Intelligent Automation & Soft Computing

Tech Science Press

DOI: 10.32604/iasc.2023.039814
Article

Real-Time Multi Fractal Trust Evaluation Model for Efficient Intrusion Detection in Cloud

S. Priya[1] and R. S. Ponmagal[2,*]

[1]Department of Computer Science and Engineering, School of Computing, SRM Institute of Science and Technology, Chengapattu, 603 203, Tamil Nadu, India
[2]Department of Computing Technologies, School of Computing, SRM Institute of Science and Technology, Chengapattu, 603 203, Tamil Nadu, India

*Corresponding Author: R. S. Ponmagal. Email: ponmagas@srmist.edu.in

Received: 18 February 2023; Accepted: 14 April 2023; Published: 23 June 2023

Abstract: Handling service access in a cloud environment has been identified as a critical challenge in the modern internet world due to the increased rate of intrusion attacks. To address such threats towards cloud services, numerous techniques exist that mitigate the service threats according to different metrics. The rule-based approaches are unsuitable for new threats, whereas trust-based systems estimate trust value based on behavior, flow, and other features. However, the methods suffer from mitigating intrusion attacks at a higher rate. This article presents a novel Multi Fractal Trust Evaluation Model (MFTEM) to overcome these deficiencies. The method involves analyzing service growth, network growth, and quality of service growth. The process estimates the user's trust in various ways and the support of the user in achieving higher service performance by calculating Trusted Service Support (TSS). Also, the user's trust in supporting network stream by computing Trusted Network Support (TNS). Similarly, the user's trust in achieving higher throughput is analyzed by computing Trusted QoS Support (TQS). Using all these measures, the method adds the Trust User Score (TUS) value to decide on the clearance of user requests. The proposed MFTEM model improves intrusion detection accuracy with higher performance.

Keywords: Intrusion detection; cloud systems; trusted service support; trusted network support; trust user score; trusted QoS support

1 Introduction

The cloud environment has a more significant influence on modern service orient architecture as it supports various service platforms. The cloud service provider provides access to multiple services independent of the platform metrics. This encourages organizations to maintain their data over the cloud and allows access to their employees and clients. Like any other environment, the cloud also faces various threats that target the overall performance of the cloud. Multiple users of the environment access the services, some of which are involved in intrusion attacks.

The intrusion attack is a threat produced by registered and unregistered environment users. The attack intends to steal information and intrude into the environment to get access to the data in the cloud. Behind the scenes, an attack can be detected in several ways. For example, the key-based approaches are used in earlier days to verify the legitimacy of the user. Malicious access is also detected according to rule-based, profile-based, activity-based, and trust-orient approaches. All the above-discussed methods have their metric and produce different results in mitigating the intrusion attack. The rule-based approach uses the defined rules in the monitoring of intrusion attacks.

Similarly, the profile-based approach verifies the profile for the user's presence in the list. In contrast, the activity-based methods monitor the user's behavior in accessing the service. Also, the trust-based approach estimates different trust measures in detecting intrusion attacks. All the above-discussed techniques have their merit and demerit. But, the methods are unsuitable for various sources of dynamic threats as the behavior of the registered users would be different at some point and needs to be monitored for the presence of an attack.

In terms of service in the environment, the service throughput must be higher. To achieve higher throughput for any service, the user who accesses the service must behave adequately and should be supportive of the service. So, the user must be measured for the user's trust in the service. Similarly, the user accesses the network in accessing the service. Such resources must be accessed in a trusted way, significantly impacting the network's overall performance. Also, the user support for achieving higher Quality of Service (QoS) performance must be evaluated to mitigate the intrusion attack. By considering all these considerations, the performance of intrusion detection and achieving higher performance can be improved.

With all these considerations, an efficient multi-fractal trust evaluation model (MFTEM) is presented in this article. The method analyzes the user's trust in how he supports the service, network, and throughput. Measuring the faith in this way helps the model achieve higher performance in all the diagonals.

2 Related Works

Against intrusion attacks, there are several mitigation techniques designed in literature. This section pinpoints a set of methods around the problem.

A deep blockchain framework (DBF) is presented in [1] towards distributed intrusion detection with blockchain technique. The method uses smart contracts with IoT networks. The process uses bidirectional long short-term memory (BiLSTM) with deep learning to perform intrusion detection. To secure cloud servers from different threats, an event-based model is presented in [2], which monitors the attack rate of the entire network to perform intrusion detection. An Auto encoder-based intrusion detection system with a deep neural network is presented in [3], which uses conditional denoising adversarial autoencoder (CDAAE) to generate specific types of malicious samples. The second model (CDAEE-KNN) is a hybrid of CDAAE and the K-nearest neighbor algorithm used in detecting the attack. A fog-enabled scheme is presented for intrusion detection in [4], which uses a multi-objective optimization model in detecting intrusion attacks with a genetic algorithm. The method considers energy consumption and execution time in the classification. A trusted virtual intrusion detection system (TVIDS) is presented in [5], which safeguards sensitive information according to maintained policies.

A fuzzy rough set-based intrusion detection scheme is presented in [6], which performs feature selection with a rough set and trains the (Convolution Neural Network) CNN to perform intrusion detection. A deep learning model is proposed to secure Dew Computing as a Service (DaaS) in EoT systems [7]. An AI-based approach is presented in [8] towards intrusion detection. The method uses a Transient search optimization (TSO) algorithm for feature selection, and uses differential evolution (TSODE) is used for classification. A high-precision Intrusion Detection Classification Model (IDCM) is presented in [9], which generates a k-dependency Bayesian network (KDBN) structural model toward classification. Adaptive Virtual Machine (AVM) Shield towards security is presented in [10], which uses system call features with an n-gram approach to perform classification with binary particle swarm optimization.

A deep forensic-based deep learning model (Deep-IFS) is presented in [11], which monitors the traffic and performs the classification of fog nodes. An auto-encoder-based deep learning model is presented in [12], which analyzes the network traffic and classifies them using Deep Neural Network (DNN). Reliable event-based anomaly detection for IoT is shown in [13], which monitors the events in the network, and based on that, the method performs intrusion detection. In [14], the scattered denial-of-service mitigation tree architecture (SDMTA) is used to detect intrusion attacks.

In [15], a Dew-Cloud-based model is designed to enable hierarchical federated learning (HFL). The hierarchical long-term memory (HLSTM) model is deployed at distributed Dew servers with a backend supported by cloud computing. A multi-tenant intrusion detection framework as a service for SaaS (MTIDaaS) is presented in [16] for detecting intrusion attacks against cloud services. A correlation-based feature selection (ECOFS) approach is proposed in [17], which removes redundant and irrelevant features using various eco features to detect intrusion attacks.

A mobile-based approach is presented in [18], which uses two different recovery stages to restore the integrity of mobile applications. An ensemble multi-binary attack model (EMBAM) is shown in [19] for intrusion detection. The method generates behavior features in developing the ensembles to perform classification. A weighted class classification model is presented in [20], combining machine learning and node details to classify attacks.

A swarm neural network-based approach is presented in [21], which identifies the attacker in the network with an edge-centric (Internet of Mobile Terminals) IoMT framework. A behavior analysis model is presented in [22], which considers the time-variant traffic with two-layer random fields to perform classification. The method monitors the events and performs classification with behavior patterns learned.

EvolCostDeep, a hybrid model of stacked auto encoders (SAE), is presented in [23], which generates a deep learning model and performs classification with data received. In [24] shows the Multi branch Reconstruction Error (MbRE) Intrusion Detection System (IDS) for edge-based anomaly detection in VANETs, which classifies each branch of a sequence as 0/1 based on the reconstruction error threshold. In [25], an ML-based IDS is presented, which uses three in-sequence tasks, pre-processing, binary detection, and multi-class detection, with a multi-tier architecture with one-, two-, and three-tier architectural configurations. We then mapped three in-sequence tasks into these architectures, assigning ten tasks.

A cluster-based intrusion detection model is presented in [26], which groups similar traces of various intrusion attacks, and based on that, the method classifies the incoming traffic to perform intrusion detection. In [27], the service availability of based modeling is presented towards intrusion detection in a cloud environment.

An optimized model of the node isolation technique is presented in [28] to support the Mobile ad-hoc networks, which consider the behavior of the nodes in isolating malicious nodes. An efficient multi-level threshold-based clustering algorithm is presented in [29] for detecting intrusion attacks. Finally, a time-variant predicate-based traffic approximation algorithm is presented in [30], which measures the traffic in different time stamps and performs intrusion detection accordingly.

All the above-discussed approaches need better performance in detecting and mitigating intrusion attacks in a cloud environment. According to the related works, it is noticed that the accuracy of intrusion detection greatly depends on various factors. In order to achieve higher performance in intrusion detection, it is necessary to analyze the trust of a user towards service growth, analyze the user towards network growth, and necessary to analyze the QoS growth. By analyzing the user trust towards various growths, the performance of intrusion detection can be improved. This research is focused on designing such an intrusion detection model for the growth of the cloud environment.

3 Multi Fractal Trust Evaluations Model-Based Intrusion Detection

The multi-fractal trust evaluation model performs intrusion detection according to different metrics. First, the method reads the traces of previous access belonging to the user. With the user traces collected, the process analyzes the user support on Service Growth, Network Growth, and Throughput Growth. Each analysis involves measuring specific support measures to support intrusion detection. The method computes Trusted Service Support (TSS). Also, the trust of the user in keeping the network stream by computing Trusted Network Support (TNS). Similarly, the user's trust in achieving a higher quality of service is analyzed by computing Trusted Throughput Support (TTS). Using all these measures, the method adds the Trust User Score (TUS) value to decide on the clearance of user requests. The detailed approach is discussed in this part.

The architecture of the proposed MFTEM model has been presented in Fig. 1, where the model's components have been presented in this section.

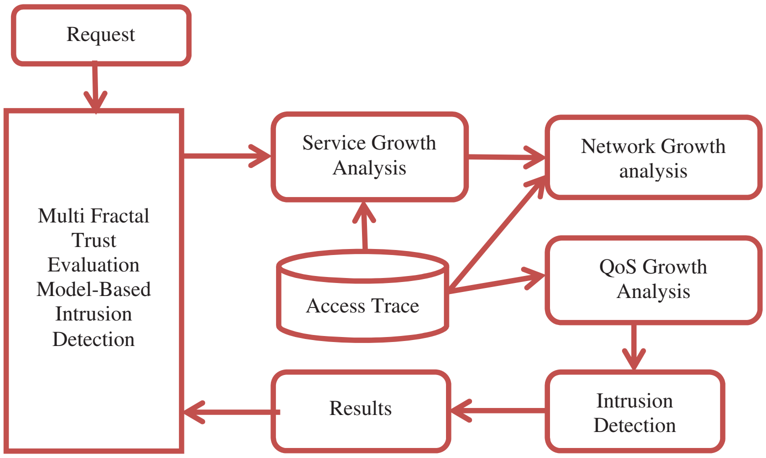

Figure 1: Architecture of proposed MFTEM intrusion detection model

3.1 Service Growth Analysis

The service growth analysis is the process of analyzing the support of users towards service growth. The service's growth will improve when the user genuinely accesses the service. If the user claims the service and interrupts its function in an intermediate way, then its performance gets affected. Measuring the user's trust in service growth is necessary before allowing the user to access the service.

The service growth analysis is performed by computing the frequency of service being accessed by the user and the frequency of other services being accessed. The method calculates the number of similar services available with frequency measures, and the specific service is selected several times. Also, the process computes the value of success frequency for service s and other services s2. Using these measures, the method computes the value of Trusted Service Support (TSS).

Algorithm:

```
Given: Service Traces ST, User ID Uid, Service Taxonomy STax, Service Id Sid.
Obtain: TSS.
Start
    Read ST, STax, Sid, and Uid.
    Find user traces$UTs = \bigcup_{i=1}^{size(ST)} ST\left(i\right).User == Uid$ (1)
    Find Similar services $Sset = \bigcup_{i=1}^{size(STax)} STax\left(i\right).Type == Sid.Type$ (2)
    Compute No of Service access $NAccess = Count\left(\sum_{i=1}^{size(UTs)} UTS\left(i\right).service == Sid\right)$ (3)
    Compute Service Access Rate $\mathrm{SAR} = \frac{NAccess}{\sum_{\mathrm{i=1}}^{\mathrm{size(UTS)}} \mathrm{UTS\ (i)}\ .ServiceId == \mathrm{Sid}}$ (4)
    Compute Other Service Access Rate $\mathrm{OSAR} = \frac{\sum_{\mathrm{i=1}}^{\mathrm{size(UTS)}} \mathrm{UTS\ (i)}\ .ServiceId! = \mathrm{Sid}}{\mathrm{Size(UTS)}}$ (5)
    Compute Success Frequency $\mathrm{SFr} = \frac{\sum_{\mathrm{i=1}}^{\mathrm{size(UTS)}} \mathrm{UTS\ (i)}\ .State == Success\&\&\mathrm{UTS\ (i)}\ .Service == \mathrm{Sid}}{\sum_{\mathrm{i=1}}^{\mathrm{size(UTS)}} \mathrm{UTS\ (i)}\ .Service == \mathrm{Sid}}$ (6)
    Compute $\mathrm{TSS} = (\mathrm{SAR} \times \mathrm{OSAR}) \times \frac{1}{\mathrm{SFr}}$ (7)
Stop
```

The service growth analysis algorithm measures the trust of the user in support of service growth. It has been estimated by computing the service frequency as the service access rate, the frequency of accessing other similar services, and the success frequency of the service. Using all these measures, the method computes the value of TSS to perform intrusion detection.

3.2 Network Growth Analysis

The performance of a cloud environment greatly depends on how efficiently the network is being accessed. It is necessary to utilize the network resources efficiently, which directly affects the performance of the entire cloud environment. Whenever the user requests access to network resources, it is necessary to measure their trust before allowing the user to access the network resources. The method performs network growth analysis to measure the user's trust in accessing network resources. The network growth analysis is the process of analyzing the support of users in maximizing network utilization in a trusted way. It has been studied by measuring bandwidth utilization support (BUS) and data rate support (DRS). The value of BUS is measured by computing the number of service packets being sent by the user and the number of them has followed the protocol. The value of DRS is measured by computing the number of data packets sent by the user and the average data length of the packets. Also, the method uses the success ratio in measuring the value of Trusted Network Support (TNS) with all the above-measured values.

Algorithm:

```
Given: Service Traces ST, User ID Uid, Service Taxonomy STax, Service Id Sid.
Obtain: TNS
Start
        Read ST, Uid, Sid.
        Find user traces UTs = $\bigcup_{i=1}^{size(ST)} ST\,(i)\,.User == Uid$
        Compute Packet Rate Pr = $\sum_{\mathrm{i=1}}^{\mathrm{Size(UTs)}} Count(\mathrm{UTs\ (i)}\ .packet - count)$        (8)
        Compute Protocol Rate Prr = $\sum_{\mathrm{i=1}}^{\mathrm{Size(UTs)}} Count(\mathrm{UTs\ (i)}\ .protocol == \mathrm{Ok})$        (9)
        Compute BUS = $\frac{Prr}{Pr}$
        Compute average data length Adl = $\frac{\sum_{\mathrm{i=1}}^{\mathrm{Size(Uts)}} \mathrm{Uts\ (i)}\ .datalength}{\mathrm{Size(UTS)}}$        (10)
        Compute DRS = $\frac{\mathrm{Adl}}{\mathrm{Pr}}$        (11)
        Compute TNS = $\mathrm{BUS} \times \frac{1}{\mathrm{DAR}}$        (12)
Stop
```

The network growth analysis algorithm analyzes the user's trust in support of network growth. It has been performed by computing bandwidth utilization support and data rate support generated. Using both values, the method adds the value of TNS. An estimated TNS value has been used to perform intrusion detection.

3.3 QoS Growth Analysis

Analyzing a user's trust toward achieving higher QoS growth is essential in enforcing intrusion detection. The user's trust can be measured by measuring different factors to compute the Trust Quality of service support (TQS). The throughput growth support produced by any user is analyzed to monitor how the user is supportive of improving the QoS performance of the environment. The method computes the TQS (Trusted quality of service Support) according to the number of access, average data transmitted, success ratio, number of failures, malicious access, etc. The method reads the access trace, computes the number of negative access notices and the total number of entries the user makes, and calculates the average data transmitted to compute the TQS value to support intrusion detection.

The above QoS growth analysis measures the user's trust in supporting the QoS performance of the user. It has been calculated according to various factors of QoS, and based on that, the value of TQS is measured to support intrusion detection.

Algorithm:

```
Given: Service Traces ST, User ID Uid, Service Taxonomy STax, Service Id Sid.
Obtain: TQS.
Start
        Read ST, Uid, Sid.
        Find user traces $UTs = \bigcup_{i=1}^{size(ST)} ST\,(i)\,.User == Uid$
        Compute Total Access Count $\mathrm{Tac} = \sum_{\mathrm{i=1}}^{\mathrm{Size(UTs)}} \mathrm{UTs}\ (\mathrm{i})\ .ServiceId == \mathrm{Sid}$    (13)
        Compute average data transmitted $\mathrm{AdT} = \frac{\sum_{\mathrm{i=1}}^{\mathrm{Size(Uts)}} \mathrm{Uts}\ (\mathrm{i})\ .datalength}{\mathrm{Size(UTS)}}$    (14)
        Compute Success Rate SR.
        $\mathrm{SR} = \sum_{\mathrm{i=1}}^{\mathrm{size(UTS)}} \mathrm{UTS}\ (\mathrm{i})\ .State == Success\ \&\&\ \mathrm{UTS}\ (\mathrm{i})\ .Service == \mathrm{Sid}$    (15)
        Compute No of failures Nf.
        $\mathrm{Nf} = \sum_{\mathrm{i=1}}^{\mathrm{size(UTS)}} \mathrm{UTS}\ (\mathrm{i})\ .State == Failed\ \&\&\ \mathrm{UTS}\ (\mathrm{i})\ .Service == \mathrm{Sid}$    (16)
        Compute no of malicious access Nmac.
        $\mathrm{Nmac} = \sum_{\mathrm{i=1}}^{\mathrm{size(UTS)}} \mathrm{UTS}\ (\mathrm{i})\ .State == Malicious\ \&\&\ \mathrm{UTS}\ (\mathrm{i})\ .Service == \mathrm{Sid}$    (17)
        Compute $\mathrm{TQS} = \frac{\mathrm{SR}}{\mathrm{Tac}} \times \frac{\mathrm{Nmac}}{\mathrm{Tac}} \times \frac{\mathrm{Nf}}{\mathrm{Tac}}$    (18)
Stop
```

3.4 Intrusion Detection

The MFTEM model performs intrusion detection according to further analysis. The user's trust is analyzed to support service growth, network growth, and QoS growth. The method uses the access traces of various services performed by the user. Accordingly, the process receives the user request and analyzes Service Growth, Network Growth, and QoS growth. Using the result of further growth analysis, the method computes the value of TUS (Trust User Score). Based on the value of TUS, the method performs intrusion detection. The value of threshold (Th) is measured according to the frequency of malicious access identified and the maximum TUS score identified on the malicious access. At each interval, the value of Th is adjusted to restrict malicious access.

Algorithm:

```
Given: Service Traces STs, Service Request SR, Service Taxonomy ST
Obtain: Boolean
Start
                Read STs, ST, and SR.
                Service Requested Sreq = Request $\in SR$
                TSS = Perform Service Growth Analysis (STS, Sreq, Uid)
                TNS = Perform Network Growth Analysis (STS, Sreq, Uid)
```

(Continued)

Algorithm: Continued

```
                TQS = Perform QoS growth analysis (STS, Uid, Sid)
                Compute $TUS = \frac{TNS}{TSS} \times TQS$          (19)
                If TUS>Th then
                        Return true
                Else
                        Return false.
                End
Stop
```

The proposed MFTEM intrusion detection model analyzes the user's trust in service growth, network growth, and QoS growth. Using the outcome of the analysis, the method computes the value of the Trusted User Score to classify the user request to perform intrusion detection.

4 Results and Discussion

The Multi-fractal trust evaluation model (MFTEM) intrusion detection is analyzed for its performance detecting intrusion attacks in a cloud environment. The analysis and detection of intrusion attacks are performed with the support of service traces maintained by the model belonging to different services. With the traces maintained, the performance of the proposed model has been evaluated under various parameters. The details of the case study used for performance evaluation are displayed in Table 1. The methods are measured for their performance in the below metrics and compared with the rest of the approaches.

Table 1: Evaluation details

Parameter	Value
Tool used	Python
Data set	Amazon
Platform	Microsoft azure
Total records	1 million
No of services	100
No of users	5000

The performance evaluation is conducted based on the E-commerce data set maintained by Amazon data set. According to the services provided by the Amazon environment, the services are monitored, and the system maintains traces. Using the trace maintained, the method analyzes the performance of the proposed system. The Amazon environment maintains the traces of various cloud services accessed in a data set. There are several services to carry out different E-commerce activities, and such access traces are maintained in the data set to support intrusion detection.

Intrusion Detection Accuracy

The methods are measured for their accuracy in detecting intrusion attacks according to the number of traces available. In each case, the performance is measured and compared with others.

$$\text{Intrusion Detection Accuracy} = \frac{TP + TN}{Total\,Attacks} \times 100 \tag{20}$$

The value of intrusion detection accuracy generated by various approaches in the presence of various records is measured. The results are compared with others in Table 2, and the MFTEM model introduces higher intrusion detection accuracy.

Table 2: Analysis of intrusion detection accuracy

Intrusion detection accuracy %			
No of records	3 Lakhs	5 Lakhs	10 Lakhs
VMShield	73	77	82
Deep-IFS	79	85	87
PSBSA	87	93	98
MFTEM	89	95	99

The value of intrusion detection accuracy produced by various approaches is measured and compared in Fig. 2. The proposed MFTEM model has made higher accuracy compared to other methods.

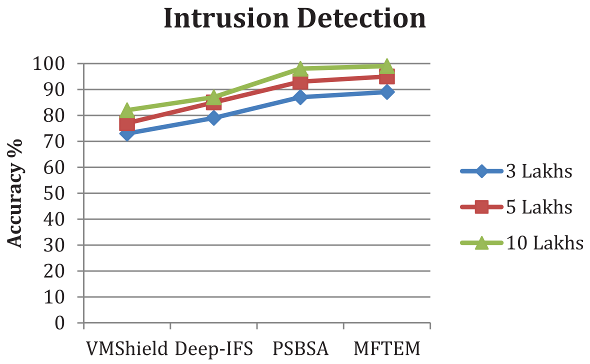

Figure 2: Analysis of intrusion detection accuracy

False Ratio in Intrusion Detection

The ratio of false classification introduced by different approaches is measured according to several True Negative (TN) and False Positive (FP) classifications made by the algorithm. It has been calculated as follows.

$$\text{FRID} = \frac{FP + TN}{Total\ Attacks} \times 100 \tag{21}$$

The false ratio in intrusion detection is measured for different methods and presented in Table 3. The proposed MFTEM model produces less false ratio compared to other models.

Table 3: Analysis of false ratio in intrusion detection

False ratio in intrusion detection %			
No of records	3 Lakhs	5 Lakhs	10 Lakhs
VMShield	27	23	18
Deep-IFS	21	15	13
PSBSA	13	7	2
MFTEM	11	5	1

The value of false classification introduced by the various models is presented in Fig. 3. The proposed model introduces fewer false ratios than others.

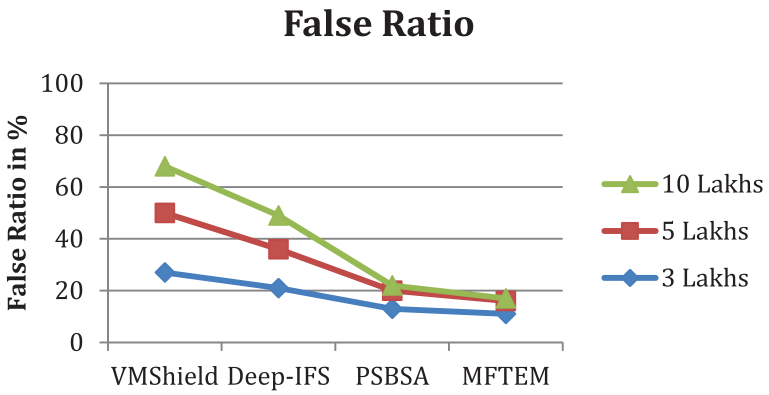

Figure 3: Analysis of false ratio in intrusion detection

Time Complexity

The value of time complexity introduced by different approaches is measured and presented in Table 4.

Table 4: Analysis of time complexity in intrusion detection

Time complexity in intrusion detection in seconds			
No of records	3 Lakhs	5 Lakhs	10 Lakhs
VMShield	28	43	78
Deep-IFS	25	35	67
PSBSA	17	27	42
MFTEM	15	24	36

The time complexity in detecting the intrusion attack is measured for various approaches and presented in Table 4. The proposed MFTEM model introduces less time complexity than others.

The value of time complexity produced by different approaches is measured and presented in Fig. 4. The proposed model makes less time complexity compared to others.

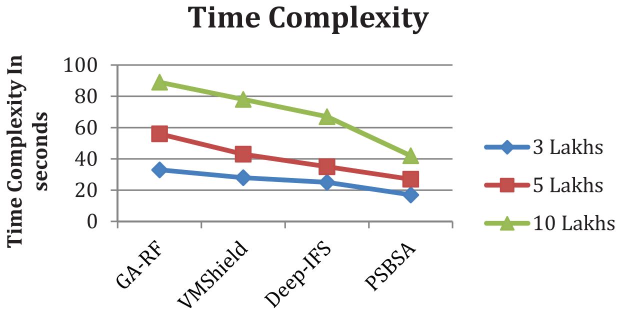

Figure 4: Analysis of time complexity in intrusion detection

5 Conclusion

This article presented a novel multi-fractal trust evaluation model (MFTEM) for intrusion detection. The model works according to the traces maintained by Amazon, and intrusion detection is performed based on the service traces. The model reads the service traces and analyzes service growth, network growth, and service growth quality to measure users' support. The method computes the trust user score according to the trust support values obtained from various growth factors. The method analyzes the service, network, and QoS growth to measure the support factors. According to the value of TUS measured, the process performs intrusion detection. The proposed MFTEM model improves the performance in intrusion detection by up to 99%. Further, the work can be extended by incorporating time-variant measures toward detecting intrusion attacks.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] O. Alkadi, N. Moustafa, B. Turnbull and K. K. R. Choo, "A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9463–9472, 2021. https://doi.org/10.1109/JIOT.2020.2996590

[2] M. Nadeem, A. Arshad, S. Riaz, S. S. Band and A. Mosavi, "Intercept the cloud network from brute force and DDoS attacks via intrusion detection and prevention system," *IEEE Access*, vol. 9, pp. 152300–152309, 2021. https://doi.org/10.1109/ACCESS.2021.3126535

[3] L. Vu, Q. U. Nguyen, D. N. Nguyen, D. T. Hoang and E. Dutkiewicz, "Deep generative learning models for cloud intrusion detection systems," *IEEE Transactions on Cybernetics*, vol. 53, no. 1, pp. 565–577, 2023. https://doi.org/10.1109/TCYB.2022.3163811

[4] A. Mourad, H. Tout, O. A. Wahab, H. Otrok and T. Dbouk, "Ad Hoc vehicular fog enabling cooperative low-latency intrusion detection," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 829–843, 2021. https://doi.org/10.1109/JIOT.2020.3008488

[5] J. Wang, S. Hao, Y. Li, Z. Hong, F. Yan *et al.*, "TVIDS: Trusted virtual IDS with SGX," *China Communications*, vol. 16, no. 10, pp. 133–150, 2019. https://doi.org/10.23919/JCC.2019.10.009

[6] Y. Wu, L. Nie, S. Wang, Z. Ning and S. Li, "Intelligent intrusion detection for internet of things security: A deep convolutional generative adversarial network-enabled approach," *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 3094–3106, 2023. https://doi.org/10.1109/JIOT.2021.3112159

[7] P. Singh, A. Kaur, G. S. Aujla, R. S. Batth and S. Kanhere, "DaaS: Dew computing as a service for intelligent intrusion detection in edge-of-things ecosystem," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12569–12577, 2021. https://doi.org/10.1109/JIOT.2020.3029248

[8] A. Fatani, M. AbdElaziz, A. Dahou, M. A. A. Al-Qaness and S. Lu, "IoT intrusion detection system using deep learning and enhanced transient search optimization," *IEEE Access*, vol. 9, pp. 123448–123464, 2021. https://doi.org/10.1109/ACCESS.2021.3109081

[9] H. Yin, M. Xue, Y. Xiao, K. Xia and G. Yu, "Intrusion detection classification model on an improved k-dependence bayesian network," *IEEE Access*, vol. 7, pp. 157555–157563, 2019. https://doi.org/10.1109/ACCESS.2019.2949890

[10] P. Mishra, P. Aggarwal, A. Vidyarthi, P. Singh, B. khan *et al.,* "VMShield: Memory introspection-based malware detection to secure cloud-based services against stealthy attacks," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 10, pp. 6754–6764, 2021. https://doi.org/10.1109/TII.2020.3048791

[11] M. A. Basset, V. Chang, H. Hawash, R. K. Chakrabortty and M. Ryan, "Deep-IFS: Intrusion detection approach for industrial internet of things traffic in fog environment," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7704–7715, 2021. https://doi.org/10.1109/TII.2020.3025755

[12] A. Bhardwaj, V. Mangat and R. Vig, "Hyperbandtuned deep neural network with well posed stacked sparse autoencoder for detection of DDoS attacks in cloud," *IEEE Access*, vol. 8, pp. 181916–181929, 2020. https://doi.org/10.1109/ACCESS.2020.3028690

[13] A. Yahyaoui, T. Abdellatif, S. Yangui and R. Attia, "READ-IoT: Reliable event and anomaly detection framework for the internet of things," *IEEE Access*, vol. 9, pp. 24168–24186, 2021. https://doi.org/10.1109/ACCESS.2021.3056149

[14] S. Kautish, A. Reyana and A. Vidyarthi, "SDMTA: Attack detection and mitigation mechanism for ddos vulnerabilities in hybrid cloud environment," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 6455–6463, 2022. https://doi.org/10.1109/TII.2022.3146290

[15] P. Singh, G. S. Gaba, A. Kaur, M. Hedabou and A. Gurtov, "Dew-cloud-based hierarchical federated learning for intrusion detection in IoMT," *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 2, pp. 722–731, 2022. https://doi.org/10.1109/JBHI.2022.3186250

[16] M. Yassin, H. O. Slimane, C. Talhi and H. Boucheneb, "Multi-tenant intrusion detection framework as a service for SaaS," *IEEE Transactions on Services Computing*, vol. 15, no. 5, pp. 2925–2938, 2022. https://doi.org/10.1109/TSC.2021.3077852

[17] W. Wang, X. Du and N. Wang, "Building a cloud IDS using an efficient feature selection method and SVM," *IEEE Access*, vol. 7, pp. 1345–1354, 2019. https://doi.org/10.1109/ACCESS.2018.2883142

[18] D. Vaz, D. Matos, M. L. Pardal and M. Correia, "MIRES: Intrusion recovery for applications based on backend-as-a-service," *IEEE Transactions on Cloud Computing*, pp. 1, 2022. https://doi.org/10.1109/TCC.2022.3178982

[19] A. A. Alhabshy, B. I. Hameed and K. A. Eldahshan, "An amelioratedmultiattack network anomaly detection in distributed big data system-based enhanced stacking multiple binary classifiers," *IEEE Access*, vol. 10, pp. 52724–52743, 2022. https://doi.org/10.1109/ACCESS.2022.3174482

[20] Z. Chkirbene, A. Erbad, R. Hamila, A. Gouissem, A. Mohamed *et al.,* "Machine learning based cloud computing anomalies detection," *IEEE Network*, vol. 34, no. 6, pp. 178–183, 2020. https://doi.org/10.1109/MNET.011.2000097

[21] S. Nandy, M. Adhikari, M. A. Khan, V. G. Menon and S. Verma, "An intrusion detection mechanism for secured IoMT framework based on swarm-neural network," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 5, pp. 1969–1976, 2022. https://doi.org/10.1109/JBHI.2021.3101686

[22] H. Ma, Y. Xie, S. Tang, J. Hu and X. Liu, "Threat-event detection for distributed networks based on spatiotemporal markov random field," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 3, pp. 1735–1752, 2022. https://doi.org/10.1109/TDSC.2020.3036664

[23] A. Telikani, J. Shen, J. Yangand and P. Wang, "Industrial IoT intrusion detection via evolutionary cost-sensitive learning and fog computing," *IEEE Internet of Things Journal*, vol. 9, no. 22, pp. 23260–23271, 2022. https://doi.org/10.1109/JIOT.2022.3188224

[24] A. Chougule, V. Kohli, V. Chamola and F. R. Yu, "Multibranchreconstruction error (MbRE) intrusion detection architecture for intelligent edge-based policing in vehicular Ad-Hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2022. https://doi.org/10.1109/TITS.2022.3201548

[25] K. Ramkumar, N. Ananthi, D. R. Denslin Brabin, Puneet Goswami, M. Baskar *et al.,* "Efficient routing mechanism for neighbour selection using fuzzy logic in wireless sensor network," *Computers & Electrical Engineering*, vol. 94, no. 3, pp. 107365, 2021. https://doi.org/10.1016/j.compeleceng.2021.107365

[26] K. G. Maheswari, C. Siva and G. Nalinipriya, "An optimal cluster based intrusion detection system for defence against attack in web and cloud computing environments," *Wireless Personal Communications*, vol. 128, no. 3, pp. 2011–2037, 2022.

[27] G. Nalinipriya, K. G. Maheswari, Balamurugan Balusamy, K. Kotteswari and Arun Kumar Sangaiah, "Availability modeling for multi-tier cloud environment," *IntellIgent Automation & Soft ComputIng*, vol. 23, pp. 485–492, 2017.

[28] G. John Samuel Babu and M. Baskar, "Trs scheduling for improved qos performance in cloud system," *Computers, Materials & Continua*, vol. 75, no. 1, pp. 1547–1559, 2023.

[29] K. Bala, A. Chandra Sekar, M. Baskar and J. Paramesh, "An efficient multi level intrusion detection system for mobile ad-hoc network using clustering technique," *International Journal of Engineering and Advanced Technology*, vol. 8, no. 6, pp. 1977–1985, 2019.

[30] S. Soundararajan, R. Prabha, M. Baskar and T. J. Nagalakshmi, "Region centric GL feature approximation based secure routing for improved qos in MANET," *Intelligent Automation & Soft Computing*, vol. 36, no. 1, pp. 267–280, 2023.