Journal of
Data Acquisition and Processing

ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7765981

JACCARD TRUST AND CONVOLUTIONAL NEURAL NETWORK MODEL BASED CLOUD MALICIOUS NODE DETECTION

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

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

Abstract— Network requirement increases in different sector for various situation for ease of human like, research, etc. Dependency of machines are shift towards cloud but this brings many challenges of security. Out of many security issue this paper monitors node malicious activity for trust evaluation. In order to reduce false alarm social trust evaluation function Jaccard trust was used. Further node machines status were used to learn its activity as a feature. Obtained Jaccard trust and feature set used for training the convolutional neural network. Experiment was done on different number of nodes and attack environment. Trained deep learning model has efficiently detect the malicious node in the network. Result shows that proposed model has increased the true alarm precision, recall, accuracy comparing parameters as compared to existing models.

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

I. INTRODUCTION

Malicious node detection is a critical aspect of cloud security that aims to identify and mitigate attacks from malicious nodes within a network. A malicious node refers to a node that intentionally sends false, misleading, or harmful information to other nodes in the cloud network. These nodes can cause significant damage to the network infrastructure, compromise sensitive information, and disrupt normal network operations [1]. Detecting malicious nodes is a challenging task, especially in large-scale networks, due to the complexity of network structures and the large number of nodes involved.

There are several approaches to detect malicious nodes in a network, including statistical analysis, machine learning, and game theory-based techniques [2]. Statistical analysis is one of the most commonly used techniques for detecting malicious nodes. It involves monitoring network traffic and identifying anomalies in the data. Anomaly detection algorithms can identify nodes that behave differently from the normal nodes in the network. These algorithms use statistical models to detect nodes that have abnormal behavior, such as transmitting more data than usual or sending data to unexpected destinations. Machine learning techniques are another effective approach for detecting malicious nodes [3]. These algorithms use historical data to train models that can classify nodes as either normal or malicious. The models are then used to analyze real-time network traffic and detect any anomalies. Machine learning algorithms can detect complex attacks that are difficult to identify using traditional signature-based approaches.

Some techniques use mathematical models to detect malicious nodes that are trying to manipulate the network for their own benefit [4]. These techniques analyze the incentives and behavior of nodes in the network and detect any abnormal behavior that indicates malicious intent. In conclusion, detecting malicious nodes is a critical aspect of network security that requires advanced techniques and algorithms. Network administrators must stay up-to-date with the latest techniques and tools to effectively detect and mitigate malicious nodes within their networks. By implementing these techniques, they can improve the security and reliability of their network infrastructure and prevent potential damage caused by malicious nodes [5].

II. Related Work

The Graph Structural-topic Similar Subgraph Merging, also known as GraphSTSGM, is a method we suggest using to extract the topological similarity between nodes. Y. Hao et al. increased the performance of malicious device identification in [6]. Graph STSGM adds local neighbourhood structural patterns to Graph Structural-topic Neural Network to extract approximative local symmetry characteristics (GraphSTONE). In this method, we construct a device-account connection graph G with devices and accounts as nodes, create an edge between related devices and accounts, and then use the merge-similar-substructures-based anonymized walk in G to estimate the approximately local symmetry of each device.

An algorithm for recognising fake recommender-data was put forth by Gianni D'Angelo et al. in [7]. We deliberately pay more attention to recommender-data than to recommenders. This is due to the possibility that some recommenders might only offer recommender-data providing a small (but restricted) set of updated information. This is a trick employed by dishonest recommenders to avoid being found. The suggested technique employs association rules to define a reliability ranking of the decision support called reputation rank, which is a confidence-based measure.

In [8], K. Gu et al. suggested a method for detecting malicious nodes in fog technology VANETs, where the fog server scores each suspect node based on the relationship between network structure and data it has collected. According to the association between the outlier detection of collected data and the influence of nodes, we construct a reputation mechanism in our suggested system to rate each suspicious node.

P. Zhang et al. suggested using checking nodes in [9] to help identify such behaviour. Then, for providers who want to engage in provider-user collusion deception, it does gain-loss analysis. The suggested trust value can be used to spot collusion and deceitful behaviour and enable policymakers to establish appropriate penalties for bad actors. As a result, in public cloud computing systems, provider-initiated collusion deceptive conduct can be strongly prohibited.

According to R. I. Minu et alproposed .'s Edge-based strategy in [10], EBAD was created as a strong identity theft and abuse prevention mechanism for smart city environments. The likelihood of carrying out the Sybil assault over a Cooperative conning attempt will be eliminated thanks to EBAD's efficiency in identifying the Sybil assault nodes (SA-CBA). To evaluate the harmful activity of the network entities, EBAD employs an Edge-based accusations analysis approach. To lessen the computational stress on the end devices, the majority of the necessary computations have been moved to the edge node. The effectiveness of EBAD has also been tested in a malicious setting.

A reliable de-swinging k-anonymity approach for location privacy protection is put forth by Manxiang Yang et al. in [11]. The de-swinging reputation evaluation technique (DREM), which creates a penalty element to prevent swinging behaviour, is what we first introduce. The entity honesty degree, based system entropy, and service shift degree are used in this method to calculate reputation. Also, based on our suggested DREM, a reliable cloaking area is built to safeguard the requester's location privacy. Nodes in the region have the option of picking some nodes who have a solid reputation for finishing the anonymous domain creation. Lastly, we create reputation contracts that use smart contracts to automatically calculate credit.

III. Proposed Methodology

This section detailed the proposed Jaccard Trust and Convolutional Neural Network based Malicious Node Detection JTCNN-MND model. Fig. 1 shows the various steps of node behavior data collection. Fig. 2 shows JTCNN-MND blocks. Table 1 is list of notations used in the work for the explanation of Jaccard and CNN uses.

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

JT	Jaccard Trust
CNN	Convolutional Neural Network

Virtual Machine: In [12], the paper describes the use status of fetch machines for the study of node activity. Based on that, our work has also taken into account some fundamental aspects of the machine, such as how the processor, memory, and communication bandwidth are used to complete tasks. Since the system may behave abnormally if a machine is used excessively or beyond a certain capacity. It is necessary to learn how to create an alarm when there is an overuse. In this stage of the model, the virtual machine, the maximum utilisation cap, etc. are developed.

Dummy Cloud Communication: The sessions were taken into consideration while determining trust in order to boost learning about the nodes' status. This real-time communication session could result in data, privacy, or resource losses. So, study has tried to learn behaviour in a cloud-based fake environment. This fake environment starts up after a predetermined amount of time and for a predetermined clock d count. These dummy cloud communication clocks were used to count the number of successful sessions, unsuccessful sessions, and the amount of resources used.

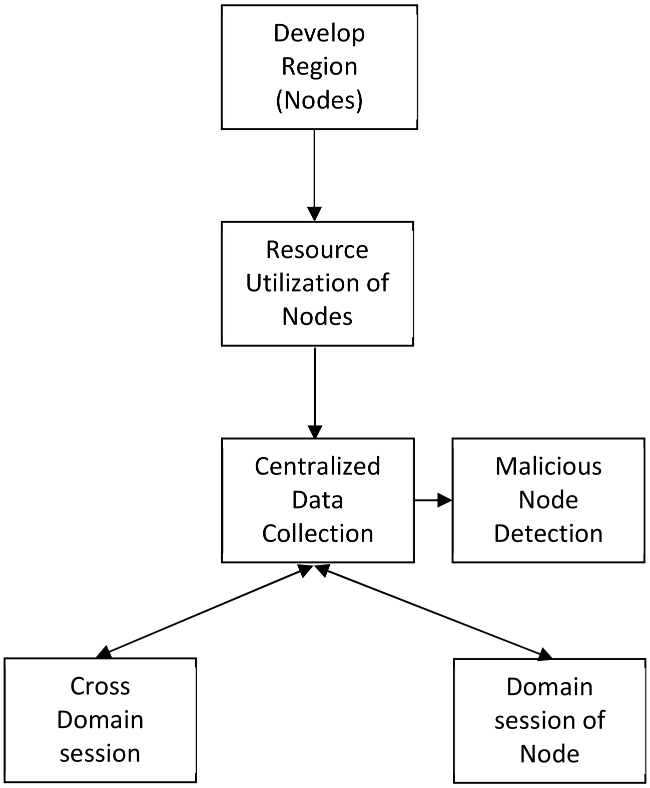

Fig. 1 Nodes data collection centrally CNTT.

Trust Assessment

Resource usage and machine interaction with other machines when employing the Jaccard algorithm were used to evaluate cloud machine trustworthiness.

Node Trust value

Every cloud node has a trust value that ranges from 0 to 1. This value may rise or fall depending on how the nodes behave in terms of transaction success. In order to assess the usefulness of this work, cloud bridge storage tables were used. The first step involved calculating the node direct trust value by dividing the total number of transactions by the number of successful transactions. Hence, let Tsij stand for the total number of transactions and Ttij for the number of successful transactions between nodes I and j, respectively [13]. The direct trust value was estimated using Equ.

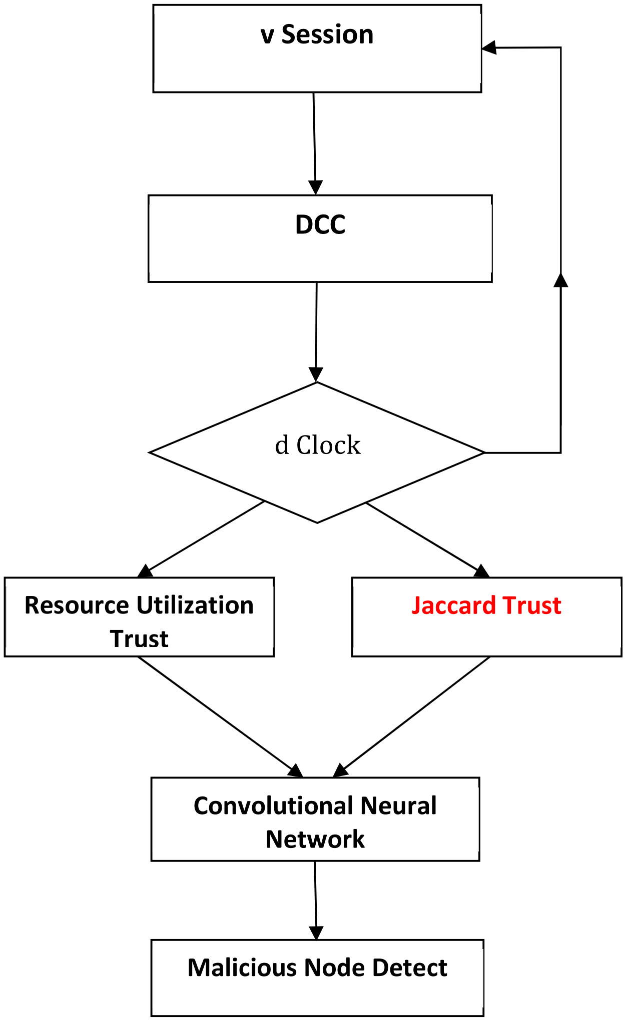

Fig.2 Proposed JTCNN-MND work training module.

$$D_{ij} = \frac{Ts_{ij}}{Tt_{ij}}$$

The direct trust values for each node are given in the eq. 1 above, although there is room for variation in node to node interactions. As malevolent nodes give certain nodes good service while giving others poor service. In order to further process this trust value, the Jaccard coefficient function is required. This function takes all of a node's direct trust values and generates a single node value based on various beahvious interactions that node has with other nodes.

Coefficient of Jaccard

A trust value was generated by Eq. using the Jaccard function, which was first introduced by [16], based on several attributes between two elements.

$$JC = \frac{A \cap B}{A \cup B}$$

In the equation above, A and B are cloud nodes, and features represent a direct measure of trust between them. By obtaining a lower direct trust value between A and B for identical nodes like AC and BC, AB is obtained. By obtaining a greater direct trust value between A and B for identical nodes like AC and BC, AB is obtained.

Resource Utilization Trust

The model uses virtual cloud machines, or VCMs, to raise the calibre of the services. Each device specifies the maximum capacity for many factors, including bandwidth, CPU, and memory [14, 15]. In order to estimate resource utilisation trust, this central unit maintains a matrix of highest resource utilisation, or MRU. Trust value is calculated in two ways: first, when no use is taking place, and second, when resource utilisation is over the maximum level. In the first scenario, the VCM's trust value is 1. The ratio of the upper limit to the total amount of resources used added additional trust value.

Malicious Node Detection

The state of the node is generated by a trained convolutional neural network using feature values from the centralised unit. Value 1 on the output of a convolutional neural network indicates a malicious node that needs to be removed from the network. CNN output value 0 denotes a node that is operating normally.

Proposed JTCNN-MND **Algorithm**

```
Input: n // Number of VCM
Output: CNN, M // CNN: Convolutional Neural Network
VCM←Virtual_Cloud_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
JT←Jaccard_Trust (CB)
EndLoop
    Loop 1:itr
    CNN←Train_CNN(CDCU, JT, RUT)
    EndLoop
```

The proposed algorithm's detailed stages demonstrate that nodes that engaged in harmful activity in the cloud are sorted and eliminated after each update to the jaccard values.

IV. EXPERIMENTS & RESULTS ANALYSIS

The implementation model was created using the 2016a version of MATLAB. For the following parameters, experimental values were contrasted [8, 9]. The first environment had no attack, and the other had a 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 [15].

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

Cloud Nodes		Methods		
Normal	Malicious	TMM [15]	CMMTT [16]	JTCNN-MND
60	5	88.89	88.24	90.59

90	5	90	89.41	96.47
90	10	88.89	88.24	95.29
120	10	91.67	93.04	98.26
100	0	100	100	100

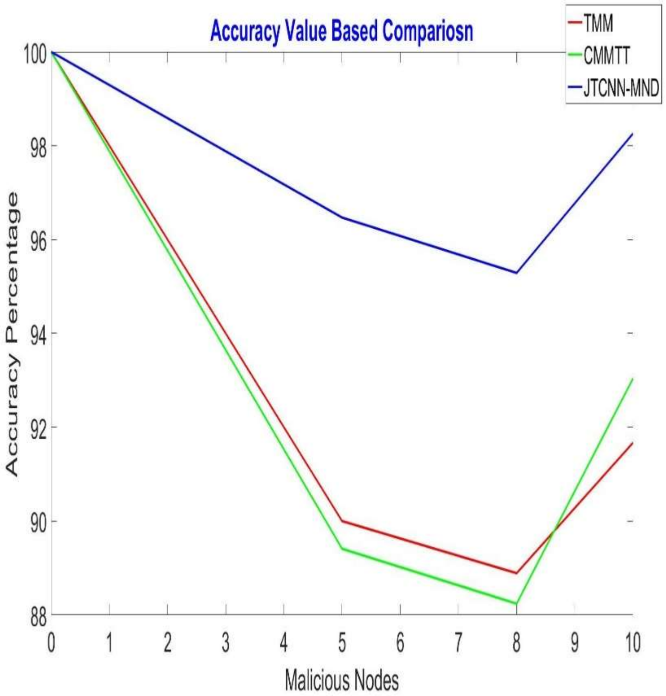

Fig. 3 Malicious node detection accuracy percentage value based comparison.

Accuracy of malicious node detection in cloud environment of proposed model JTCNN-MND is high as compared to work proposed in [15, 16]. It was found that use of social trust has high accuracy of detection and low false alarm rate in attack/ideal environment. Further table 2 shows that proposed JTCNN-MND has improved the accuracy by 4.336% as compared to TMM and 4.232% as compared to CMMTT.

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

Cloud Nodes		Methods		
Normal	Malicious	TMM [15]	CMMTT	JTCNN-MND
60	5	1	1	0.9467
90	5	1	1	0.974

90	10	1	1	0.9494
120	10	0.8889	1	0.9817
100	0	1	1	1

Recall values for hostile virtual node identification in cloud environments are presented in table 3. It was discovered that the suggested CMMTT model had 100% genuine node detection capability under all testing scenario. Proposed JTCNN-MND has almost match CMTT recall values as both uses social trust models.

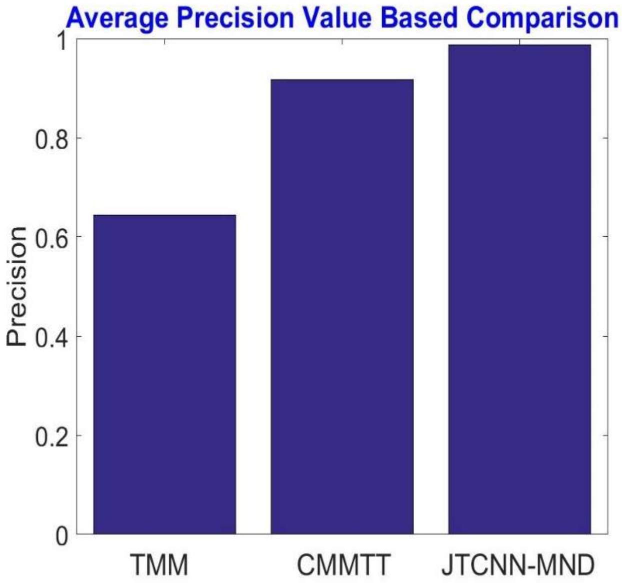

Fig. 4 Average precision percentage value based comparison.

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

Cloud Nodes		Methods		
Normal	Malicious	TMM [15]	CMMTT	JTCNN-MND
60	5	0.889	0.8824	0.9467
90	5	0.1	0.8941	0.9868
90	10	0.5652	0.8824	1
120	10	0.6667	0.9304	1
100	0	1	1	1

It was obtained that use of learning model and social trust evaluation in cloud node behavior analysis has increased the work performance in CMMTT and JTCNN-MND. Table 4 shows that porposed model precision values is high in all conditions with each comparing models. So true positive rate of the JTCNN-MND is efficient by use of CNN in the work.

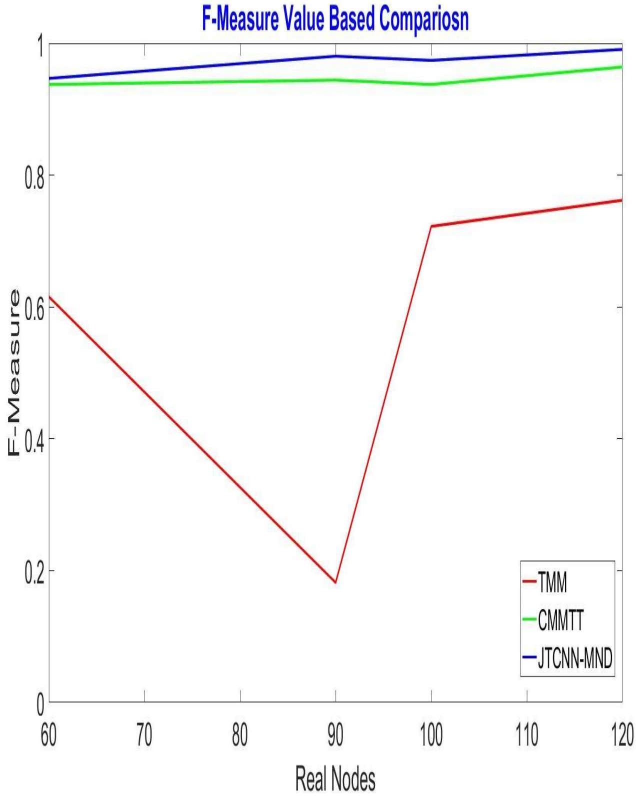

Fig. 5 Real node variation based f-measure value based comparison.

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

Cloud Nodes		Methods		
Normal	Malicious	TMM [15]	CMMTT	JTCNN-MND
60	5	0.6154	0.9375	0.9467
90	5	0.1818	0.9441	0.9804
90	10	0.7222	0.9375	0.974
120	10	0.7619	0.964	0.9907
100	0	1	1	1

Proposed JTCNN-MND model inverse average of precision and recall values were shown in table 4. It was obtained that use of CNN model for the learning of node feature and trust value has improved the detection accuracy of the work. Further table 2 shows that proposed JTCNN-MND has improved the f-measure value by 0.3321 as compared to TMM and 0.0217 as compared to CMMTT.

V. Conclusions

Cloud based work is new trend in today organizations. Hence its security requirement is also very high. This paper has increases the cloud security by collecting the node behavior using social Jaccard Trust evaluation. Apart from trust paper has collect few information of nodes like utilization of memory, processor corresponding to its limits. Extracted features were used to train convolutional neural network model. It was found that use of trust base malicious node detection was high. Experiment was done on different environment of virtual machine. Result shows that proposed JTCNN-MND has improved the accuracy by by 4.336% as compared to TMM and 4.232% as compared to CMMTT.

References

1. Ray and I. Ray, “Trust-based access control for secure cloud computing,” High-Performance Cloud Auditing and Applications, Springer, New York, NY, USA, pp. 189–213, 2014.
2. Rakesh Jha, Asst. Prof. Sumit Sharma. "Moth Flame Based Feature Selection for Ransom Ware Detection Training Model". Volume 9 issue 6, ijset.in, International Journal of Science, Engineering and Technology, 2021.
3. J. A. Lochhead, R. Rowland, R. Stephen, and J. Johnson, “Manikandan Subramanian, and Emmanuel Kothapally. “System and method for customer provisioning in a utility computing platform,” vol. 8, p. 484, 2013.
4. K. Nagarajan, A. Rajagopalan, S. Angalaeswari, L. Natrayan, and W. D. Mammo, “Combined economic emission dispatch of microgrid with the incorporation of renewable energy sources using improved mayfly optimization algorithm,” Computational Intelligence and Neuroscience, vol. 2022, Article ID 6461690, 22 pages, 2022.
5. Rani Danavath, Asst. Prof. Dr. V. B. Narsimha. "Load Balancing in Cloud Computing Through Multiple Gateways". International Journal of Scientific Research & Engineering Trends, ijsret.com Volume 8, Issue 6, Nov-Dec-2022.
6. Y. Hao, Q. Lu and X. Chen, "A Graph Representation Learning Algorithm for Approximate Local Symmetry Feature Extraction to Enhance Malicious Device Detection Preprocessing," in IEEE Access, vol. 10, pp. 53418-53432, 2022.
7. Gianni D’Angelo, Francesco Palmieri, Salvatore Rampone, Detecting unfair recommendations in trust-based pervasive environments, Information Sciences, Volume 486, 2019, Pages 31-51.
8. K. Gu, X. Dong and W. Jia, "Malicious Node Detection Scheme Based on Correlation of Data and Network Topology in Fog Computing-Based VANETs," in IEEE Transactions on Cloud Computing, vol. 10, no. 2, pp. 1215-1232, 1 April-June 2022.

9. 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.

10. R. I. Minu, G. Nagarajan, A. Munshi, K. Venkatachalam, W. Almukadi and M. Abouhawwash, "An Edge Based Attack Detection Model (EBAD) for Increasing the Trustworthiness in IoT Enabled Smart City Environment," in IEEE Access, vol. 10, pp. 89499-89508, 2022.

11. Manxiang Yang , Baopeng Ye , Yuling Chen , Tao Li , Yixian Yang , Xiaobin Qian , Xiaomei Yu . “A trusted de-swinging k-anonymity scheme for location privacy protection”. Journal of Cloud Computing: Advances, Systems and ApplicationsVolume 11Issue 1Sep 2022.

12. 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.

13. Sucheta Soundarajan , John Hopcroft. "Using community information to improve the precision of link prediction methods". Proceedings of the 21st International Conference on World Wide WebApril 2012

14. 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.

15. 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.

16. Pragya Richhariya, Dr. Shailja Sharma. "Cloud Malicious Node Detection and Resource Management by Tversky Trust". volume 11 issue 3, 2022.