Cloud Malicious Node Detection and Resource Management by Tversky Trust

Pragya Richhariya
Research Scholar
Computer Science and Engineering
Rabindranath Tagore University
MP, India

Dr. Shailja Sharma
Associate Professor
Computer Science and Engineering
Rabindranath Tagore University
MP, India

ABSTRACT— With the development of the Internet, many businesses will attract users by marketing activities on terminal devices, such as: Android Phone and IOS Phone. These marketing activities also attract a lot of malicious users, who usually profit from malicious behaviors. This kind of malicious behavior will cause the merchants to invest a lot of money in marketing activities without really attracting users and causing a lot of losses. Tversky trust was evaluate from the cloud nodes in order to identify nature of infected and un-infected nodes. For prediction of model virtual nodes other features were also used like memory, processor, utilization, etc. Training of error back propagation neural network was done by use of features and predict the nature of node. Experiment was done on different environment by varying number of nodes and result shows that proposed model has increases the work detection accuracy as compared to previous existing models.

Index Terms— Cloud computing, Trust Coefficient, Page Rank, Classification, Neural Network.

I. INTRODUCTION

Cloud computing allows for the integration of multiple resources, including computational resources, to provide an integrated service to the end user. Cloud Computing refers to IT and business resources such as servers, storage, networks, applications, and processes that can be dynamically allocated based on the demands and workload of the user. Cloud computing refers to the storage and retrieval of data and programs via the internet from a remote place. Local storage and computing occur when we store data on or run a program from the hard disc of our local computer. The NIST defines cloud computing as "a concept for enabling ubiquitous, accessible, on-demand network access to a shared pool of programmable computing resources that can be swiftly produced and delivered with minimal administrative effort or cost." [1]. This is made up of five key features, three service models, and four deployment models.

The co-resident attack is a serious security vulnerability in cloud infrastructure. Tenants' virtual machines (VMs) can be allotted on the same host using virtualization technologies offered by the Cloud Service Provider. A multi-tenant system allows hostile renters to undertake a co-resident attack and steal information from other tenants via side channels. Previous research has focused on reducing side channels to prevent this type of assault, with only a few studies focusing on VM deployment strategy [2]. As a result, we concentrate on deploying VMs with a safe and effective allocation method to limit the likelihood of VM.

Because the cloud provides a truly massive and internet-based environment, the vulnerabilities for cyber-attacks are greater than with traditional systems. If the environment has some scaling limitations, then the services, applications, and users with access control are completely controlled and monitored. However, because cloud computing environments are built on internet connections, all of the services contained in the internet are running in the same condition. Cyber-attacks on the internet are also becoming a potential hazard in cloud computing. Virtualization technology is likely to introduce new vulnerabilities. It is a significant issue for security professionals to safeguard all VMs in the computer environment.

This study [3] describes the hypervisor's functionality and discusses exploits and weaknesses in virtualization environments[4-5]. The hypervisor should strictly manage the communication between the VMs, limit the resources, and check the usage of the resources by the VM on a regular basis to prevent DoS attacks.

II. Related Work

Gartner [4] identified four factors influencing cloud adoption in January 2020, with dispersed multi-cloud scenarios becoming more widespread. One of them is dealing with related security and privacy issues.

In 2019, the authors of [5] explored data security concerns from the perspective of a developing country, namely Nepal. The study identified developing-country difficulties such as confidentiality, billing model, breaches, segregation, access, integrity, security, storage, data centre management, service level agreement, charging model, costing model, and proximity. According to the findings of this study, the top security risks are storage, virtualization, and networks.

In [6] provides a survey of the previous five years of research articles on consumer-oriented IoT cloud applications for the understanding of smart IoT cloud systems. The author introduced a novel IoT cloud paradigm and performed a security study of the IoT cloud system.

In [7], a research study proposes Modified Elliptic Curve Cryptography (MECC) based on the Diffee Hellman algorithm, which provides increased security through alternate key generation. When the encryption, decryption, upload, and download times are calculated, it is determined that the technique suggested in this research study uses less time for all of these measures when compared to other existing algorithms. The improved ECC achieves high efficiency thanks to characteristics such as reduced memory, excellent operational performance, tiny sized keys, a speedy key generation process, and effective resource savings.

In [8] authors present a two-part approach that allows the hypervisor to build believable trust relationships with guest Virtual Machines (VMs) by analyzing objective and subjective trust sources and aggregating them using Bayesian inference. On top of the trust model, we create a trust-based max min game between DDoS attackers attempting to decrease the detection of the cloud system and hypervisors attempting to maximize this minimization under a limited budget of resources. The game solution leads the hypervisor in real-time detection load distribution among VMs to maximise DDoS assault detection.

In [9], first build the fundamental Verifiable SE Framework (VSEF), which can survive internal KGA and accomplish verifiable searchability. We then demonstrate the upgraded VSEF, which supports multi-keyword search, multi-key encryption, and dynamic updates (e.g., data modification, data insertion, and data deletion) at the same time, emphasising the relevance of practicability and scalability of SE in real-world application scenarios.

The authors provide a trust-based safe multi-cloud collaboration architecture for Cloud-Fog-Assisted IoT systems in their study [10]. To ensure user security, we develop a role-based trust evaluation approach to improve the trustworthiness of MCSC. We design an efficient user authentication method and a safe collaboration system to provide a collaborative user authentication and access control mechanism for MCSC in order to maintain service security.

In [11] recommended using checking nodes to help detect such behavior. It then does a gain-loss analysis for providers who plan to engage in provider-user collusion deception. The proposed trust model can be utilized to successfully aid in the recognition of collusion deceitful conduct and allow policymakers to establish appropriate losses to punish malevolent providers. As a result, provider-initiated collusion deceitful behaviour can be significantly reduced.

III. Proposed Methodology

Cloud malicious nodes were detected by the activities done by the machines. So this work performed this monitoring centrally by collecting session status complete or missed. Nodes instant information of memory, processor utilization was also reordered and used for the study. Fig. 1 and 2 shows CMMTT (Cloud Machine Management by Tversky Trust) proposed model node status (Normal/Malicious) [12]. All types of nodes will report to centralized data collection unit in cloud.

Table 1 CMMTT notation list.

Notation	Meaning
DCC	Dummy Cloud Communication
VCM	Virtual Cloud Machine
CDCU	Centralized Data Collection Unit
RUT	Resource Utilization Trust
d	Clock count
n	Number of VCM
H	Hidden layers of neural etwork
TI	Tversky Value
E	Error
NN	Neural Network
W	Weight between neurons

Virtual Machine: In [8], paper has fetch machines utilization status for the learning of node behavior. Based on that this work has also considered some of basic features of machine like utilization of communication bandwidth, processor and memory for task completion. As the over utilization of machine beyond some maximum limit may leads to abnormal behavior of the system. This over utilization needs to learn for generating a alarm. Development of virtual machine , maximum utilization limit, etc is done in this step of the model.

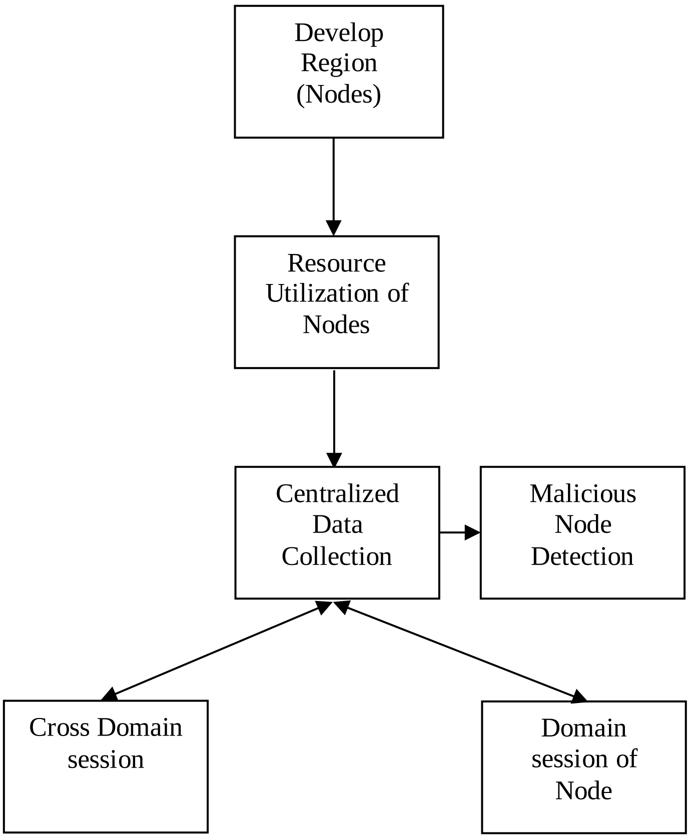
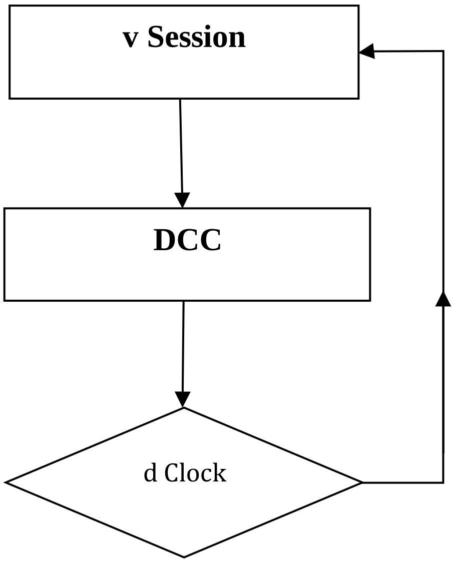


Fig. 1 Nodes data collection centrally CNTT.

Dummy Cloud Communication: In order to increase learning about the nodes status the sessions were consider for evaluating trust. This session on real communicate may lead to data / privacy/ resource losses. Hence paper has worked to learn behavior in dummy environment of cloud. This dummy environment is initiate after a fix number of duration and for fix clock d count. Counting of successful session, unsuccessful session, resource utilization was done in this dummy cloud communication clocks.

Trust Evaluation

Cloud machine trusts were evaluate by the behavior done b machine with other machines using tversky and by resource utilization.



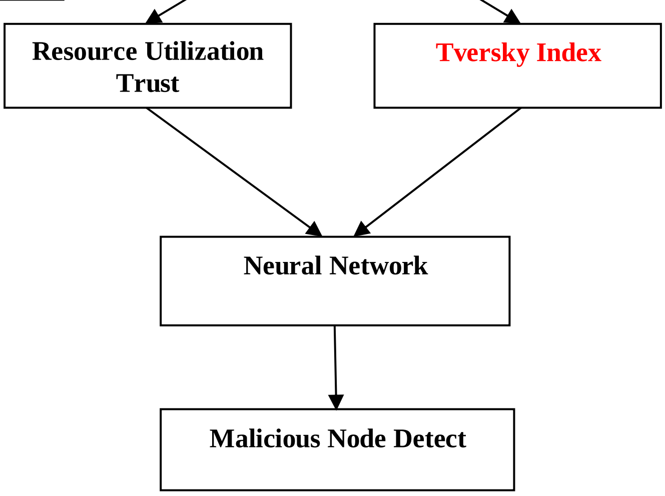


Fig.2 Proposed work training module.

Resource Utilization Trust

Virtual Cloud Machines VCM, were used by the model for improving the services quality. Each machine mention maximum utilization limit of different parameters like bandwidth, processor, memory [8]. Hence this centralized unit maintains a matrix of maximum resource utilization MRU for estimating resource utilization trust.

Trust value is estimated by two way first when no utilization was done and other was when utilization of resources were crossing maximum limit. In first case trust value is 1 for the VCM. Further trust value was the ratio of maximum limit to the resources utilization sum.

Tversky

The Tversky index is an asymmetric similarity measure on sets that compares a variant to a prototype [13]. For sets X and Y the Tversky index is a number between 0 and 1 given by [14]:

$$TI(X,Y)=|X\cap Y| / (|X\cap Y|+ \alpha|X-Y|+\beta|Y-X|)$$

where α , β >=0

Training of Neural Network

Neural network consider takes input training vector and desired output during training. For each set of training vector neuron weight value adjust for e number of epochs [15]. Tained neural network was directly used for predicting the virtual cloud machine status as malicious or normal.

In order to understand above steps let us consider an example where Wij have some weight values.

$$\text{Wij} = \begin{vmatrix} W_{11} & W_{12} & W_{13} \\ W_{21} & W_{21} & W_{21} \\ W_{31} & W_{31} & W_{31} \end{vmatrix}$$

Now this act as input $H1_{input}$ to next layer of hidden neurons. In this some biasing is also possible which was neglect in this example. So weight values of the neuron for next level is assumed as shown in below matrix.

$$W_{jk} = \begin{vmatrix} W_{11} & W_{12} & W_{13} \\ W_{21} & W_{21} & W_{21} \\ W_{31} & W_{31} & W_{31} \end{vmatrix}$$

Where each value obtained from the previous weight matrix multiplication is passed through the sigmoidal function. Therefore small variation in the output value was done by.

$$SoftMax = e^{O_{ij}} \sum_{k=1}^{j} e^{O_k}$$

The difference between the expected with obtained is consider as the error. This error need to be correct by adjusting the weight values of each layer. So here forward movement of the neural network is over and error back propagation starts.

In similar fashion other values can be calculate to find other set of derivatives.

$$\frac{\partial O_i}{\partial H_i} = \frac{\partial(e^{O_{ij}} \sum_{k=1}^{j} e^{O_k})}{\partial H} = \frac{e^{O_{H1}} \times (e^{O_{H2}} + e^{O_{H3}})}{e^{O_{H1}} + e^{O_{H2}} + e^{O_{H3}}}$$

$$\begin{vmatrix} \frac{\partial O_1}{\partial H_1} \\ \frac{\partial O_2}{\partial H_2} \\ \frac{\partial O_3}{\partial H_3} \end{vmatrix}$$

For each input to neuron let us calculate the derivative with respect to each weight. Now by using chain rule final derivates were calculated [16]. Here multiplication of each derivative was done in following way:

So overall ∂W_i can be obtained by getting value of weight from above equation, here all set of weight which need to be update are change.

- So error corresponds to the input data was estimate by differencing desired output obtain from output layer.

$$e_k(n) = d_k(n) - y_k(n)$$

- The ebpnn weight updation was done by above matix of ∂W_i $$w_{ij} = w_{ij} + \Delta w_{ij}$$
- So end of above iteration steps over when error obtained from the output layer get nearer to zero or some constant such as 0.001.

Malicious Node Detection

Trained neural network takes feature values from the centralized unit and generate status of the node. Neural network output value 1 means node is malicious and need to be removed from the network. While neural network output value 0 means node is working normally.

Proposed CMMTT Algorithm

```
Input: n // Number of VCM
Output: NN, M // NN: Neural Network, M: Malicious nodes
CB←Virtual_Machine(n)
Loop 1:d
    Loop 1:r // r: number of resources
        i←Rand()
        j←Rand()
        CDCU←Session(i, j)
    EndLoop
    RUT←Resource_Utilization_Trust(CDCU)
    Loop 1:n //n: number of machines
        TI←TverskyIndex (CB)
    EndLoop
    Loop 1:itr
        NN←Train_neural_Network(CDCU, TI, D)
    EndLoop
```

Detail steps of the proposed algorithm shows that after each tversky values were update and nodes which performed malicious activity in cloud are filtered and removed.

IV. EXPERIMENTS & RESULTS ANALYSIS

Implementation model was developed on MATLAB platform of 2016a version. Experimental values were compared on below parameters [8, 9]. Fewer than two environments first was no attack and other was DDoS attack.

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

$$Recall = \frac{TP}{TP + FN}$$

$$FMeasure = \frac{2 * Precision * Recall}{Precision + Recall}$$

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

Results

Comparison of proposed model was done with existing tenant cloud node malicious activity detection TMM proposed in [8].

Table 2 Malicious node detection model accuracy based comparison with different number of nodes.

Cloud Nodes		Methods	
Normal	Malicious	TMM [8]	CMMTT
60	5	88.89	88.24
90	5	90	89.41
90	10	88.89	88.24
120	10	91.67	93.04
100	0	100	100

Table 2 shows virtual cloud machine malicious node detection accuracy in different number of nodes environment. It was found that proposed model has improved the detection accuracy by the use of tversky method. Further it was found that in ideal condition no false alarm was generate by the model, hence 100% accuracy of normal node detection.

Table 3 Malicious node detection model Recall based comparison with different number of nodes.

Cloud Nodes		Methods	
Normal	Malicious	TMM [8]	CMMTT
60	5	1	1
90	5	1	1
90	10	1	1
120	10	0.8889	1
100	0	1	1

Recall values shown in table 3 for malicious virtual node detection in cloud environment. It was obtained that proposed CMMTT model has 100% true node detection accuracy in all set of conditions applied for testing/

Table 4 Malicious node detection model precision based comparison with different number of nodes.

Cloud Nodes		Methods	
Normal	Malicious	TMM [8]	CMMTT
60	5	0.889	0.8824
90	5	0.1	0.8941
90	10	0.5652	0.8824
120	10	0.6667	0.9304
100	0	1	1

Table 4 is precision value comparison of mode status detection in cloud environment. It was obtained that CMMTT model has improved the precision value by 29.8% as compared to TMM model proposed in [8]. This enhancement was archived by the use of tversky trust value evaluation in dummy cloud communication.

Table 5 Malicious node detection model F-measure based comparison with different number of nodes.

Cloud Nodes		Methods	
Normal	Malicious	TMM [8]	CMMTT
60	5	0.6154	0.9375
90	5	0.1818	0.9441
90	10	0.7222	0.9375
120	10	0.7619	0.964
100	0	1	1

Table 5 shows virtual cloud machine malicious node detection f-measure in different number of nodes environment. It was found that proposed model has improved the detection f-measure by the use of tversky method. Further it was found that in ideal condition no false alarm was generate by the model, hence 100% accuracy of normal node detection.

Table 5 Malicious node detection model execution time based comparison with different number of nodes.

Cloud Nodes		Methods	
Normal	Malicious	TMM [8]	CMMTT
60	5	43.6785	0.4285
90	5	33.8605	0.3817
90	10	42.0724	0.4998
120	10	39.9207	0.8284
100	0	42.85	0.5724

Use of neural network for nodes status detection has reduced the prediction time as compared to previous model. Estimation of feature tversky is also less time consuming.

V. Conclusions

Role of software increases in digital world hence dependency on the cloud SAAS is indirectly increases. This dependency attracts attacker to harm cloud environment. In order to detect such malicious machines of cloud this paper has proposed a model CMMTT. This model extract virtual machines features as per the utilization and its behavior in dummy cloud communication. Extracted features were used to predict the status of the node either malicious or normal. Experiment was done on different environment of the cloud and it was found that proposed model has improved the work precision value by 29.81% and f-measure value by 31.39%. In future scholar can perform this work in other type of attack detection for enhancing the security of cloud.

References

1. Xiao, S., Ge, X., Han, Q.-L., Zhang, Y.: Secure distributed adaptive platooning control of automated vehicles over vehicular Ad-Hoc networks under Denial-of-service attacks. IEEE Trans.

2. Alsarhan, A.; Al-Ghuwairi, A.R.; Alshdaifat, E.; Idhaim, H. A Novel Scheme for Malicious Nodes Detection in Cloud Markets Based on Fuzzy Logic Technique. Int. J. Interact. Mob. Technol. 2022, 16, 136–150.

3. Wang, Y.; Ding, Y.; Wu, Q.; Wei, Y.; Qin, B.; Wang, H. Privacy-Preserving Cloud-Based Road Condition Monitoring with Source Authentication in VANETs. IEEE Trans. Inf. Forensics Secur. 2019, 14, 1779–1790

4. 4 Trends Impacting Cloud Adoption in 2020. Available online: https://www.gartner.com/smarterwithgartner/4-trendsimpacting-cloud-adoption-in-2020/ (accessed on 8 August 2020).

5. Chen, F.; Luo, D.; Xiang, T.; Chen, P.; Fan, J.; Truong, H.L. IoT Cloud Security Review: A Case Study Approach Using Emerging Consumer-oriented Applications. ACM Comput. Surv. (CSUR) 2021, 54, 1–36.

6. Giri, S.; Shakya, S. Cloud Computing and Data Security Challenges: A Nepal Case. Int. J. Eng. Trends Technol. 2019, 67, 146–150.

7. S. Udhaya Chandrika . “Modified ECC for Secure Data Transfer in MultiTenant Cloud Computing”. I. J. Computer Network and Information Security, 2022, 6, 76-88.

8. O. A. Wahab, J. Bentahar, H. Otrok and A. Mourad, "Optimal Load Distribution for the Detection of VM-Based DDoS Attacks in the Cloud," in IEEE Transactions on Services Computing, vol. 13, no. 1, pp. 114-129, 1 Jan.-Feb. 2020

9. Y. Miao, Q. Tong, R. H. Deng, K. -K. R. Choo, X. Liu and H. Li, "Verifiable Searchable Encryption Framework Against Insider Keyword-Guessing Attack in Cloud Storage," in IEEE Transactions on Cloud Computing, vol. 10, no. 2, pp. 835-848, 1 April-June 2022

10. J. Zhang, T. Li, Z. Ying and J. Ma, "Trust-Based Secure Multi-Cloud Collaboration Framework in Cloud-Fog-Assisted IoT," in IEEE Transactions on Cloud Computing, doi: 10.1109/TCC.2022.3147226.

11. P. Zhang, M. Zhou and Y. Kong, "A Double-Blind Anonymous Evaluation-Based Trust Model in Cloud Computing Environments,"

in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 51, no. 3, pp. 1805-1816, March 2021.

12. Peiyun Zhang, *Senior Member, IEEE*, Yang Kong, And Mengchu Zhou. “A Domain Partition-Based Trust Model For Unreliable Clouds”. IEEE Transactions On Information Forensics And Security, VOL. 13, NO. 9, SEPTEMBER 2018.
13. Rahnama, J., Hüllermeier, E. (2020). Learning Tversky Similarity. In: , et al. Information Processing and Management of Uncertainty in Knowledge-Based Systems. IPMU 2020. Communications in Computer and Information Science, vol 1238. Springer, Cham.
14. Satish Chander, P. Vijaya, Roshan Fernandes, Anisha P Rodrigues, Maheswari R. "Dolphin-political optimized tversky index based feature selection in spark architecture for clustering big data" Advances in Engineering Software, Volume 176, 2023.
15. Latifi, N., Amiri, A. (2011). Partial and Random Updating Weights in Error Back Propagation Algorithm. In: Pichappan, P., Ahmadi, H., Ariwa, E. (eds) Innovative Computing Technology. INCT 2011. Communications in Computer and Information Science, vol 241. Springer, Berlin, Heidelberg.
16. Purwa Hasan Putra Muhammad Zarlis, H Mawengkang. "Analysis Algorithm Kohonen and Momentum on the Back propagation Neural Network". International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878,Volume-7, Issue-6S5, April 2019.