*Computer Systems Science & Engineering*
DOI: 10.32604/csse.2023.023378
***Article***

Tech Science Press

# Fuzzy User Access Trust Model for Cloud Access Control

**Aakib Jawed Khan*** **and Shabana Mehfuz**

Department of Electrical Engineering, Jamia Millia Islamia, New Delhi, 110025, India
*Corresponding Author: Aakib Jawed Khan. Email: aakibjawed@gmail.com

Received: 06 September 2021; Accepted: 29 November 2021

**Abstract:** Cloud computing belongs to a set of policies, protocols, technologies through which one can access shared resources such as storage, applications, networks, and services at relatively low cost. Despite the tremendous advantages of cloud computing, one big threat which must be taken care of is data security in the cloud. There are a dozen of threats that we are being exposed to while availing cloud services. Insufficient identity and access management, insecure interfaces and Applications interfaces (APIs), hijacking, advanced persistent threats, data threats, and many more are certain security issues with the cloud platform. APIs and service providers face a huge challenge to ensure the security and integrity of both network and data. To overcome these challenges access control mechanisms are employed. Traditional access control mechanisms fail to monitor the user operations on the cloud platform and are prone to attacks like IP spoofing and other attacks that impact the integrity of the data. For ensuring data integrity on cloud platforms, access control mechanisms should go beyond authentication, identification, and authorization. Thus, in this work, a trust-based access control mechanism is proposed that analyzes the data of the user behavior, network behavior, demand behavior, and security behavior for computing trust value before granting user access. The method that computes the final trust value makes use of the fuzzy logic algorithm. The trust value-based policies are defined for the access control mechanism and based on the trust value outcome the access control is granted or denied.

**Keywords:** Cloud architecture; fuzzy logic; trust-based access mechanism

## 1 Introduction

Most cloud inferences in the present days provide APIs as a means of management and interaction of the inferences. However, this raises concerns in terms of the susceptibility of the cloud server in respect to encryption, activity monitoring, and authentication. It can be further noted that the multi-tenancy in cloud computing raises concerns for increased system vulnerabilities, even though this aspect is managed amongst other IT needs of the companies.

Even as organizations evolve their understanding and application of computing servers and software, there is also a rise in the technical skills of malicious insiders and hijackers. This, in turn, raises concerns for cyber-attacks and advanced persistent threats (APTs), through the means of spear phishing, malware-loaded USB drives, compromised third-party networks, and direct attacks amongst others. Other threats in this respect include direct attacks and possible loss of all data, inability to access the scope of infiltration through internal stakeholders in respect to data manipulation and stealing, DoS attacks, and service abuse. It is reflected that the premise of cloud computing is sharing of technology to increase innovation, overcome infrastructural limitations and facilitate growth through increased server interaction. However, this sharing also raises concerns with respect to system over-ride and shared business risks for many organizations. Reflecting upon the challenges of the cloud, researchers have identified that multiple layers are the reasons for increased vulnerabilities of the service, and also that the system offers multiple layers which serve to enhance the attack surface. Understandably the process has limitations in the form of SQL injection flaws, challenges associated with shared resources, data breaches, insecure application programming interfaces, and misconfiguration issues.

The biggest challenge faced by cloud applications and cloud service providers is ensuring 100% secure authentication, authorization, and data protection. The focus of this work is the access control mechanism. An access control mechanism's main objective is to ensure that access to the cloud services and their key resources is granted to an authorized person only. Thus, the access control mechanism decides whether the user has the needed permission to access the resource or modify it. For the past decade, Privilege Access Management (PAM) has been treated as a solution for only those scenarios when internal technical professional employees (System Administrators, DBAs) need to interact with data center servers (Unix boxes, Oracle databases) to execute patching, upgrade kind of activity on the native OS, databases or other installed applications. PAM scenarios during these interactions and also available solutions have matured a lot. However, with the wide adoption of cloud platforms like AWS and Azure and the need to have a completely automated DevOps pipeline, the interactions that technical professionals have with PaaS, SaaS, and mainly IaaS platforms, have undergone drastic changes. Numerous access control mechanisms have been proposed for normal cloud service users as well as for privilege access management. But each mechanism has its challenges which leave the scope for improvement. The traditional access control mechanisms are mostly based on policies that monitor access logs and fail to ensure that the data flow among the user and the service provider can be trusted or not. To ensure data integrity there is a need to monitor user behavior. Thus, this work extends the existing access control mechanism with a novel model which uses concepts of fuzzy logic to come up with a robust access control mechanism.

The present work reflects upon the challenges and security issues of the Cloud Computing Systems and demonstrates an access control mechanism for generic as well as the privileged user of the cloud applications. The access control mechanism is mainly based on user behaviour, network behaviour, and control policies.

## 2 Cloud Security Challenges

Fig. 1 summarizes the main aspects of the cloud paradigm. The major aspects which make cloud computing a successful computational model are the layered architecture of the cloud; its on-demand self-service capabilities, and the various types of service models [1].

The cloud architecture is a layered architecture comprising of four layers, the hardware layer consisting of the datacentres, the infrastructure layer the platform layer, and the application layer. All together this layer functions as a computing platform serving each other layer to provide storage and processing abilities to applications and services. Cloud services can be classified as Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS).

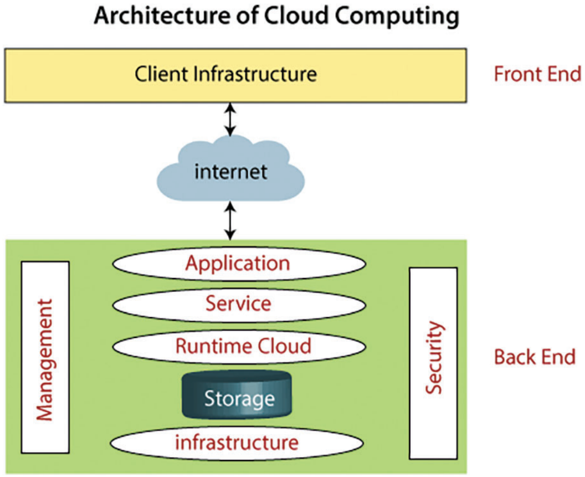


**Figure 1:** Overall cloud paradigm

In the Software as a Service (SaaS) cloud services the applications are generally made accessible through thin clients or web browsers and these applications are running on the cloud environment. The Platform as a Service (PaaS) provides common system support-based services like operating system support services, software development framework-based services to its clients. In the Infrastructure as a Service (IaaS) based services the cloud provides processing abilities, storage capabilities, and network resources to the users [1]. Fig. 2 summaries the security challenges in the cloud.

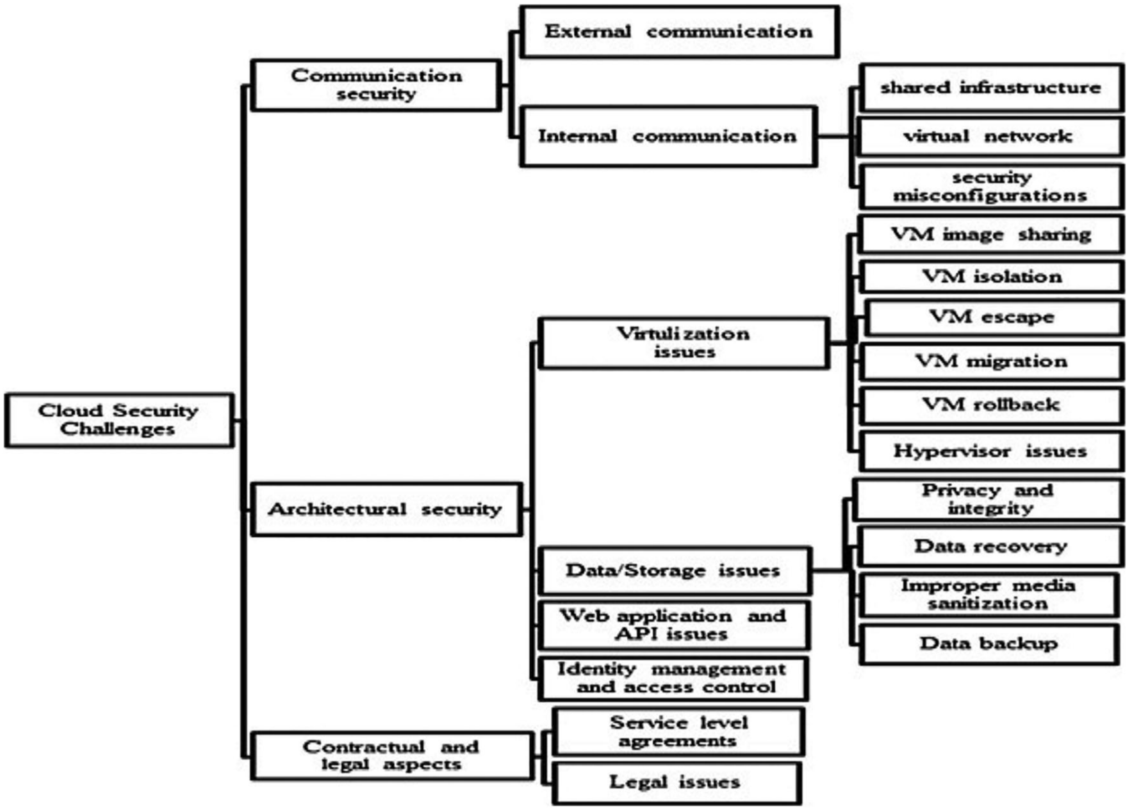


**Figure 2:** Cloud security challenges

Organizations that have already deployed cloud computing face certain difficulties; especially they have many constraints in implementation. Major challenges faced by organizations in making use of cloud computing are as follows:

I. Lack of trust: Customers still feel that cloud providers may not guarantee the security and privacy controls as needed by the organizations. The survey result shows that security and privacy are the major concerns [2].

II. Resistance to changing the traditional working styles [1].

III. In some cases, cloud computing service providers don't provide privileges to audit the data. Such results in a lack of governance [3].

IV. Due to the rise in the number of vendors in the market, some vendors over-commit to resources and functionalities [4]. But to maintain their revenue in profit, these vendors cut short some value system services like limit cloud access, etc. This results in customer dissatisfaction. Thus, it is necessary to have a good thorough check up of vendor services before deploying [3–5].

V. Security issues like network breaks, poor encryption key usage and management, public management interface, separation failure, and privilege abuse also exist [5,6].

## 3 API and Access Control Literature Review

An API is a set of functions or routines that accomplishes specific tasks or provides a simplified method of interacting with a software component, often allowing the automation of common processes that interact with services running on other machines. APIs can be in the form of a library that includes specifications for routines, data structures, object classes, and variables, or simply a specification of remote calls exposed to the API consumer [7]. Some APIs are based on international standards such as POSIX (Portable Operating System Interface), while others are made public in open source or vendor documentation. For example, Microsoft's Windows API enables developers to create software for the Windows platform [7].

The API makes the cloud computing process easier and aids in automating complex business requirements such as configuring cloud instances for multiple providers and use of third-party platforms for cloud and on-premises needs. The confidentiality, integrity, availability, and accountability can be harmed if the API is insecure. Choosing a set of secured channels, providing proper authentication and authorization, enabling message protection, and delivering the best code and development practices are some key areas that should be focused on for API security [8]. PAM (privileged access management) often known as privileged Account Management or Privileged Session management allows a user to access or perform critical functions on the server. To safeguard data and prevent vulnerabilities PAM is preferred in an organization especially when the employees are huge in number [9,10].

The basic functions delivered by PAM are that it grants privileges only to the systems where the users have authentication [11]. They see to it that the access is granted only when it is needed and revoked once the session expires. The privileged users need not have local or system passwords, thereby saving time. PAM can manage heterogeneous systems and can provide central access to the users [12]. It runs an audit trail for every privileged operation which cannot be altered. The entire architecture of PAM can be broken down into different components namely:

- Access manager acts as a single point defining policy and reinforcement for access management.
- The super admin can manage the accounts in the access manager [13].
- Next is the password vault by which the access can be controlled towards critical systems.
- The actions taken during a privileged session are tracked by a session manager [14].

Access control mechanisms are specially designed programs that restrict user access to key resources [15]. Generally, all access control mechanism records the user access logs and their attempts to the services. Through these records, it is capable of finding unauthorized access [16]. In context to cloud-based platforms access control mechanism also performs authentication and user identification before granting permission of access to the key resources.

The common access control mechanism in a cloud-based environment is Mandatory access control, discretionary access control, role-based access control, and attribute-based access control mechanism [17]. These are the traditional access control mechanism employed in a cloud-based environment [18]. Over the years numerous researches have been carried out in access control mechanisms concerning cloud-based platforms and successful extension of each model has been done. For example, the context-aware access control mechanism is used to manage vital and sensitive information is an extension of the role-based access control mechanism [18,19]. It makes use of contextual conditions before deciding to limit the data access or allowing the access.

Similarly, ontology-based access control mechanism which makes use of ontological techniques were also used [20]. These again made use of role-based access control mechanisms with ontological extension for making access policies [20]. Cryptography based access control and data a security in cloud have been under constant research. [21] Proposed a new steganography-based access control model that used Improved Key generation scheme of RSA for encrypting data over cloud and substring indexing and keyword search mechanism for extracting the encrypted stored data. The research compared its proposed method with the proposed method and compare with existing cryptographic methods over New York State Department of Health dataset [21]. The outcome of the research proved that the new steganography-based access control model enhanced cloud security and helped in efficient retrieval of data [21]. Tab. 1 shows the Access control mechanism like Concept, advantages, and Challenges etc.

With an objective of improving the accessibility of the data accessibility [22] introduced the concept of reputation valuation based on user roles and access management based on tasks. With input parameters from user tasks, permission, roles, sessions and data object under access a trust value is generated. This new reputation valuation based trust value access control mechanism is capable of reducing illegal access and has proven to enhance the security of the cloud data [22]. This research work motivates in moving ahead with new and more enhanced trust based access control mechanism for enhancing cloud access control and data a security.

**Table 1:** Access control mechanism: concept, advantages, and challenges

| Access control mechanism | Concept | Advantages | Challenges disadvantages |
|---|---|---|---|
| Mandatory access control | It is based on security attributes. | Checks that both subject and object has security permission or not. Good for military and highly secure applications [23]. | Time-consuming, difficult and complex [11]. |
| Discretionary access control | Based on a chain-based access mechanism where the owner of the subject decides on access permission and can grant other subjects [11]. | Greater flexibility [23]. | Low-security performance due to no control on subjects gaining authority via chained users [23]. |

(Continued)

**Table 1 (continued)**

| Access control mechanism | Concept | Advantages | Challenges disadvantages |
|---|---|---|---|
| Role-based access control | Defines roles and assign users with roles. Each role has defined permission [12]. | Reduces authority management complexities. Eliminates the direct mapping of user and authority. Enhances flexibility [23,24]. | It only verifies user identity before assigning a role. Any user with valid permissions can access and thus it fails to monitor user actions [11]. |

The role-based access control mechanism is suitable for the closed and centralized working environment. It is not suitable for open distributed networks [23]. To overcome the challenges in the traditional access control mechanism, a trust-based access control mechanism were proposed and implemented. Tang et al. [18] proposed the "trust management" concept into the access control mechanism. For granting access to the services the trust values are computed. To compute the trust values the user behavior is assessed [17]. The roles are defined and the authentication based on roles is done. But each user role is defined based on the computed trust value. A trust model measures the security strength and computes a trust value. Thus, this mechanism is more safe and reasonable for assigning the requisite amount of user's required authority. The model is depicted in Fig. 3.

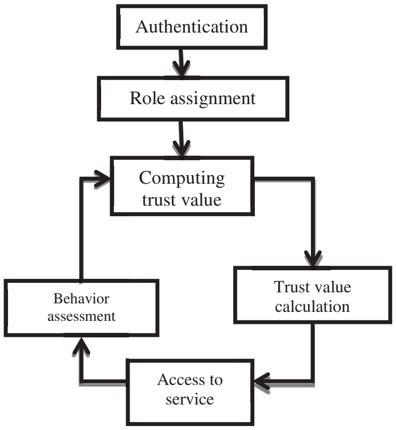


**Figure 3:** Trust-based access control model

A key challenge in traditional access control methods is that they only guarantee access to the cloud services and resources to the users but they fail to analyze the operations performed by the users. Failure of doing so can lead to security attacks like IP spoofing, phishing and can cause an impact on the integrity of the information.

## 4 Proposed Work

From the literature survey, it is evident that there is a need for an access control method that shall ensure data integrity and this thus requires access control mechanisms that are capable of monitoring user behavior. Such an access control mechanism can be used to secure the privilege access management as well. The approach to finding a solution begins with the usage patterns of cloud computing users as well as the behaviour of the service provider.

The main advantage of analyzing usage patterns is that feasible solutions developed can be implemented irrespective of the software, languages, and middleware [9]. There are some cloud models proposed by the research team like Certificate Status Authority (CSA), National Institute of Standards and Technology (NIST) [17]. The usage pattern analysis is one way of extending the model and finding a deployment method and a model which can fit a particular application [9]. There are several compound patterns in cloud computing. The main application of this pattern analysis is when the number of users accessing the cloud is huge. An application of this pattern analysis is in facebook.salesforce.com and other social networking sites. The outcome of such pattern analysis is to build a model that can decide when to grant and when to drop the access request [17].

Automating such a model further eliminates human effort to calculate and take decisions. When the number of requests served increases, the system examines the usage pattern and upon reaching a finite limit, the access must be prioritized and served [17]. Controlling the access intelligently can result in a better user experience. The granting and dropping mechanism should be administrated properly to implement an efficient solution [9]. The number of access must be increased only in case of emergency and that too in a controlled manner. It should not be held for too long, a drop back mechanism should work in parallel to bring down the access once the emergency is being served. Building a model using cloud user and service provider usage patterns, automating it, and having administrative control over it can deliver a solution for access management and a better user experience.

Thus, in this work, a key factor of user and service provider usage behavior is considered and that is the trust value. The trust value between user and service provider enables the API to grant access to the user. This value changes dynamically from time to time. This research work makes use of user behaviour patterns. These patterns are passed over a fuzzy logic based model that calculates user access trust value. This fuzzy logic model is named as user access trust value calculation and classification algorithm.

This user access trust value calculation and classification algorithm makes use of the concepts of the Mamdani Fuzzy logic system to generate and classify the trust values. These classified trust values are then used to facilitate the decision of granting access or revoking the access based on its outcome.

**Algorithm:**

The primary feature of this model is the user access trust value calculation & classification algorithm. The basic working of this algorithm is as follows:

A user when wants to access the system sends an access request to the cloud service provider. The API forwards this request to the user access trust value calculation & classification model.

```
Begin:
Step 1: User sends the request for accessing the cloud or to know the trust value of cloud service providers.
Step 2: The User access trust value calculation and classification center shall analyze the user request based on user behavior patterns, history of log sessions, and control policies set for the user. The classified trust value is sent to the trust value database.
Step 3: The trust management center then uses this trust value to determine the value is acceptable and whether the user access can be granted or denied.
Step 4: The user is notified with a response.
End
```

The user access Trust value calculation and classification model is using fuzzy logic concepts to compute the correct trust value as depicted in Fig. 4. User access Trust value parameters considered are as follows:

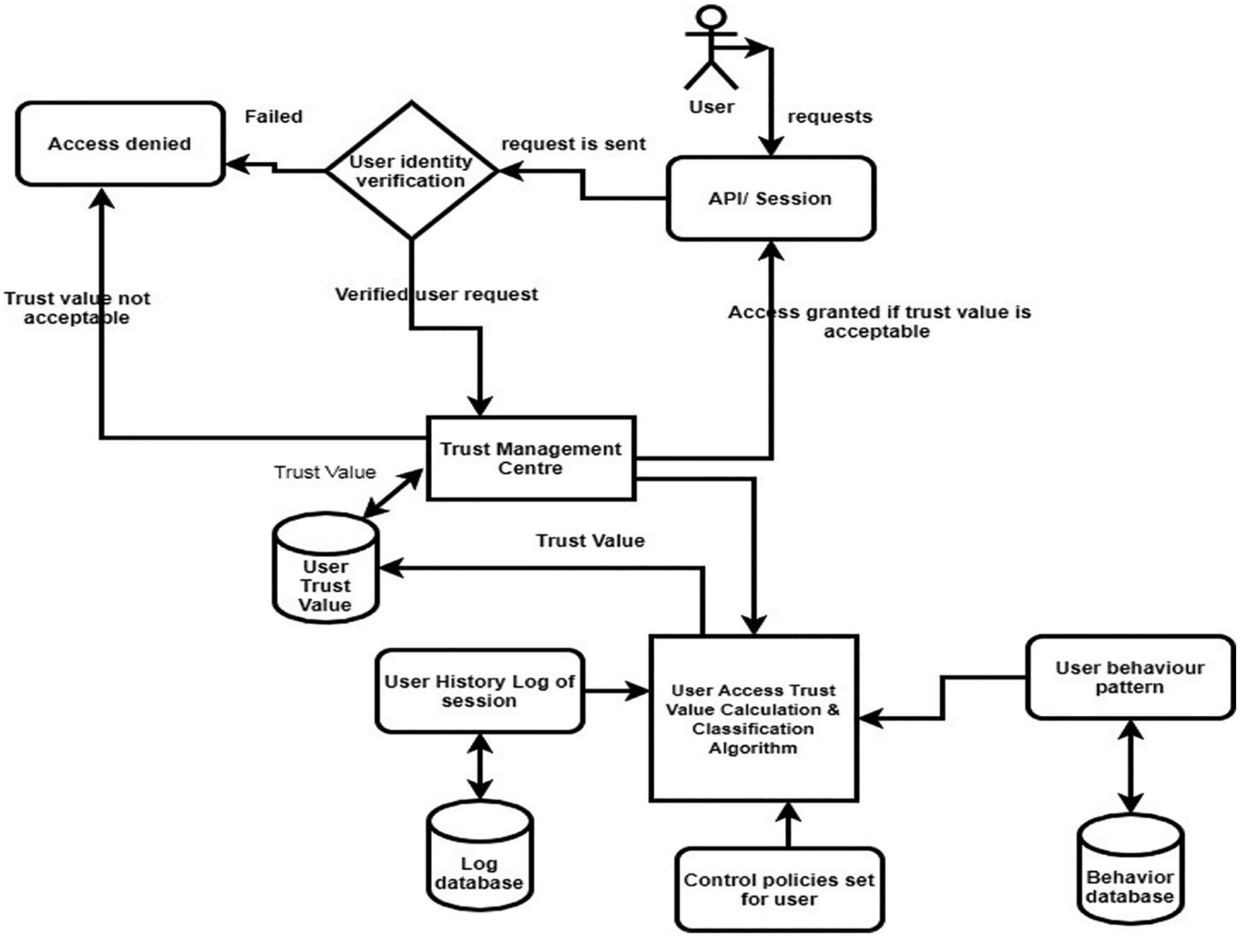

User
requests
API/ Session
request is sent
User identity verification
Failed
Access denied
Verified user request
Trust value not acceptable
Access granted if trust value is acceptable
Trust Management Centre
Trust Value
User Trust Value
Trust Value
User History Log of session
Log database
User Access Trust Value Calculation & Classification Algorithm
Control policies set for user
User behaviour pattern
Behavior database

**Figure 4:** User access trust value calculation and classification model

- Control policies (CP)
- Security levels (SL): Set by the SLA admin
- Previous Workload Time (WT)
- Session time (ST)
- Response time (RT)
- Usability (U)
- Availability (A)
- Scalability (SC)
- Number of sessions (NS) granted
- Number of session Denied (ND)
- Bad request (BR)

All these above parameters can be grouped as:

1. Security parameters (SP): Control Policy and Security Levels
2. Work Parameters (WP): Previous workload Time, Response Time Usability, Availability, Scalability.

3. Demand Parameters (DP): Session Time
4. Good record (URG): Number of sessions granted
5. Bad record: Number of sessions Denied and Bad request (URB)

Control Policy: These are defining the control policies for the user based on the user roles.

Security Levels: These are set by the cloud network administrator for the user based on the user roles and permission. These can be low, medium, or high levels.

Previous Workload Time: This tells us about the workload of the processing time spent by the user on the cloud.

Session Time: The time taken for the previous user session.

Usability: It defines the ease with which the cloud can provide the user its expected goals.

Availability: It defines how the network is available for the user.

Scalability: It defines the network's ability to adjust to the growing user demands.

Number of Session Granted: These are generated from previous records and provide an estimate on the number of sessions granted to this user previously.

Number of Sessions Denied: how many sessions were denied to the user due to wrong password details, or unauthorized attempts, etc.

Bad Request: Number of bad requests coming from unknown sources and with wrong credentials.

Based on [25] trust value evaluation strategy we compute trust value as:

Trust Value $= [w_1\ (SP + WP + DP + URG) - w_2\ (URB)]$

Where $w_1$, $w_2$ are the fuzzy weight factors such that $w_1 + w_2 = 1$.

The outcome is the trust value and and classification into various groups. The groups are:

- High Trust
- Mean Trust
- NoTrust

## 5 Experimentation

The rules for the fuzzy system have been developed and the model is implemented in MATLAB fuzzy inference system. MATLAB fuzzy inferenence system toolbox is simulated on a 1.70 GHz, intel Core i3 based computing system with 1TB of hard disk, 4 GB of RAM and Windows 7 (64 Bit) Operating Systems. The parameters considered in this work are mapped onto the MATLAB toolbox fuzzy inference system with their membership function values. For this work, the Mamdani Fuzzy inference system is selected.

### *5.1 Input Variables*

Membership functions for all input parameters were developed. For example, membership function for control Policy is given in Fig. 5. These are defining the control policies for the user based on the user roles. These are categorized as low level, medium level and high level. Similarly Fig. 6 gives membership functions for bad records. These are categorized as low level, medium level and high level.

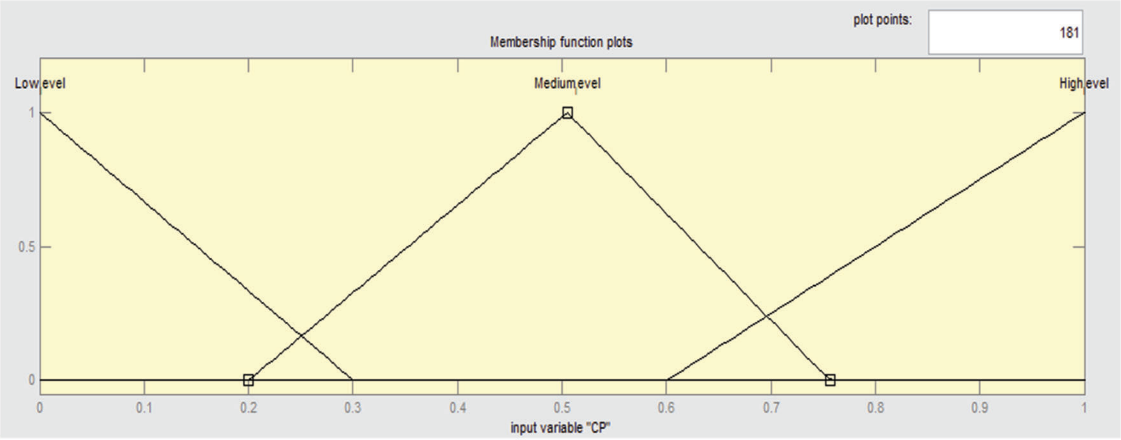


**Figure 5:** Control policy membership functions

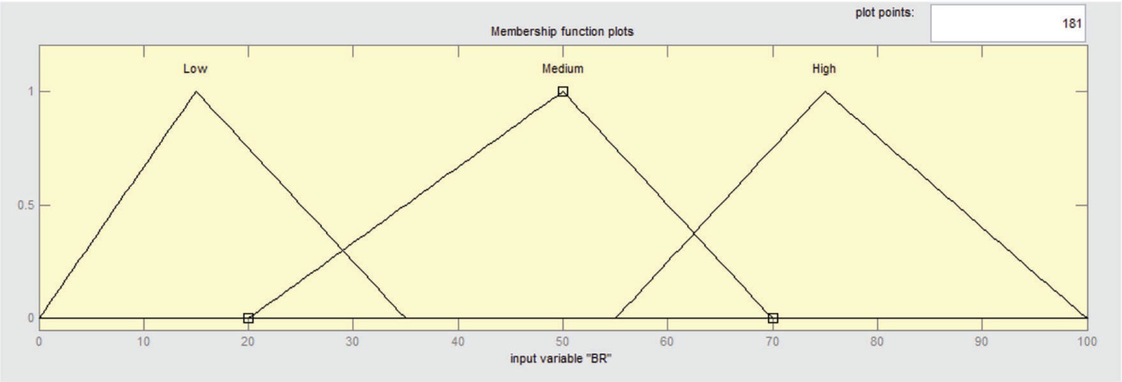


**Figure 6:** Membership functions for bad record

### *5.2 Output Variable*

Trust Value: This is the final output value and it is classified into three levels namely: No Trust, Medium Trust, and High Trust. It is displayed in Fig. 7.

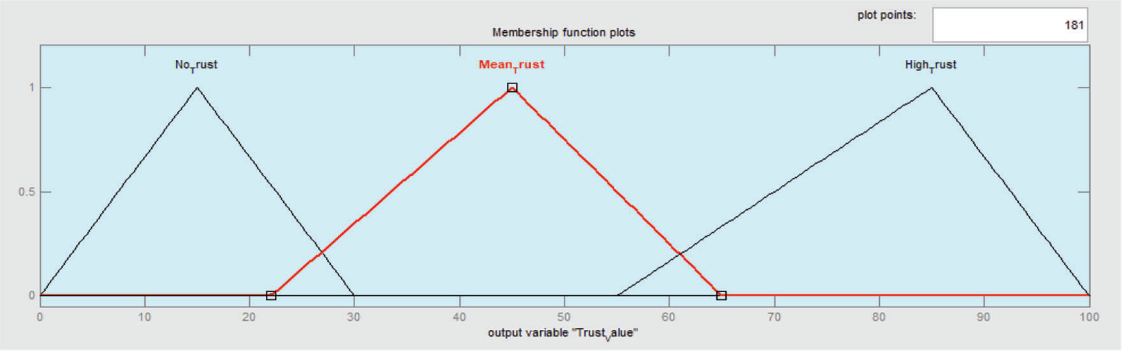


**Figure 7:** Output parameter

### *5.3 Adding Rules*

The rules for each parameter are added in the rule editor box as per the security policies. The various input parameters are:

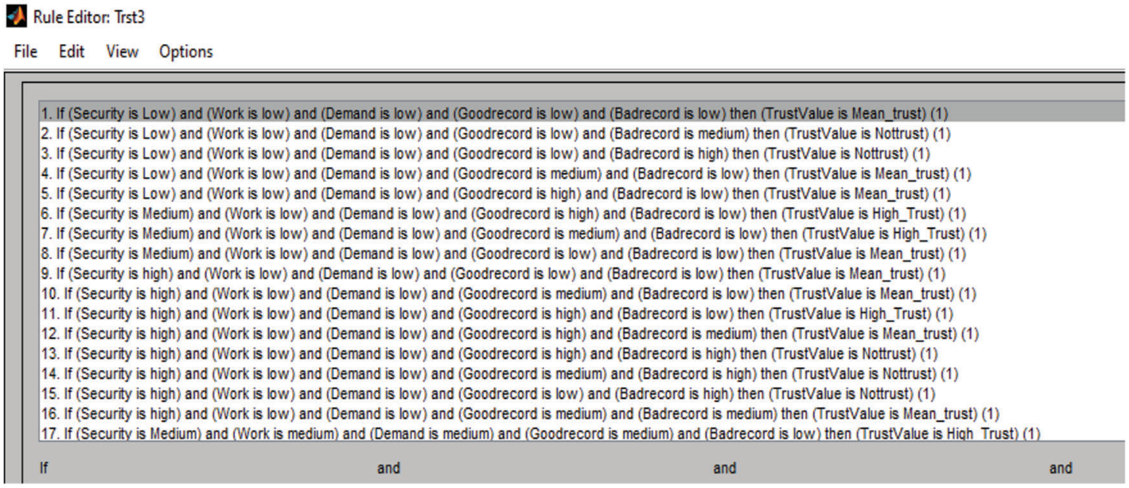
1. Security parameters (SP): Control Policy and Security Levels
2. Work Parameters (WP): Previous workload Time, Response Time Usability, Availability, Scalability.
3. Demand Parameters (DP): Session Time
4. Good record (URG): Number of sessions granted
5. Bad record: Number of sessions Denied and Bad request (URB) and output parameter is TrustValue.

Then a simple rule could be as follows:

Rule 1: If the values of SP, WP, DP, URG, URB are all low then TrustValue is No Trust or PoorTrust.

Rule 2: if SP value is High, WP value is medium, DP value is medium, URG value is High and URB value is low Then the TrustValue is Hightrust.

Thus, the rules for the system are given in Fig. 8.



```
1. If (Security is Low) and (Work is low) and (Demand is low) and (Goodrecord is low) and (Badrecord is low) then (TrustValue is Mean_trust) (1)
2. If (Security is Low) and (Work is low) and (Demand is low) and (Goodrecord is low) and (Badrecord is medium) then (TrustValue is Nottrust) (1)
3. If (Security is Low) and (Work is low) and (Demand is low) and (Goodrecord is low) and (Badrecord is high) then (TrustValue is Nottrust) (1)
4. If (Security is Low) and (Work is low) and (Demand is low) and (Goodrecord is medium) and (Badrecord is low) then (TrustValue is Mean_trust) (1)
5. If (Security is Low) and (Work is low) and (Demand is low) and (Goodrecord is high) and (Badrecord is low) then (TrustValue is Mean_trust) (1)
6. If (Security is Medium) and (Work is low) and (Demand is low) and (Goodrecord is high) and (Badrecord is low) then (TrustValue is High_Trust) (1)
7. If (Security is Medium) and (Work is low) and (Demand is low) and (Goodrecord is medium) and (Badrecord is low) then (TrustValue is High_Trust) (1)
8. If (Security is Medium) and (Work is low) and (Demand is low) and (Goodrecord is low) and (Badrecord is low) then (TrustValue is Mean_trust) (1)
9. If (Security is high) and (Work is low) and (Demand is low) and (Goodrecord is low) and (Badrecord is low) then (TrustValue is Mean_trust) (1)
10. If (Security is high) and (Work is low) and (Demand is low) and (Goodrecord is medium) and (Badrecord is low) then (TrustValue is Mean_trust) (1)
11. If (Security is high) and (Work is low) and (Demand is low) and (Goodrecord is high) and (Badrecord is low) then (TrustValue is High_Trust) (1)
12. If (Security is high) and (Work is low) and (Demand is low) and (Goodrecord is high) and (Badrecord is medium) then (TrustValue is Mean_trust) (1)
13. If (Security is high) and (Work is low) and (Demand is low) and (Goodrecord is high) and (Badrecord is high) then (TrustValue is Nottrust) (1)
14. If (Security is high) and (Work is low) and (Demand is low) and (Goodrecord is medium) and (Badrecord is high) then (TrustValue is Nottrust) (1)
15. If (Security is high) and (Work is low) and (Demand is low) and (Goodrecord is low) and (Badrecord is high) then (TrustValue is Nottrust) (1)
16. If (Security is high) and (Work is low) and (Demand is low) and (Goodrecord is medium) and (Badrecord is medium) then (TrustValue is Mean_trust) (1)
17. If (Security is Medium) and (Work is medium) and (Demand is medium) and (Goodrecord is medium) and (Badrecord is low) then (TrustValue is High_Trust) (1)
```

**Figure 8:** Rules

## 6 Results

Once the rules are formulated, the rules are triggered to compute the set of values for any dataset. For computation, 10 user dataset is used (Tab. 2). These 10 users' data contains values for each parameter and the actual trust value.

**Table 2:** Calculation of relative error

| User | Actual trust | Predicted trust | RE | MMRE |
|---|---|---|---|---|
| 1 | 45 | 55.5 | 0.2 | 17.0853 |
| 2 | 5 | 6.67 | 0.3 | |
| 3 | 32 | 34.4 | 0.1 | |

(Continued)

**Table 2** (continued)

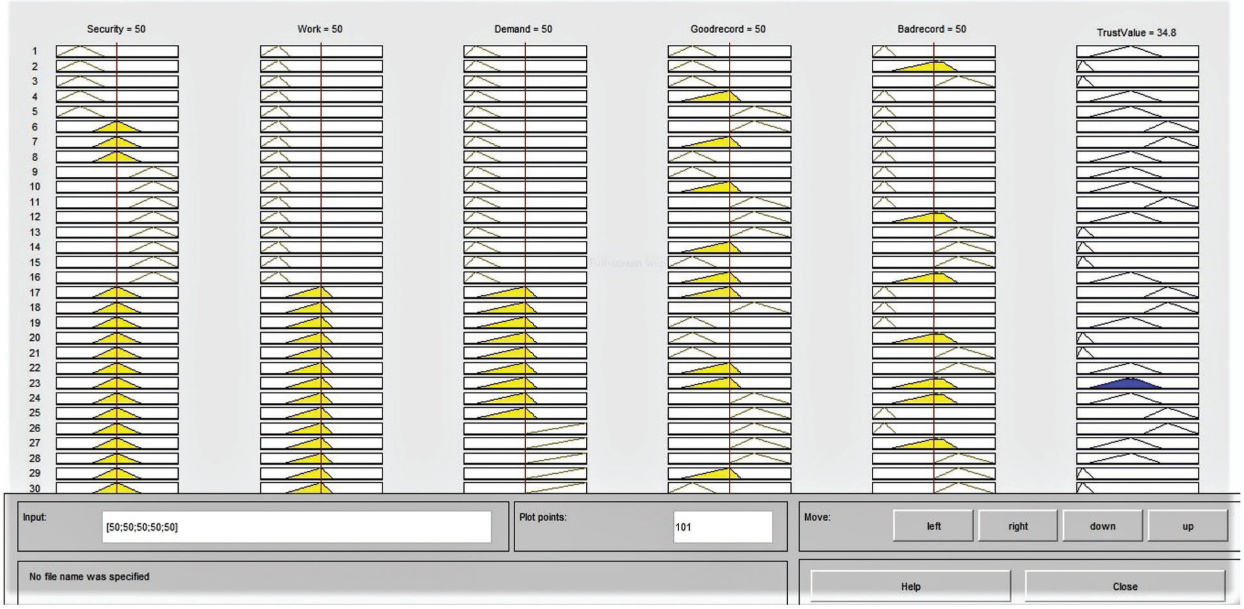
| User | Actual trust | Predicted trust | RE | MMRE |
|---|---|---|---|---|
| 4 | 45 | 34.6 | 0.2 | |
| 5 | 48 | 55.2 | 0.2 | |
| 6 | 30 | 34.6 | 0.2 | |
| 7 | 27 | 31.3 | 0.2 | |
| 8 | 75 | 76.7 | 0.0 | |
| 9 | 55 | 69.6 | 0.3 | |
| 10 | 32 | 34.7 | 0.1 | |

For each user, the data set values are used to compute fuzzy-based trust values. Once the fuzzy-based trust values are computed then relative error between the actual value and the fuzzy-based value is computed.

The relative error between the actual trust value and the predicted trust value from our model is calculated as follows:

$$\text{Relative Error RE} = \text{Absolute value}\left(\frac{\text{Estimated trust value} - \text{Actual trust value}}{\text{Actual value}}\right) \tag{1}$$

The triggering of rules for a data set is given in Fig. 9.



Input: [50;50;50;50;50]

Plot points: 101

No file name was specified

**Figure 9:** Rules triggered

Surface plots are analyzed to understand the relationship among the various parameters. The surface plot for control policies and security levels is directly proportional to the trust value. Security parameters and bad records depicted in Fig. 10 that security parameters are having a direct impact on bad records. As bad records value are high the security levels are low. As Bad request and No of session Denied concerning Trust Value increase the trust value decreases. This is depicted in the Fig. 11.

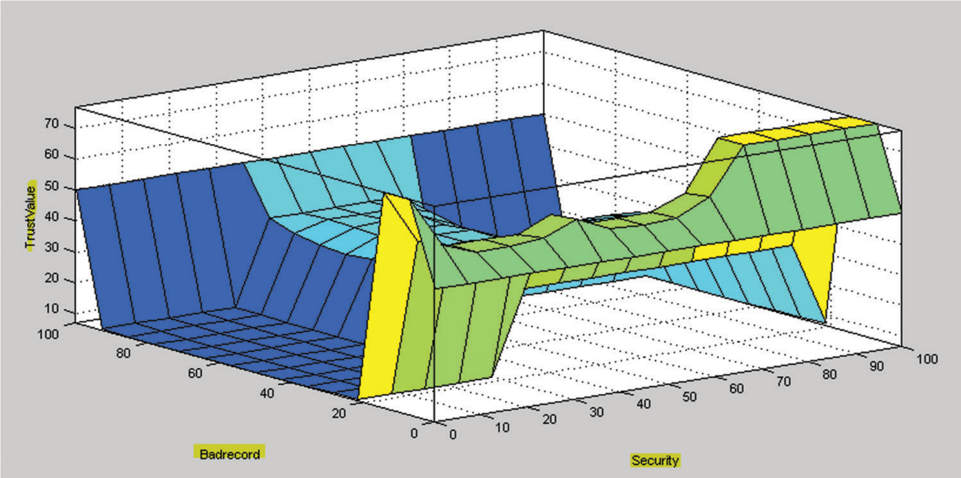


**Figure 10:** Security parameters and bad records

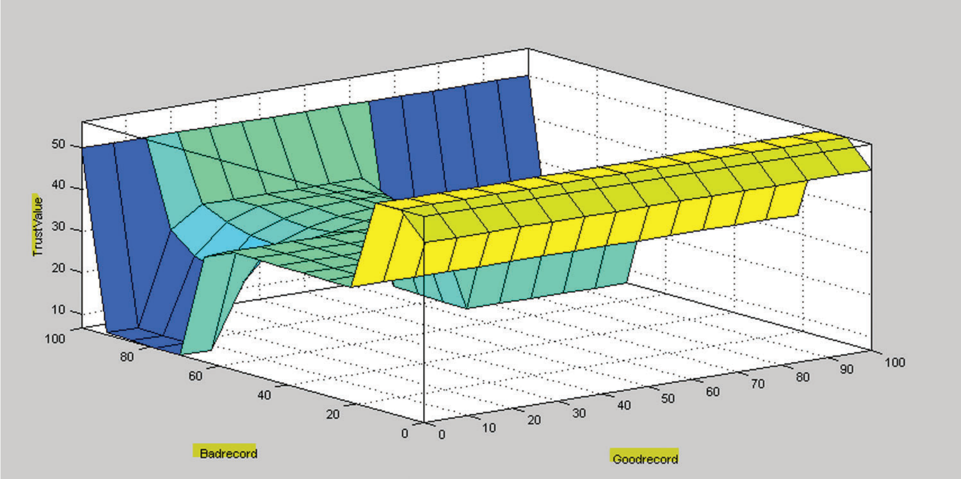


**Figure 11:** Good record *vs.* bad record

For evaluation, the trust values of 10 users were used as the data set. The data set had the actual trust value and this was used for comparison with our model. Evaluation was done by calculating the relative error among the actual and the predicted trust values. From the relative error the Magnitude of relative error is calculated as relative error *100. Then the Mean magnitude of relative error (MMRE) value is calculated from the magnitude relative error using the below formula.

MMRE is calculated as:

$$\text{MMRE} = \frac{Sum\ of\ all\ magnitude\ relative\ error}{number\ of\ users} \tag{2}$$

For the proposed work, the MMRE value computed is 17.0853. The dataset, the actual Trust value, the predicted trust value, the relative error (RE) and the Mean Magnitude of Relative Error (MMRE) is given Tab. 2.

The trust value of proposed model with respect to the original (actual) model is plotted in the graph shown in Fig. 12.

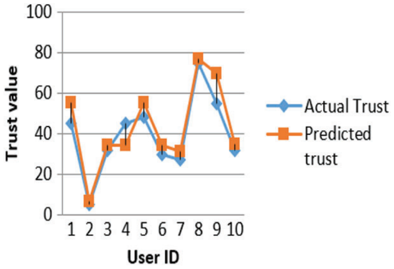


**Figure 12:** Actual and predicted trust values

It was observed that as the number of bad requests and the number of requests denied increases; the trust value of proposed model decreases. The trust value is less than the actual trust value. For users with good behaviour who have a very few bad requests and very little count of sessions denied the trust value is less than the actual value. But for the same users, if the bad request or number of sessions denied increases, the trust value decreases. Thus, proposed model is providing a signal to the trust model that despite the user having good records or behaviour any increase in the count of bad request or any increase in the count of sessions denied is considered as an attempt to breach the security of the user account and this decreases the trust value. This decreased trust value should be considered as an alarm signal to the devices/algorithms like intrusion detection system or access control mechanism so that they can monitor the user activities. The message could be the recommendation of changing passwords or access control rules or changes in security policies based on roles. Thus, the proposed model is progressive and can be integrated with other security models to deliver an enhanced solution.

## 7 Conclusion

The objective of this research was to examine the state of the art of cloud computing security challenges with respect to the user perspective. It was found that the biggest challenge is ensuring 100% data integrity and security in terms of access control. Various measures are used to ensure access control. Among the methods used by access control are the role-based access control methods. Various models for role-based access control have emerged. These rely on trust values. But most of the trust values computed are either in the form of signatures of certificates and they fail to address user behaviour and network demands or patterns. Thus, this work makes use of an integrated model and applies fuzzy logic to come up with an integrated enhanced model that considers security patterns, cloud network patterns, user behaviour, and demand patterns to compute user trust value. This is used by the trust centre to grant request or deny the request. The findings of the result are that the proposed model outperforms the other existing model and the actual trust values differ with a Mean Magnitude of relative error by 17.0853 which is better than the existing KNN models. This research work opens a new door for futuristic research on integrating all the primary parameters for computing trust values and rendering access control, a higher degree of security.

**Acknowledgement:** The authors want to express their gratitude to the reviewers for their careful and meticulous reading of the paper.

**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] M. S. Babu, A. M. Babu and M. C. Sekhar, “Enterprise risk management integrated framework for cloud computing,” *Int. J. Advanced Networking and Applications*, vol. 5, no. 3, pp. 1939–1950, 2013.

[2] A. Bisong and S. S. M. Rahman, “An overview of the security concerns in enterprise cloud computing,” *International Journal of Network Security & Its Applications*, vol. 3, no. 1, pp. 30–45, 2011.

[3] A. Agarwal and A. Aggarwal, “The security risks associated with cloud computing,” *International Journal of Computer Applications in Engineering Sciences*, vol. 1, special issue on CNS, pp. 257–259, 2011.

[4] A. Kayes, J. Han, W. Rahayu, T. Dillon, M. S. Islam *et al.,* “A policy model and framework for context-aware access control to information resources,” *The Computer Journal*, vol. 62, no. 5, pp. 670–705, 2018.

[5] K. Hamlen, M. Kantarcioglu, L. Khan and V. Thuraisingham, “Security issues for cloud computing,” *International Journal of Information Security and Privacy*, vol. 4, no. 2, pp. 39–51, 2010.

[6] G. Raghvender, D. Lakshmi and S. Venkatshwearlu,“Security issues and trends in cloud computing,” *International Journal of Computer Science and Information Technologies*, vol. 6, no. 2, pp. 1156–1159, 2015.

[7] L. Barthelus, “Adopting cloud computing within the healthcare industry: Opportunity or risk,” *Online Journal of Applied Knowledge Management*, vol. 4, no. 1, pp. 1–16, 2016.

[8] N. Gonzalez, C. Miers, F. Redigolo, M. Simplicio, T. Carvalho *et al.,* “A quantitative analysis of current security concerns and solutions for cloud computing,” *Journal of Cloud Computing*, vol. 1, no. 11, pp. 1–18, 2012.

[9] L. Tian, C. Lin and Y. Ni, “Evaluation of user behaviour trust in cloud computing,” in *Int. Conf. on Computer Application and System Modeling (ICCASM)*, Taiyuan, China, vol. 7, pp. 567–572, 2010.

[10] S. M. Habib, S. Ries and M. Muhlhauser, “Towards a trust management system for cloud computing,” in *IEEE 10th Int. Conf. on Trust, Security, and Privacy in Computing and Communications*, Changsha, China, 2011, pp. 933–939, 2011.

[11] W. Li, L. Ping, Q. Qiu and Q. Zhang, “Research on trust management strategies in cloud computing environment,” *Journal of Computational Information System*, vol. 8, no. 4, pp. 1757–1763, 2012.

[12] P. K. Behera and P. M. Khilar, “A novel trust-based access control model for cloud environment,” in *The Proc. of Int. Conf. on Signal, Networks, Computing, and Systems*, India, pp. 285–295, 2017.

[13] K. Ren, C. Wang and Q. Wang, “Security challenges for the public cloud,” *IEEE Internet Computing*, vol. 16, no. 1, pp. 69–73, 2012.

[14] W. A. Jansen and Cloud hooks, “Security and privacy issues in cloud computing,” in *44th Hawaii Int. Conf. on System Sciences*, Kauai, HI, USA, pp. 1–10, 2011.

[15] M. H. Song, “Analysis of risks for virtualization technology,” *Applied Mechanics and Materials*, vol. 539, pp. 374–377, 2014.

[16] S. Subashini and V. Kavitha, “A survey on security issues in service delivery models of cloud computing,” *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 1–11, 2011.

[17] R. Srivastava and A. Daniel, “Efficient model of cloud trustworthiness for selecting services using fuzzy logic,” *Emerging Technologies in Data Mining and Information Security*, vol. 755, pp. 249–260, 2019.

[18] C. Tang and J. Liu, “Selecting a trusted cloud service provider for your SAAS program,” *Computers and Security*, vol. 50, pp. 60–73, 2015.

[19] P. M. Khilar, V. Chaudhari and R. R. Swain, “Trust-based access control in cloud computing using machine learning,” *Cloud Computing for Geospatial Big Data Analytics*, Studies in Big Data 49, pp. 55–79, 2019.

[20] S. Hosseinzadeh, S. Virtanen, N. D. Rodríguez and J. Lilius, “A semantic security framework and context-aware role-based access control ontology for smart spaces,” in *The Proc. of Int. Workshop on Semantic Big Data*, San Francisco, CA, USA, pp. 1–6, 2016.

[21] P. Chinnasamy and P. Deepalakshmi, “HCAC-EHR: Hybrid cryptographic access control for secure EHR retrieval in healthcare cloud,” *Journal of Ambient Intelligence and Humanized Computing*, online, pp. 1–19, 2021.

[22] A. R. Arunachalam and G. Michael, “A trusted-role based access control model for secure cloud storage,” *International Journal of Pure and Applied Mathematics*, vol. 119, no. 12, 12243–12248, 2018.

[23] R. Charanya and M. Aramudhan, “Survey on access control issues in cloud computing,” in *Int. Conf. on Emerging Trends in Engineering, Technology and Science, (ICETETS)*, Pudukkottai, India, pp. 1–4, 2016.

[24] M. Ali, S. U. Khan and A. V. Vasilakos, “Security in cloud computing: Opportunities and challenges,” *Information Sciences*, vol. 305, pp. 357–383, 2015.

[25] P. K. Behera and P. M. Khilar, “A novel trust based access control model for cloud environment,” in *Proc. of the Int. Conf. on Signal, Networks, Computing, and Systems*, Springer, New Delhi, pp. 285–295, 2017.