 College of Computer Science and Technology, Jilin University, Changchun 130012, China
2 School of Computer Technology and Engineering, Changchun Institute Of Technology, Changchun 130012, China
52868081@qq.com

***Abstract*—The failure prediction method of virtual machines (VM) guarantees reliability to cloud platforms. However, the uncertainty of VM security state will affect the reliability and task processing capabilities of the entire cloud platform. In this Study, a failure prediction method of VM based on AdaBoost-Hidden Markov Model was proposed to improve the reliability of VMs and overall performance of cloud platforms.This method analyzed the deep relationship between the observation state and the hidden state of the VM through the hidden Markov model, proved the influence of the AdaBoost algorithm on the hidden Markov model (HMM), and realized the prediction of the VM failure state. Results show that the proposed method adapts to the complex dynamic cloud platform environment, can effectively predict the failure state of VMs, and improve the predictive ability of VM security state.**

Keywords- Virtual Machine; Failure Prediction; Hidden Markov Model

I. INTRODUCTION

Virtualization services offer a new service mode for public facilities. Virtualization services not only save cost, but also provide convenient and efficient computing resources [1]. Meanwhile, the dynamic and complexity of the cloud platform environment cause virtual machine (VM) operation failures to increase, and the reliability and security of VMs gradually rise, which become an important factor restricting its development [2].As a very important security mechanism, security state prediction of VM in cloud platform is an important means to ensure the reliable operation of VM. Numerous researches have reported about security state modeling [3], Fault Tolerance of VM [4], etc. It is generally believed that learning and predicting of the security state of VMs plays an important role in ensuring the reliable operation of VMs. Currently, most of these methods use the observable state of the cloud platform, such as server load, log, and other data information, to learn and predict the security state of VM. Hidden Markov model (HMM) is one of the commonly used methods [5]. J. Wu et al. [6] proposed a software state assessment prediction method based on HMM. This method used K-means method to construct the observation state of the system and used linear regression method to predict the characteristic parameters of the system, which solved the prediction analysis of the system parameters on the time series to some extent. P. Kumar et al. [7] proposed a method of data mining clustering based on HMM to solve the security problem of cloud computing. However, the initial cluster center selection had a great influence on the clustering result, and the effective clustering result may not be obtained, so that the observable state could not reflect the internal security state. X. Zan et al. [8] proposed a tracking prediction attack intent algorithm using HMM, which could identify false alarms and give credible prediction results. These methods could effectively speculate the hidden security state through the HMM model. At the same time, due to the complex structure of the cloud platform VM system, it is difficult to accurately evaluate and predict the actual internal security state. Therefore, the internal security state of the VM can only be inferred through the observable state, and the VM failure prediction is realized.

In this study, the security state of VM in the cloud platform was established through the HMM to describe the VM running state when performing VM failure prediction. Then, the observation state prediction was realized by AdaBoost algorithm, and the influence of prediction error on hidden state learning prediction of HMM was proved. Finally, the failure prediction method of VM was implemented. The proposed method solved the problem that the security state of VM was difficult to predict, made up for the impact of potential security threats on the VM system, and improved the security of the cloud platform VM system.

II. VIRTUAL MACHINE FAILURE PREDICTION METHOD

A. VM failure state prediction modeling

The VM failure state prediction is to estimate and predict the VM security state in the cloud platform by collecting the data of the running state of the VM in the cloud platform and observing the CPU, memory, network, VM load and other information generated by the hidden state of the VM. The VM observation state sequence is probabilistically related to the hidden state sequence, which is consistent with the HMM that can be applied to the prediction of the VM running state. In this study, the VM state running model is established by HMM, and the model parameters are trained and learned to predict the state of VM, so as to adjust the response strategy.

A HMM consists of hidden state space Y, observation space X, a state transition probability A, output observation probability B, and initial state probability π. Parameter $\lambda = (A, B, \pi)$ is set.

Firstly, the state observation set and the hidden state set of VM are determined. In this study, the hidden state of VM is divided into four states: normal state, warning state, error state and failure state, that is $n = 4$. So the transfer matrix of hidden state probability is:

978-1-7281-1307-4/19/$31.00 ©2019 IEEE
DOI 10.1109/ICITBS.2019.00173

$A = \begin{bmatrix} a_{11} a_{12} a_{13} a_{14} \\ a_{21} a_{22} a_{23} a_{24} \\ a_{31} a_{32} a_{33} a_{34} \\ a_{41} a_{42} a_{43} a_{44} \end{bmatrix}$, $\sum_{j=1}^{4} a_{ij} = 1$, The VM state changes as shown in Figure 1:

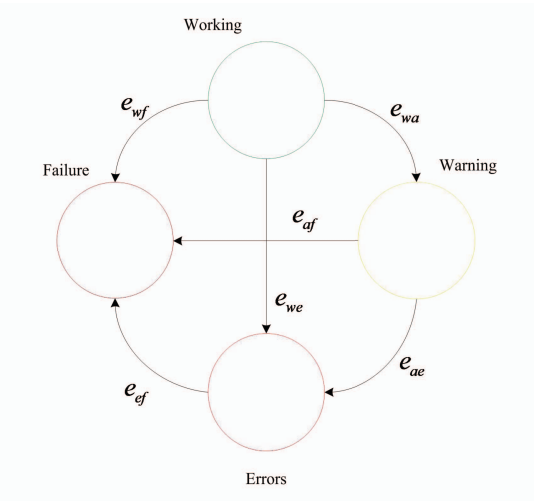

Figure 1. Virtual machine running state diagram

The VMs in the cloud platform are working during the running of the VM. With the increase of running time, the VM is in the warning state when a software error (e_{wa}) occurs. If the cloud platform management software does not detect the error, the VM may enter an error (e_{ae}) or fault state (e_{af}) as the error accumulates. In this state, the VM cannot complete its work and enter the failure state. It is also possible for a VM to enter errors (e_{we}), failure state (e_{wf}), and enter a failure state (e_{ef}) from errors. Therefore, this study uses the HMM to discover the hidden security state inside the VM in the cloud platform.

The real-time and accuracy of the learning prediction of the observation state plays a key role in VM failure state prediction. The observation state of the VM often has a certain regularity. The observation state parameter has a strong correlation with the time, meanwhile, the observation state parameter at the previous moment has a great influence on the observation state parameter at the next moment. So the prediction of the observed state selects the AdaBoost algorithm. AdaBoost is an iterative algorithm that combines multiple linear regression weak learners and then produces a strong learner.

Definition 1: Weak predictor. Using the initial weight distribution D_n to train the i-th weak predictor, $G_n(x): \hat{x}_i(t) = \theta_0 + \theta_1 * t$.

$$\theta_0 = \frac{\sum (t-1)^2 \sum x_j(t-1) - \sum (t-1) \sum (t-1) * x_j(t-1)}{n \sum (t-1)^2 - (\sum (t-1))^2} \tag{1}$$

$$\theta_1 = \frac{n \sum (t-1) * x_j(t-1) - \sum (t-1) \sum x_j(t-1)}{n \sum (t-1)^2 - (\sum (t-1))^2} \tag{2}$$

$\hat{x}_i(t)$ is the observation state prediction value at time t . θ_0 , θ_1 are obtained by least squares method.

Definition 2: Error rate,

$$e_n = P(|\hat{x}_i(t) - x_i(t)| > c) = \sum_{i=1}^{t} w_{ni} I(|\hat{x}_i(t) - x_i(t)| > c) . \tag{3}$$

where $|\hat{x}_i(t) - x_i(t)| > c$ means when the absolute value difference between observation state prediction value $\hat{x}_i(t)$ and actual observation value $x_i(t)$ is higher than c , error calculating starts.

Definition 3: Error coefficient,

$$\mu_n = \frac{1}{2} \log \frac{1 - e_n}{e_n} . \tag{4}$$

Definition 3: Weight $D_{n+1} = (w_{n+1,1}, w_{n+1,2}, ..., w_{n+1,t})$,

$$w_{n+1,i} = \begin{cases} \frac{w_{ni}}{Z_n} e^{-\mu_n}, & |\hat{x}_i(t) - x_i(t)| \le c \\ \frac{w_{ni}}{Z_n} e^{\mu_n}, & |\hat{x}_i(t) - x_i(t)| > c \end{cases} . \tag{5}$$

where Z_n is a normalization factor. That is, when $(|\hat{x}_i(t) - x_i(t)| > c)$ or $(|\hat{x}_i(t) - x_i(t)| \le c)$, perform summation to Formula (5) $Z_n = \sum_{|\hat{x}_i(t) - x_i(t)| > c} w_{ni} e^{\mu_n} + \sum_{|\hat{x}_i(t) - x_i(t)| \le c} w_{ni} e^{-\mu_n}$.

Definition 5: Strong predictor of VM observation state,

$$G(x) = \mathrm{sign}\left(\sum_{n=1}^{N} \mu_n G_n(x) \right) . \tag{6}$$

Through the Baum-Welch method [9], the maximum likelihood estimation of the HMM parameters is derived from the VM observation state sequence, which can solve an approximate solution to the HMM parameter learning problem, and the description is omitted here. Using the VM observation state predictor, it can predict the observation state value of the VM for the next time period.

B. Theoretical analysis and related proof of VM state prediction

The generalization ability of VM observation state learning and predicting is an important property of predicting method. Usually, the upper bound of probability of generalization error is used to evaluate the generalization ability of learning method. Because the VM state data set is limited, the predicting of the observation state aims to make the AdaBoost model have good prediction ability not only for the known observation state but also for the unknown observation state. So this study theoretically analyzes the generalization ability of the AdaBoost model.

Theorem 1: The upper bound of the training error of the AdaBoost algorithm on the VM observation state data is

$$\varepsilon = \frac{1}{t} \sum_{i=1}^{t} I(|\hat{x}_i(t) - x_i(t)| > c) \le \prod_n Z_n = \prod_{n=1}^{n} [2\sqrt{e_n(1-e_n)}] . \tag{7}$$

Proof: Deform the training error formula $\frac{1}{t} \sum_{i=1}^{t} I(|\hat{x}_i(t) - x_i(t)| > c)$ into $\frac{1}{t} \sum_{i=1}^{t} I = \begin{cases} 1 & |\hat{x}_i(t) - x_i(t)| > c \\ 0 & |\hat{x}_i(t) - x_i(t)| \le c \end{cases}$, thus

$\frac{1}{t}\sum_{i=1}^{t} I(|\hat{x}_i(t)-x_i(t)|>c) \le 1$. At the same time, $\prod_n Z_n = Z_1 Z_2 ... Z_n$.

Known by Formula (5), $Z_n w_{n+1,i} = \begin{cases} w_{ni} e^{-\mu_n}, & |\hat{x}_i(t)-x_i(t)| \le c \\ w_{ni} e^{\mu_n}, & |\hat{x}_i(t)-x_i(t)| > c \end{cases}$, then $\prod_n Z_n \ge 1$ can be known. Thus, $\frac{1}{t}\sum_{i=1}^{t} I(|\hat{x}_i(t)-x_i(t)|>c) \le \prod_n Z_n$.

Meanwhile, the error rate of each basic linear regression predictor is known to be $e_n = P(\hat{x}(t)-x_i(t)|>c) = \sum_{i=1}^{t} w_{ni} I(|\hat{x}(t)-x_i(t)|>c)$, where w_{ni} indicates weight of the i-th linear regression predictor at the nth round. According to the Gauss-Markov theorem [10], each linear regression predictor has a minimum variance. It can be known from Formula (5), $Z_n = \sum_{|\hat{x}_i(t)-x_i(t)|>c} w_{ni} e^{\mu_n} + \sum_{|\hat{x}_i(t)-x_i(t)|\le c} w_{ni} e^{-\mu_n}$, thus $\sum_{|\hat{x}_i(t)-x_i(t)|>c} w_{ni} e^{\mu_n} + \sum_{|\hat{x}_i(t)-x_i(t)|\le c} w_{ni} e^{-\mu_n} = (1-e_n)e^{-\mu_n} + e_n e^{-\mu_n} = 2\sqrt{(e_n(1-e_n))}$.

It can proves that the upper bound of the training error $\prod_n Z_n = \prod_{n=1}^{n} [2\sqrt{e_n(1-e_n)}]$.

Theorem 1 shows that the AdaBoost algorithm can predict the observation state data of VM, and has the upper bound of the training error, which can accurately predict the state of the next moment. At the same time, according to Reference [11], the smaller the training error is, the smaller the generalization error will be.

Theorem 2: The influence of AdaBoost algorithm on HMM hidden state learning depends on the observation state error c.

Proof: The upper bound of the error is known by Theorem 1, set $\varepsilon = \prod_{n=1}^{n} [2\sqrt{e_n(1-e_n)}]$, e_n is expressed by Formula (3). Only when $|\hat{x}_i(t)-x_i(t)|>c$, can the error rate e_n start to calculate. Hence, the value of c influences the upper bound of error. It can be seen from Formula (3) that, when the value of c is smaller, the error rate e_n becomes large. If all predicted values exist $|\hat{x}_i(t)-x_i(t)| \le c$, at this time $e_n = 0$, upper bound of error $\varepsilon = 0$.

As can be seen from Baum-Welch method, the maximum probability prediction value of the hidden state at time of $t+1$ is $\hat{\delta}_{t+1}(i) = [\max_j \delta_t(j) a_{ij}] b_{i(\hat{x}_{t+1})}$, where $b_{i(\hat{x}_{t+1})}$ represents probability of the hidden state to the predicted observation state. At the same time, the prediction value of observation state is the sum of the actual state value and the upper bound of the error, $\hat{x}_{t+1} = x_{t+1} + \varepsilon$. Therefore, when ε is approaching 0, that is $\lim_{\varepsilon \to 0} \frac{\hat{\delta}_{t+1}(i)}{\delta_{t+1}(i)} = \lim_{\varepsilon \to 0} \left(\frac{[\max_j \delta_t(j) a_{ij}] b_{i(x_{t+1}+\varepsilon)}}{[\max_j \delta_t(j) a_{ij}] b_{i(x_{t+1})}} \right) = 1$. It can be proved that if the observation state error c is reasonable, the HMM model hidden state has less influence at the next moment.

Theorem 2 indicates that the value of c reflects the generalization learning ability of AdaBoost. The training and learning of the observation state data set of the cloud platform through AdaBoost, can make the prediction value more accurate at the next moment $t+1$. Then the learning of hidden state of the HMM model at time $t+1$ will be more accurate, and the hidden state of the cloud platform can be predicted more accurately.

This section analyzes the relationship between HMM and AdaBoost. The AdaBoost error and the impact of the error on the security state of the cloud platform are also analyzed. Theorem 1 and Theorem 2 show that the prediction result of the VM observation state by AdaBoost, combined with HMM, the internal security state probability of VM can be inferred.

C. *Virtual machine failure prediction algorithm*

The overall architecture of the VM failure prediction algorithm is shown in Figure 2. Firstly, the observation state data of the cloud platform is collected. The AdaBoost algorithm is used to predict the observation state data of VM at the next moment. Next, the observation state data is preprocessed to conform to the input of the HMM model. We can get $\lambda = (A, B, \pi)$ through the HMM model parameter learning. Finally, the HMM model is used to calculate the corresponding VM hidden state. The parameters of the HMM model and the AdaBoost algorithm are obtained through data training as described above. The algorithm can predict the probability of VM future failure state. We can choose the appropriate strategy based on the future security state to ensure the normal running of the cloud platform.

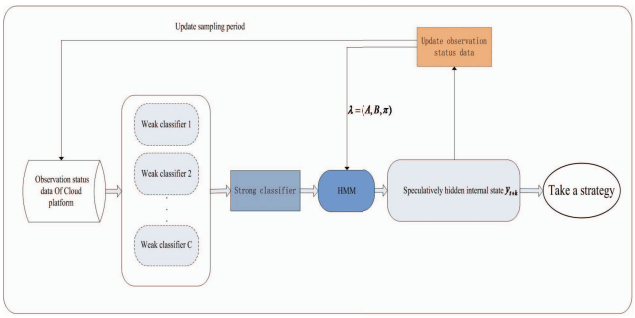

Figure 2. VM failure prediction algorithm diagram

III. EXPERIMENT AND RESULT ANALYSIS

A. *Experimental Environment Deployment*

`The cloud platform server is equipped with one Sugon A840-G10 server. CPU: AMD 6376, 16 cores 2.3GHz × 4. 256 GB memory, and Gigabit LAN. Software environment deployment: Configure the Xen virtualization platform on the server .The Red Linux 5.0 operating system is installed on the VM, and the Nigix application service program is installed on it to establish a distributed website, version JDK1.6.

B. *Experiment and result analysis*

Firstly, the observation state of VM needs to be predicted, and the result is taken as the input of the HMM, according to VM failure state prediction algorithm Figure 2. By attacking DDos with Low Orbit Ion Canon (LOIC) simulation and initiating simulation job scripts, VM observation state are tested and data are collected. The AdaBoost prediction method is based on the description of Theorem 2.The value of the observation state error are $c = 0.005$, $c = 0.01$ and $c = 0.1$, and test the effect of on the observation state of VM.

The experimental results are shown in Figure 3, and the mean square error (MSE) of $c = 0.005$, $c = 0.01$ and $c = 0.1$ are 0.0027, 0.0036 and 0.0173 respectively. It can be seen that the smaller the value of c is , the more accurate observation state results of VM according to the experimental results. However, the more iterations will affect the learning efficiency.

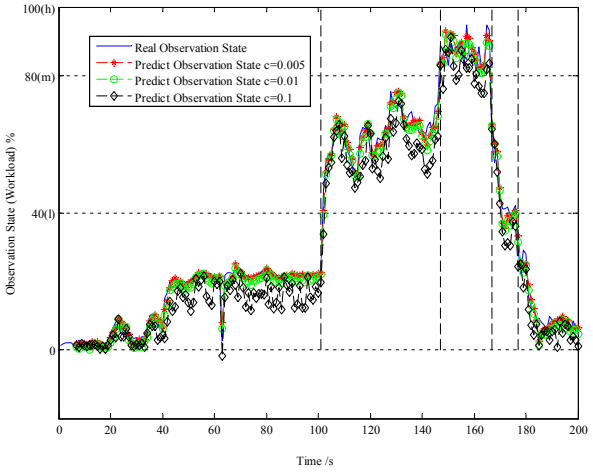

Figure 3. Observation state prediction of VM

The following is to verify the prediction ability of AdaBoost-HMM about the running state of VMs. By predicting and analyzing the current state of VM in real time, and outputting the relevant probability result, it is determined whether the VM is failure. The experiment results are shown in Table 1. The experiment results show that the value of c has an influence on the probability output of HMM. The smaller the value of c is, the more accurate the failure prediction of VM.

Table 1. HMM probability output results

	Time	Working	Warning	Error	Failure
$c = 0.005$	150-170 s	0.02	0.332	0.443	0.205
	170-200 s	0.403	0.326	0.252	0.019
$c = 0.01$	150-170 s	0.019	0.308	0.401	0.272
	170-200 s	0.486	0.309	0.109	0.096
$c = 0.1$	150-170 s	0.145	0.201	0.301	0.353
	170-200 s	0.376	0.106	0.209	0.309

In this experiment, the prediction accuracy of the VM observation state and the impact of the prediction error based on HMM have been verified. The results demonstrate that AdaBoost-HMM method can effectively predict the future operation state of VM, which can very well protect the service capability of the VM platform system.

IV. CONCLUSION

In order to ensure the reliable operation of VMs, a virtual machine failure prediction method based on AdaBoost-Hidden Markov model was proposed. The relationship model of VM resource state, load, and runtime can reflect the security state of VM in time and prevent security risks of VM. Based on the AdaBoost-HMM model, the solution to the security state probability of the VM in the future time period is realized in this study, which can learn and predict the security state of the VM in a certain period of time. However, in learning and predicting modeling process, the observation state of VM is valued by the load, and the relationship between observation state of VM and internal security state needs to be further explored. The problem will be studied to improve applicability of the learning and predicting method of VM failure state.

ACKNOWLEDGEMENTS

This work is supported by following projects: the Key Program for Science and Technology Development of Jilin Province of China (Grant No. 20130206052GX). Project supported by the National Natural Science Foundation of China (Nos. 61602057), the Science and Technology Department of Jilin Province, China (No. 20170520059JH)

REFERENCES

[1] A. Li, X. Yang, S. Kandula, et al. CloudCmp: comparing public cloud providers. Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, ACM, 2010: 1-14.

[2] M. Pearce, S. Zeadally, and R Hunt. Virtualization: Issues, security threats, and solutions. ACM Computing Surveys, 2013, 45(2): 1-40.

[3] H. Wei, G. Y. Hu, Z. J. Zhou, et al. A new BRB model for security-state assessment of cloud computing based on the impact of external and internal environments. Computers & Security, 2018, 73: 207-218.

[4] Z. Li, L. Liu, et al. Study on Fault Tolerance Method in Cloud Platform based on Workload Consolidation Model of Virtual Machine. Journal of Engineering Science & Technology Review, 2017, 10(5): 41-49.

[5] L. R. Rabiner, B. H. Juang. An introduction to hidden Markov models. ieee assp magazine, 1986, 3(1): 4-16.

[6] J. Wu, W. R. Zeng H. L. Chen, et al. Approach of Measuring and Predicting Software System State Based on Hidden Markov Model. Journal of Software, 2016, 27(12): 3208-3222.

[7] P. Kumar, V. Sehgal, K. Shah, et al. A novel approach for security in cloud computing using hidden markov model and clustering. Proceedings of the Information and Communication Technologies, 2011 World Congress on IEEE, 2011: 810-815.

[8] X. Zan, F. Gao, J. Han, et al. A Hidden Markov Model based framework for tracking and predicting of attack intention. Proceedings of the Multimedia Information Networking and Security, 2009 International Conference on IEEE, 2009: 498-501.

[9] L. R. Welch. Hidden Markov models and the Baum-Welch algorithm. IEEE Information Theory Society Newsletter, 2003, 53(4): 10-13.

[10] D. Harville. Extension of the Gauss-Markov theorem to include the estimation of random effects. The Annals of Statistics, 1976, 4(2):384-395.

[11] Y. Freund, R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences, 1997, 55(1): 119-139.