*International Journal of Basic and Applied Sciences*, 14 (SI-1) (2025) 585-593

International Journal of Basic and Applied Sciences

*Website: www.sciencepubco.com/index.php/IJBAS*
*https://doi.org/10.14419/0g6p7166*
***Research paper***

# Enhancing Cloud Service Performance by Mitigating DDoS Attacks with Multilevel Time-Oriented Analysis

**Dhiyanesh B. [1] *, Dayana R. [1], Saranya N. [2], Daniel Madan Raja S. [3], Jayashree V. [4]**

*[1] Assistant professor/CSE, SRM Institute of Science and Technology, Vadapalani campus, Chennai, Tamil Nadu, India*
*[2] Assistant Professor/AIDS, Karpagam College of Engineering, Coimbatore, Tamil Nadu, India*
*[3] Associate Professor, CSE, Karunya Institute of Technology and Sciences, Coimbatore, Tamil Nadu, India*
*[4] Assistant Professor/IT, K. Ramakrishnan College of Engineering, Trichy, Tamil Nadu, India*
**Corresponding author E-mail: dhiyanu87@gmail.com*

*Received: July 15, 2025, Accepted: July 24, 2025, Published: November 1, 2025*

**Abstract**

In many organizations, cloud computing is becoming increasingly common. Distributed Denial of Service (DDoS) attacks are a concern that can be mitigated in several ways, but they have a noticeable impact on service performance. Most methods perform trust verification only at the initial stage, but malicious activities often occur at the middle or end stage. It highly affects the service performance of the user, and it should be monitored and stopped initially. A Multilevel Time Oriented Analysis (MLTOA) is performed to identify such malicious requests. The high-threat service is determined by tracking the service history in different states. Using the MLTOA scheme, the service history can be grouped into other states, and the user service break that occurs in any state can be identified. Then the method backtracks two states: one is the initial state of the request, and the second is how the malicious user starts the service request. Thus, detecting DDoS attacks can be performed efficiently by monitoring such states and deciding whether they are malicious or genuine. Therefore, the method does not allow the user to access the service. DDoS detection accuracy increases dramatically with the proposed method, reduced time complexity, and improved service utilization.

***Keywords***: *Multilevel Time-Oriented Analysis; Service Performance; Trust Verification; Service History Monitoring; DDoS Detection; Malicious Requests.*

## 1. Introduction

Organizations are turning to cloud computing, an emerging technology, to avoid the enormous costs associated with resource management and infrastructure development. The cloud environment offers a gateway for the organization to provide services through the cloud, which is cost-effective. A service provider offers services on behalf of the organizations to execute their jobs, and the cloud user can access the service by registration. Certain restrictions are maintained based on identity management and authentication mechanisms that are complicated and produce more network overhead for accessing cloud users.

The Third-Party Auditor (TPA) has been used to illustrate such methods; its purpose is to preserve the user identity. The supplier of services will communicate with the TPA to verify the identity when a request is received. The presentation of the TPA has introduced a set of problems with the security of the cloud, and exchanging user information with a third party will not be accepted in many situations. So, detecting unauthorized access has adapted to the service provider's latest infrastructure designs. A generic implementation of Identity Access and Management will lead to increased time complexity, resulting in additional network overhead.

The Third-Party Auditor (TPA) is depicted as a mechanism primarily responsible for preserving the individual's identity. The service provider will communicate with the TPA to verify the identity when it receives a request. The presence of TPA has introduced a set of problems in the security of the cloud, and exchanging user information with a third party will not be acceptable in many situations. Therefore, detecting such unauthorized access has been adapted with the service provider in the latest infrastructure designs.

The cloud user, upon registration, could access the services provided by the CSP, and there is no limit or boundary for them. Sometimes, some malicious users generate an enormous number of requests to the service provider to stop or degrade the service. DDoS attacks will identify and eliminate the problem of service unavailability for the users in the cloud environment.

Several methods are available for identity access management, but users can still engage in some malicious activities. The loosely coupled nature is more prone to various threats and high exposure to the attacks of DDoS. The DDoS attack is one that a different user produces with the focus on degrading the service quality of the environment. Any cloud service has a particular capability in accepting the user request. For example, if the service can handle 100 requests, then there will be some malicious users who produce 10000 requests at a time. It degraded the service, and the genuine user does not get access to the service. It must be identified and removed to improve the cloud performance.

The services available in any SOA are at different levels, with the cloud SaaS always on top. Whatever service they want to access must be approached through the level. The software services have specific protocols for accessing them. For example, the user identity verification service has the authentication mechanism: "Username: Password," which must be submitted to the service. If the service is not accessed in the correct format, then it will be identified as malicious service access. Distributed Denial of Service is also recognized in the above mechanism, which produces more requests. Instead of making incorrect data, the malicious user can also have a large amount of data. For example, for a service that requires data within the size of 20 KB, the malicious user can submit 10 MB of data. They degraded the service throughput.

The DDoS attacks can be performed in different ways, producing a higher amount of data and many requests. The malicious user will spoil the entire service throughput and reduce the performance. In another way, the malicious user can follow the service to the final stage by having correct data, but at the final stage, they can break the service sequence and quit. So, a different strategic approach is required to handle the concern of DDoS with the SaaS architecture.

## 2. Methods Explored

The MLTOA networks the traffic according to the different time windows. The method considers the packet features with various focuses according to the time factor. Such techniques are discussed in this section.

In this dissertation, Yang Xiang et al. [1] inventively propose the use of two new information metrics to differentiate low-rate DDoS attack activity from ordinary traffic. The results of the experiment show that the suggested information metrics can detect low-rate DDoS assaults with a significantly reduced false positive rate. In addition, the proposed IP traceback algorithm can identify attacks from a local area network (LAN) and ignore attack traffic.

Kandula et al. [2] demonstrate the idea and implementation of Kill-Bots, a kernel plugin for DDoS defense against flash crowd-like DDoS attacks on Web servers.

Compared to Xiang et al. [1], which is centered on low-rate DDoS attacks, which are limited in terms of scale, the MLTOA methodology proposes multilevel state analysis to detect both high-rate and low-rate attacks with dynamic cloud workloads. Equally, Kandula et al. [2] examine how Kill-Bots can be used to address DDoS attacks; however, their construction is also limited in scalability because it depends on fixed, static state checks. In its turn, MLTOA can easily change as attack patterns evolve because of its multilevel approach, which improves the detection in different cloud settings. In contrast to other systems that employ graphical checks for authentication, Kill-Bots offers this service. Kill-Bots initially uses an intermediate stage to identify the IP addresses that consistently fail the test, ignore them, and flood the server with requests. Because these devices aim to overload the server, they are referred to as bots. When a computer is identified by Kill-Bots, it blocks its requests, turns off the graphical checks, and allows access to authorized people who can't or won't pass the tests. Then, Kill-Bots sends a test and assesses the client's answer without allowing unauthorized clients to access sockets, TCBs, or worker processes. Consequently, it protects the authentication process against DDoS attacks. Thirdly, KillBots incorporates authentication and admission control systems. Efficient operation is achieved, irrespective of the cause of the server overload, a DDoS attack, or an actual Flash Crowd.

B.Dhiyanesh& S.Sakthivel [3] in this research, a unique F2C-(Flow-Frequency-Completeness) algorithm is discussed to enhance performance in mitigating DDoS attacks. The process generates traces about the service access and keeps track of how various users access the service. The service frequency is then calculated using the access trace, and the quantity of payload given for each service access is noted. Similarly, the number of services accessed from the service access trace results in an identified success flag to compute the completeness measure. The approach calculates the user's trustworthiness to reduce the harmful service requests using all these safeguards. The method effectively reduced the effect of DDoS and increased performance by up to 99.8%.

Q.He [4] This study suggests Chord4S, a decentralized approach to peer-to-peer service discovery, to address those issues. Greater data availability and efficient searches with little overhead are made possible by Chord4S, according to the experimental study. Decentralized service distribution and discovery are made possible by Chord4S, which uses the widespread Chord's data distribution and lookup features. Data accessibility is further increased by the dissemination of published descriptions of functionally equivalent services to different successor nodes organized into virtual segments in the Chord4S circle. With a wildcard, Chord4S also allows service discovery. The Chord routing protocol has also been expanded to facilitate the quick finding of several services with a single query. This makes it possible for service users and several potential service providers to negotiate Service Level Agreements (SLAs) later.

X.Wang et al. [5], to conduct a more thorough and reliable reputation evaluation, this study introduced a broad trust model termed RLM. The author specifically developed a complete reputation evaluation approach based on two characteristics: reputation value and reputation prediction variance. A quality indicator of the reputation value derived from a sum of feedback is the reputation prediction variance. For feedback aggregation, the author proposes the novel Kalman aggregation method, which can naturally promote trustworthy trust evaluation. The resistance of the RLM design to the two primary types of feedback attacks, adulating and defaming attacks, was examined theoretically. When it comes to accuracy and resilience against attacks, the RLM model performs better than the popular summation-based trust models. It can faithfully depict the development of a reputation. Specifically, under the risk of collusive harmful feedback, RLM provides enhanced resilience for reputation prediction and a decreased false positive rate for malicious feedback identification.

B.Dhiyanesh & S.Sakthivel [6], the UBP-Trust model, which is discussed in this paper, keeps track of users' usage patterns in various cloud settings. Based on the monitoring results, a user behavior pattern is created, which includes how many times it has been accessed, how many times it has been successfully accessed, how much data was sent, how many false invocations were made, and any protocol variances. Considering all of these aspects, the technique creates a behavioral pattern and determines the user trust weight for every user under observation. The suggested method yields accurate DDOS detection findings with less time complexity and false classification rates.

AlZain et al [7] evaluate the state-of-the-art research on possible solutions and single and multi-cloud security. The research community has shown more interest in single clouds than in multi-cloud providers when it comes to security maintenance. Because they can lower security threats that impact cloud computing users, multi-clouds are the focus of this initiative, which attempts to encourage their adoption.

Xiaohuiet. Al [8] proposes a confidentiality defense device for service-oriented identity verification in this examination.

B.Dhiyanesh et al. [9] This study surveys the potential security benefits of using several clouds simultaneously. Several novel designs are presented and examined in relation to their privacy and security features.

J.Bohli et al. [10] performed a broad survey on different methods at the service and architecture levels. Furthermore, the cloud paradigm introduces new security approaches, techniques, architectures, and security concerns. This paper surveys how multiple distinct clouds can be used simultaneously to improve security.

Pitropakis Nikolaos. [11] This study attempts to detect co-residency and network stressing attacks by applying the Smith-Waterman genetic algorithm to KVM-based cloud environments. Testing the suggested strategy in a test-bed environment confirmed its efficacy.
Sakthivel.S & Dhiyanesh. B [12] highlighted problems with data integrity related to cloud storage. Although storing and managing files on the cloud can reduce expenses, issues can develop if the data becomes corrupted. It will take a while for users to become aware of this problem unless they try to retrieve them. Somesh et al. [13] use fuzzy set theory to maintain cloud security. The method generates a trust chain in which the user authentication is verified.
Yamato, Yoji. [14] propose a two-tier abstraction, in which software is grouped into software and functional groups, and test cases are selected based on their contents. Using Jenkins, they confirmed the feasibility of the proposed method on OpenStack. Test efforts were evaluated, including creating, extracting, and executing test cases.
S.Sakthivel& B.Dhiyanesh [15] explained their paradigm, presenting several fresh difficulties for access control and data protection. Almost all services encrypt and store user data in a way that is concerned with security vulnerabilities. Once users outsource complicated tasks to a cloud server, the information shared in those servers is not inside the data owners' domain.
A. Sahi et al. [16] suggest detection and prevention, such as SVM, KNN, and NB, in the cloud environment. Likewise, A. Aljuhani et al. [17] introduced Machine Learning techniques for combating DDoS attacks in modern networking circumstances.
A. Bhardwaj et al. [18] have presented a DNN algorithm with an Auto-Encoder (AE) to identify the DDoS attack. AE and DNN techniques reduce the low false rate and time complexity. However, these techniques give the wrong result for attack detection.
S. Kautish et al. [19], the SDMTA approach is used to detect DDoS. Nonetheless, due to high traffic, legitimate users cannot use these services, resulting in financial loss.
M. Nadeem et al. [20] explained brute force and DDoS attack detection systems and prevention. O. A. Wahab et al. [21] study focused on Virtual Machine (VM) based DDoS attack detection using a Two-Fold Solution. However, the suggested solutions do not meet the requirement challenges.
In N. Agrawal et al. [22], the authors seek to identify cloud-based DDoS attack vulnerabilities. Because low-volume DDoS attacks are stealthy and generate little bandwidth, they are challenging to detect.
D. Erhan et al. [23] designed a Matching Pursuit Algorithm (MPA) to detect the DDoS attack. However, resource-intensive DDoS attacks are still challenging problems. A. Alsirhani et al. [24] performed a Fuzzy Logic (FL) based DDoS attack detection. However, the suggested solution is challenging to detect DDoS attacks.

## 3. MLTOA Approach

The problem of DDoS has been handled in different methods, but the impact of DDoS attacks is highly noticeable on the service performance. The MLTOA approach is discussed in this paper to solve the issue. The users in trust can access the cloud services, and it follows to access multiple services in the chain. Most methods perform trust verification only at the initial stage, but the malicious ending occurs at the middle or end state. It highly affects the service performance and should be stopped initially. To identify such malicious requests, an MLTOA is performed. They monitor the service history in a different state and identify the service with a higher threat.
The KDDcup99 dataset has become a popular one in DDoS attack detection studies and serves to test mitigation measures. Although it is more ancient, it has been relevant because it has offered a broad range of attack types, including DDoS. Future studies, however, consider examining more recent data sets, such as CICIDS2017, that present more recent attack patterns and network traffic simulations that are applicable to current-day cloud infrastructure. KDDcup99 was chosen in this research because it has a long track record of use in academic DDoS detection, and the results of this work are comparable to other previous studies.

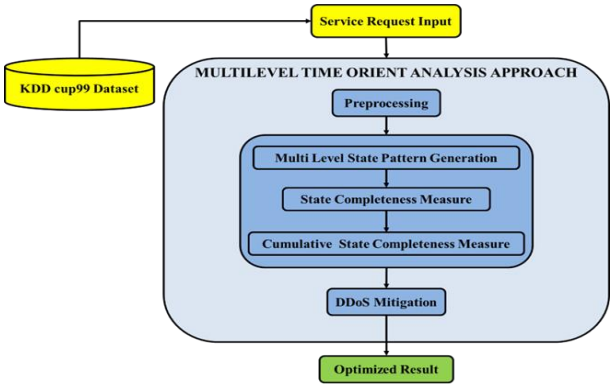


**Fig. 1:** Proposed System Framework Components.

Fig. 1 defines a proposed system framework component. First, we collect the KDDcup99 dataset from the online Kaggle repository. Then, preprocess the dataset to reduce null values and fill missing values from the collected dataset. Then the method backtracks the initial state of the request and how the malicious user starts the service request. It increases the performance of cloud services and increases the throughput ratio. By monitoring such states, the detection of DDoS attacks will perform efficiently.

### 3.1. Preprocessing

The cloud trace contains information about the user access and the state of the user service access, the cloud trace and identifies the presence of all information. Incomplete traces are removed from the log. The preprocessed log is used to generate a state pattern.

Algorithm:

Input: CT
Output: PL.
Start
Go through CT.

```
        For each log L
                Identify the presence of all features.
                If CT(l) ∈ ∀Features(CT) then
                        Leave the trace.
                Else
                        Remove the log.
                End
        End
Stop
```

From the gathered KDDcup-99 dataset, the previously mentioned methodology stages produce an effective preprocessed dataset. The first step is to analyze each log to identify the presence of all attributes.

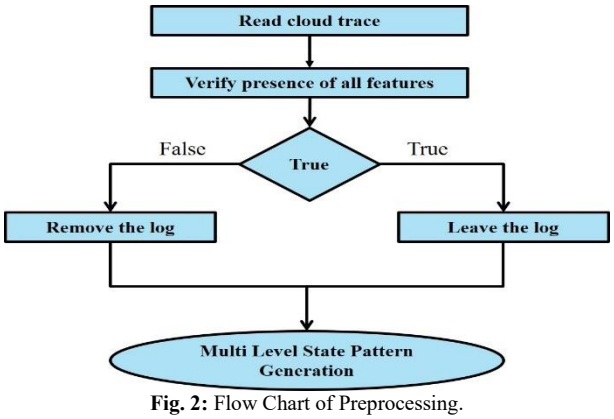


**Fig. 2:** Flow Chart of Preprocessing.

The preprocessing flow chart and list of steps are displayed in Figure 2. The preprocessing algorithm reads the cloud traces and determines the collection of all distinct features found in various trails. Further, each trace has been verified for the existence of each of the features. Verify all features in the cloud log, and if any of the logs do not contain all the features, they will be removed from the log. The logs or traces identified with incomplete features or missing values are eliminated from the traces. Such noise-removed logs perform pattern generation for distributed denial of service attack detection.

### 3.2. Multi-level state pattern generation

As part of this process, the method identifies the states through which the service request has passed. First, the list of states followed by the service request is determined. Then the technique generates the multilevel pattern using the states, and according to the value of the state, the method generates the multilevel state pattern. Generated patterns are added to the pattern set, which will be used to perform Distributed Denial of Service attack detection.

Algorithm:

Input: Preprocessed Log PL
Output: Multi-Level State Pattern Set MLSPS.
Start
Read Preprocessed Log PL.
Identify the list of levels and states.
$Ls = \sum Levels \in PL$
Split the log into different time windows.
For each time window Ti
Identify a list of traces.
$Tl = \int_{i=1}^{size(Tw)} \sum Pl(i).Time == Ti$
For each log l
Identify a list of states $ss = \sum states \in tl$
For each state, identify the status.
Generate pattern $Pi=\{\sum states, Status\}$
End
End
Stop.

The above algorithm steps give a Multilevel State Pattern Set (Mlsps) from the preprocessed dataset. First, read the logs, then identify the list of levels and states to split the log into different time windows. Identify the list of traces and states, create a pattern Pi, and obtain the pattern set Mlsps.

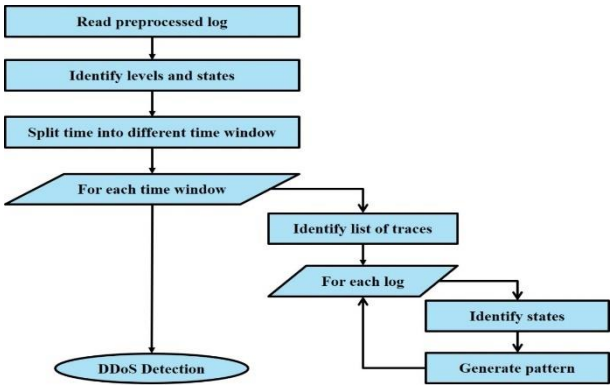


**Fig. 3:** Flow Chart of Multilevel State Pattern Generation.

Fig. 3 shows the flow chart of Multilevel State Pattern Generation used to perform DDoS detection. The above algorithm identifies the states and status of service at each level to produce the pattern. The service being accessed by the user could have been left in any state to generate a Distributed Denial of Service attack. Therefore, the state can be identified by generating the pattern of access, how the malicious user performs the attack, and in what state they leave the service at the intermediate. To achieve the method reads the logs, and from each log, the method finds the set of states of the service access. According to the states identified, the process generates a pattern of service access. This has been generated for different time stamp identification patterns to support Distributed Denial of Service attack detection.

### 3.3. DDoS detection

As part of this process, the method identifies the list of patterns broken at an intermediate state with a failure mode. Then, for each state, the method computes the state completeness measure. The method selects the state with a higher threat based on this measure. The same measure is used to identify the state that has a higher threat. Then the methodology measures the cumulative state, and if the state completeness measure is less than any specific threshold, the method identifies the user as malicious.

Algorithm:

Input: Cloud Trace CT
Output: Boolean
Start
PL = Preprocess (CT).
Perform multilevel pattern set generation (PL).
Split the pattern set into different time windows.
For each time window
For each state
Compute the state completeness measure Scm.
$Scm = \frac{\sum States.Status==Success}{Total\ number\ of\ States} \times size\ of\ pattern\ set$
End
End
Compute the cumulative state completeness measure Cscm.
$Cscm = \sum Scm / size(Tw)$
If Cscm>CTh then
Return true
Else
Deny service.
Choose the state with the lowest completeness measure.
End
Stop.

Fig. 4 illustrates the flow chart for spotting Dos. The DDoS attack detection approach reads the cloud trace and performs preprocessing to eliminate the incomplete records. Further, the method generates the multilevel pattern by splitting the logs into several time windows and access states. Such patterns generated are used to measure the state completeness measure in each time window and at each state. According to the value of the state completeness measure and the threshold, the presence of DDoS has been detected.

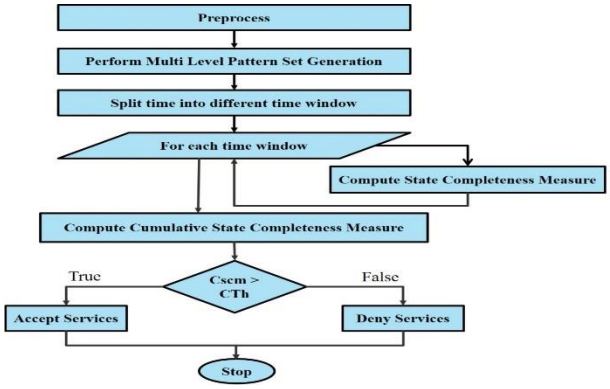


**Fig. 4:** Flow Chart of DDoS Detection.

The 'state completeness measure' (SCM) and 'cumulative state completeness measure' (Cscm) are central to the MLTOA methodology. The state completeness measure quantifies the extent to which a service request follows the expected state transitions. Specifically, Scm is defined as the ratio of completed states to the total states within a time window. Mathematically, it is given by:

$$\text{Scm} = \frac{\text{Number of successful states}}{\text{Total number of states observed in a time window}}$$

The cumulative state completeness measure (Cscm) is the running total of Scm across all time windows, providing a cumulative assessment of service request integrity. It is defined as:

$$\text{Cscm} = \sum_{i=1}^{n} \text{Scm}_i$$

Where I represents each time window in the sequence.

## 4. Result and Discussion

It has been proven to be efficient to implement the Multilevel Time Orient scheme and evaluate its effectiveness. The method has been assessed with different simulation setups and various services and users. All factors have been evaluated efficiently by this method. As a result of the technique, Distributed Denial of Service attacks can now be detected and mitigated more effectively. Table 1 below displays the specifics of the simulation:

**Table 1:** Specifications of the Protocol Simulation

| Parameter | Value |
|---|---|
| Used Instrument | Cloud Sim |
| No of Facilities | 150 |
| No of operators | 300 |
| Window Time | 6 Months |
| Name of the dataset | KDDcup99 |
| No of records | 494021 |
| Tool | Anaconda |
| Language | Python |

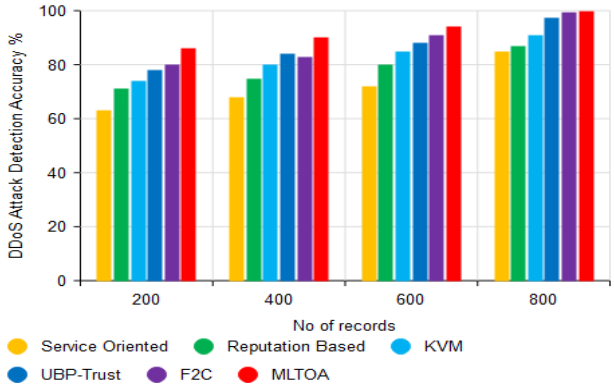


**Fig. 5:** Accuracy of DDoS Detection Comparison.

It is evident from Fig. 5 detection accuracy performance that the proposed strategy beats the other methods in accuracy. The accuracy of DDoS detection has risen to 99.68% with the suggested MLTOA methodology. Also, the performance in DDoS accuracy has dramatically increased in the ratios 14.68, 12.68, 8.68, 2.5, and 0.23 % to service-oriented, reputation-based, F2C, KVM, and UBP Trust methodologies, respectively. The proposed MLTOA approach produces higher performance than F2C through carrying out risk assessment at every stage according to the state pattern.

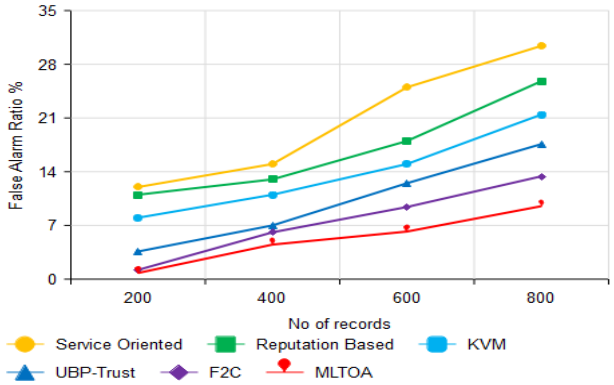


**Fig. 6:** Comparison Graph of MLTOA False Alarm Ratio.

It indicates that in the above graph, the comparison of Fig. 6 of the false alarm ratio achieved by several techniques. The suggested methodology produced fewer false alarms in the MLTOA test compared with other methodologies. The proposed MLTOA algorithm reduces the false alarm ratio being produced up to 0.8 %, and it comparatively reduces the false alarm ratio in the range of 11.2 %, 10.2 %, 7.2 %,

2.8%, and 0.4% less than service-oriented, reputation-based, KVM, UBP Trust, and F2C algorithms. By detecting the threat by classifying the service access state in different levels, the false alarm ratio has been reduced more than the F2C algorithm.

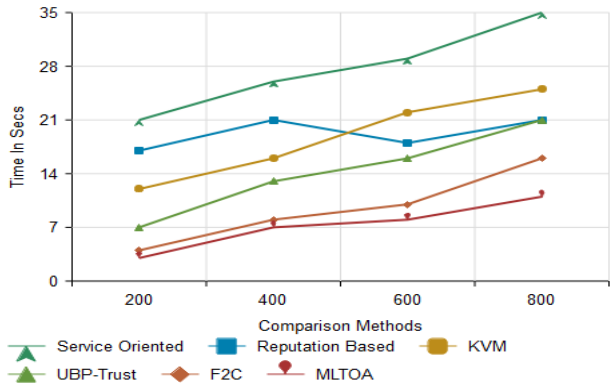


**Fig. 7:** Time Complexity Comparative Graph.

Predictably, through the comparison graph of Fig. 7, the proposed method performs DDoS detection on Time Complexity. The above graph shows that the MLTO Approach has yielded less Time Complexity compared to alternative techniques. Up to three seconds have been significantly cut from the time complexity of DDoS detection by the MLTA, where it is comparatively less than the previous methods in the ratios of 18, 14, 9, 4, and 1 seconds toward service-oriented, reputation-based, F2C, KVM, and UBP Trust methodologies, correspondingly. The time complexity has been decreased by classifying the service access in accordance with multilevel time analysis, which divides the service access into various levels based on success rate. Because there are chances that the service access would be classified as malicious in the initial state, which hugely reduces the time complexity, this produces less time complexity than the F2C approach.

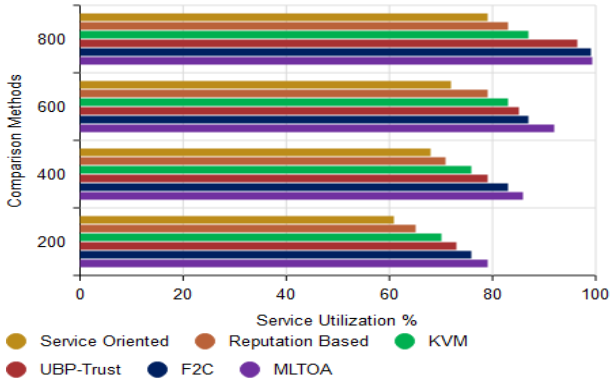


**Fig. 8:** Comparison Graph of Service Utilization.

According to Fig. 8, the proposed MLTOA approach presented higher Service Utilization results than other methods. The proposed method, the MLTOA algorithm, up to 99.4 %, will improve service utilization. The service utilization performance increased by 20.4 %, 16.4 %, 12.4 %, 2.9 %, and 0.4 %towards service-oriented, reputation-based, and KVM, UBP Trust, and F2Calgorithms, respectively. The service utilization improved over the F2C approach because the malicious access or threat was detected early, which supports the increase of genuine service access and improves service utilization.

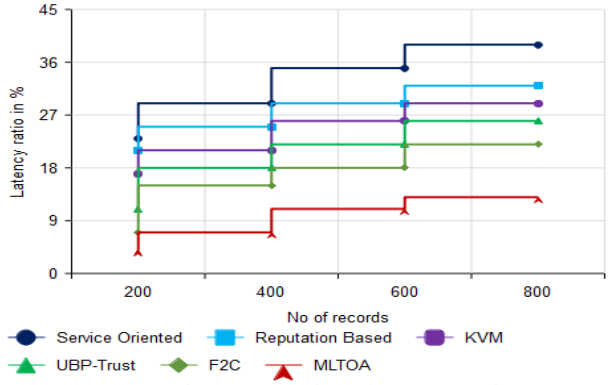


**Fig. 9:** The Latency Ratio Performance Graph.

The results in Fig.9 show similarities in the various approaches used to introduce a latency ratio; the suggested method yields a lower latency ratio than the rest of the approaches.19% less latency than the service-oriented approach and 17% less latency than the reputation-based method have been achieved with the suggested MLTOA, up to 4%. Correspondingly, it is 13 % and 7 % less than the KVM and UBP Trust approaches. The latency ratio of the MLTOA algorithm is 3 % less than the F2C algorithm. By classifying the service access according to the state measures, any service request has been classified with the least time complexity than other F2C approaches.

## 5. Conclusion

This paper introduces the MLTOA technique to solve DDoS attack detection problems in the cloud environment. Initially, we collected the KDDcup99 dataset from the Kaggle repository and then preprocessed it to prepare the dataset from the collected dataset. Then the proposed MLTOA scheme, the service history can be grouped in different states, and the user service break that occurs in any form can be identified. Based on defined values, the process generates the state pattern. Finally, the method computes the State Completeness and Cumulative Measure to perform Distributed Denial of Service attack detection. Thus, detecting DDoS attacks can be performed efficiently by monitoring such conditions and deciding as malicious or genuine. The MLTOA has generated improved results in threat detection with the lowest false ratio and time complexity.

Future Directions: The KDDcup99 dataset has become a popular one in DDoS attack detection studies and serves to test mitigation measures. Although it is more ancient, it has been relevant because it has offered a broad range of attack types, including DDoS. Future studies, however, consider examining more recent data sets such as CICIDS2017 that present more recent attack patterns and network traffic simulations that are applicable to current-day cloud infrastructure. KDDcup99 was chosen in this research because it has a long track record of use in academic DDoS detection, and the results of this work are comparable to other previous studies.

## References


[1] Y. Xiang, K. Li, and W. Zhou, "Low-Rate DDoS Attacks Detection and Traceback by Using New Information Metrics," IEEE Transactions on Information Forensics and Security, vol. 6, no. 2, pp. 426-437, 2011.Article (CrossRef Link) https://doi.org/10.1109/TIFS.2011.2107320.

[2] Srikanth Kandula, Dina Katabi and Matthias Jacob Arthur Berger, "Botz-4-sale: surviving organized DDoS attacks that mimic flash crowds," in Proc. 2nd conference on Symposium on Networked Systems Design and Implementation, pp. 287–300, 2005.Article (CrossRef Link).

[3] B. Dhiyanesh and S. Sakthivel, "F2C: An Novel Distributed Denial of Service Attack Mitigation Model for Saas Cloud Environment," Asian Journal of Research in Social Sciences and Humanities, vol. 6, pp. 192-203, 2016. Article (CrossRef Link) https://doi.org/10.5958/2249-7315.2016.00389.0.

[4] Q. He, J. Yan, Y. Yang, R. Kowalczyk, and H. Jin, "A Decentralized Service Discovery Approach on Peer-to-Peer Networks," IEEE Transactions on Services Computing, vol. 6, no. 1, pp. 64-75, 2013. Article (CrossRef Link) https://doi.org/10.1109/TSC.2011.31.

[5] X. Wang, L. Liu, and J. Su, "RLM: A General Model for Trust Representation and Aggregation," IEEE Transactions on Services Computing, vol. 5, no. 1, pp. 131-143, 2012. Article (CrossRef Link) https://doi.org/10.1109/TSC.2010.56.

[6] Dhiyanesh B and Sakthivel S, "UBP-Trust: User Behavioral Pattern Based Secure Trust Model for Mitigating Denial of Service Attacks in Software as a Service (SaaS) Cloud Environment," Journal of Computational and Theoretical Nanoscience, vol. 13, pp. 7649-7654, 2016. Article (CrossRef Link) https://doi.org/10.1166/jctn.2016.5766.

[7] Mohammed A. AlZain, Eric Pardede, Ben Soh, and James A. Thom, "Cloud Computing Security: From Single to Multi-clouds," in Proc. 45th Hawaii International Conference on System Sciences, pp. 5490-5499, 2012.Article (CrossRef Link) https://doi.org/10.1109/HICSS.2012.153.

[8] Li, Xiaohui, Jingsha He, and Ting Zhang, "A service-oriented identity authentication privacy protection method in cloud computing," International Journal of Grid and Distributed Computing, vol. 6, no. 1, pp. 77-86, 2013.Article (CrossRef Link).

[9] S. Kasthuripriya, B. Dhiyanesh, and S. Sakthivel, "LFTSM-Local Flow Trust Based Service Monitoring Approach for Preventing the Packet During Data Transfer in Cloud," Asian Journal of Information Technology" vol. 15, no. 20, pp. 3927-3931, 2016.Article (CrossRef Link).

[10] J. Bohli, N. Gruschka, M. Jensen, L. L. Iacono, and N. Marnau, "Security and Privacy-Enhancing Multicloud Architectures," IEEE Transactions on Dependable and Secure Computing, vol. 10, no. 4, pp. 212-224, 2013. Article (CrossRef Link) https://doi.org/10.1109/TDSC.2013.6.

[11] Nikolaos Pitropakis, Dimitra Anastasopoulou, Aggelos Pikrakis, and Costas Lambrinoudakis, "If you want to know about a hunter, study his prey: detection of network-based attacks on KVM based cloud environments," Journal of Cloud Computing, vol. 3, no. 20, 2014. Article (CrossRef Link) https://doi.org/10.1186/s13677-014-0020-6.

[12] S. Sakthivel and B. Dhiyanesh, "Secure Data Storage Auditing Service Using Third Party Auditor In Cloud Computing," International Journal of Applied Engineering Research, vol. 10, no. 37, 2015.Article (CrossRef Link).

[13] Somesh Kumar Prajapati, Suvamoy Changder, and Anirban Sarkar, "Trust Management Model for Cloud Computing Environment," ICTACT Journal on Soft Computing, vol. 3, no. 3, pp. 509-513, 2013.Article (CrossRef Link). https://doi.org/10.21917/ijsc.2013.0076.

[14] Yamato Y, "Automatic verification technology of software patches for user virtual environments on IaaS cloud," Journal of Cloud Computing,vol. 4, no. 4, 2015. Article (CrossRef Link) https://doi.org/10.1186/s13677-015-0028-6.

[15] S. Sakthivel and B. Dhiyanesh, "A privacy-preserving storage security for spatial data in dynamics cloud environment," in Proc. Fourth International Conference on Computing, Communications and Networking Technologies, pp. 1-6, 2013.Article (CrossRef Link) https://doi.org/10.1109/ICCCNT.2013.6726759.

[16] Sahi, D. Lai, Y. Li and M. Diykh, "An Efficient DDoS TCP Flood Attack Detection and Prevention System in a Cloud Environment," IEEE Access, vol. 5, pp. 6036-6048, 2017. Article (CrossRef Link) https://doi.org/10.1109/ACCESS.2017.2688460.

[17] Aljuhani, "Machine Learning Approaches for Combating Distributed Denial of Service Attacks in Modern Networking Environments,"IEEE Access, vol. 9, pp. 42236-42264, 2021.Article (CrossRef Link) https://doi.org/10.1109/ACCESS.2021.3062909.

[18] Bhardwaj, V. Mangat, and R. Vig, "Hyperband Tuned Deep Neural Network With Well Posed Stacked Sparse AutoEncoder for Detection of DDoS Attacks in Cloud," IEEE Access, vol. 8, pp. 181916-181929, 2020.Article (CrossRef Link) https://doi.org/10.1109/ACCESS.2020.3028690.

[19] S. Kautish, R. A 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, Sept. 2022.Article (CrossRef Link) https://doi.org/10.1109/TII.2022.3146290.

[20] 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.Article (CrossRef Link) https://doi.org/10.1109/ACCESS.2021.3126535.

[21] O. A. Wahab, J. Bentahar, H. Otrok, and A. Mourad, "Optimal Load Distribution for the Detection of VM-Based DDoS Attacks in the Cloud," IEEE Transactions on Services Computing, vol. 13, no. 1, pp. 114-129, 2020.Article (CrossRef Link) https://doi.org/10.1109/TSC.2017.2694426.

[22] N. Agrawal and S. Tapaswi, "Defense Mechanisms Against DDoS Attacks in a Cloud Computing Environment: State-of-the-Art and Research Challenges," IEEE Communications Surveys & Tutorials, vol. 21, no. 4, pp. 3769-3795.Article (CrossRef Link) https://doi.org/10.1109/COMST.2019.2934468.

[23] D. Erhan and E. Anarim,"Hybrid DDoS Detection Framework Using Matching Pursuit Algorithm,"IEEE Access, vol. 8, pp. 118912-118923, 2020.Article (CrossRef Link) https://doi.org/10.1109/ACCESS.2020.3005781.

[24] Alsirhani, S. Sampalli, and P. Bodorik, "DDoS Detection System: Using a Set of Classification Algorithms Controlled by Fuzzy Logic System in Apache Spark," IEEE Transactions on Network and Service Management, vol. 16, no. 3, pp. 936-949, 2019.Article (CrossRef Link) https://doi.org/10.1109/TNSM.2019.2929425.

[25] Nayak, A. (2024). Bio-inspired edge intelligence: Neuromorphic architectures for real-time biomedical signal classification. Electronics, Communications, and Computing Summit, 2(4), 32–41.

[26] Rahim, R. (2025). AI-Driven Fault Diagnosis in Three-Phase Induction Motors Using Vibration and Thermal Data. National Journal of Electrical Machines & Power Conversion, 21-28.

[27] Romano, G., & Conti, A. (2024). The role of Customer Feedback Loops in driving Continuous Innovation and Quality Improvement. National Journal of Quality, Innovation, and Business Excellence, 1(2), 30-39.
[28] Martínez, G. (2024). Cultural Heritage Tourism: Balancing Preservation with Visitor Experience. Journal of Tourism, Culture, and Management Studies, 1(2), 17-27.
[29] Ramchurn, R. (2025). Advancing autonomous vehicle technology: Embedded systems prototyping and validation. SCCTS Journal of Embedded Systems Design and Applications, 2(2), 56–64.
[30] Manaa Barhoumi, E., Charabi, Y., & Farhani, S. (2023). FPGA Application: Realization of IIR Filter Based Architecture. Journal of VLSI Circuits and Systems, 5(2), 29–35. https://doi.org/10.31838/jvcs/05.02.05
[31] Karthika, J. (2025). Sparse signal recovery via reinforcement-learned basis selection in wireless sensor networks. National Journal of Signal and Image Processing, 1(1), 44–51.
[32] Kavitha, M. (2025). Deep learning-based channel estimation for massive MIMO systems. National Journal of RF Circuits and Wireless Systems, 2(2), 1–7.
[33] Veerappan, S. (2025). Harmonic feature extraction and deep fusion networks for music genre classification. National Journal of Speech and Audio Processing, 1(1), 37–44.
[34] Sadulla, S. (2025). IoT-enabled smart buildings: A sustainable approach for energy management. National Journal of Electrical Electronics and Automation Technologies, 1(1), 14–23.