# Soft Actor Critic (SAC) based Automatic Policy Generation and Effective Framework for Dynamic Trust Management for Securing SDN

**Sahana D S** ( ssanthos@gitam.edu )
GITAM University
**Brahmananda S H**
GITAM University

**Research Article**

**Keywords:** Software Defined Network(SDN), Soft Actor Critic(SAC), Automatic policy Generation, Trust Management

**Posted Date:** January 10th, 2023

**DOI:** https://doi.org/10.21203/rs.3.rs-2385922/v1



**Additional Declarations:** No competing interests reported.

# Soft Actor Critic (SAC) based Automatic Policy Generation and Effective Framework for Dynamic Trust Management for Securing SDN

**Sahana D S[1] and Dr. Brahmananda S H[2]**

*[1]Assistant Professor, Department of CSE, GITAM School of Technology, GITAM University, Bengaluru, Karnataka, India*
*[2] Professor, Department of CSE, GITAM School of Technology, GITAM University, Bengaluru, Karnataka, India*

*Email: ssanthos@gitam.edu, bsavadat@gitam.edu

**Abstract**

The Software Define Network (SDN) integrated with Internet of Things (IoT) reduces the scalability of IoT devices by managing the network, however the SDN are easily vulnerable to attacks as they used centralized controller for managing the network which can be easily manipulate by the attackers. The existing approaches focused on secure access control to the SDN controller but limits with controller scalability and trust management. By leveraging the problems in existing works, we propose SDMAC-Secure DynaMic Access Control framework which improves the security and provide efficient services to entities. Initially, all the users and applications are registered with attributes based on the registration, the authentication is performed to ensure the legitimacy. The policies are generated for the legitimate users by using Soft Actor Critic (SAC) which considers attributes, actions permitted, and temporal features to enhance network security, the conflicts between the policies are reduced by validating and storing the policies to database by the administrator. The proposed work is validated using iFog Sim tool and the performance comparisons between proposed and existing works are validated with several metrics. The simulation result shows that the proposed model work outperforms better than existing works.

**Keywords:** Software Defined Network(SDN), Soft Actor Critic(SAC), Automatic policy Generation, Trust Management

## I. INTRODUCTION

A new paradigm called Software Defined Networking (SDN) makes networks programmable and separates the control plane from the data plane, changing how networks are managed. Flexibility, automation, orchestration, and cost savings in both capital and operational expenditures are all benefits of the separation [1]. Due to SDN's centralised network management of the control plane, security of the network control plane has received special attention. Applications handle the majority of the crucial SDN functional operations; therefore, it's important to understand how each application works and only permit those functions that are absolutely necessary. For instance, an SDN application cannot enforce the flow rules to the data plane if it is only permitted to read network statistical information. In this situation, the majority of ideas for SDN control security propose a permission model to limit SDN

application operations that are not permitted. Indeed, these authorization models can increase the security of SDN controllers [2].

Effective access control must be put into place in order to protect SDN nodes when they are gaining access to resources. Access control mechanisms can prevent unauthorised access to and usage of network resources. The network node may move continually and access data object which may change in real time during the SDN application process. The node may access and exit constantly at the same moment, exhibiting a strong dynamic. Therefore, an SDN-oriented access control mechanism must address dynamic issues including timely access node updating and active access rights adjustment [3]. They do, however, carry inherent risks that need to be addressed. They frequently believe that an SDN application developer will sincerely compile a list of required permissions and that a network administrator will only approve those permissions. If an SDN application requests more or fewer permissions than necessary in this situation, a problem will result. Additionally, even when descriptions of all the listed rights for an SDN application are accessible, it can still be challenging for a network administrator to understand them all. Being a young technology, SDN is constantly gaining new features and capabilities. Therefore, it is difficult for a network administrator to take into account all of the functions used in an SDN application. This suggests that a network administrator may provide some access without sufficient expertise [4].

The security of the SDN-IoT network is improved by performing processes such as authentication of users and applications whereas this only restricts the presence of malicious users and applications, however, the threats caused by the compromised users are not mitigated [5] [6]. Several existing approaches utilized the provisioning of limited permissions for the users and applications to secure the SDN-IoT network [7] [8] [9]. The permissions were based on the policies generated by the network administrator based on the certain characteristics of the users and applications [10]. Moreover, the manual generation of policies performed by the network operator is pruned to human errors which results in adverse provisioning of permissions to the users. Edge-based SDN-IoT network is implemented to avoid high time consumption during access control to increase the throughput [11] [12]. However centralized approach reduces the security level of the network [13].

Automatic generation of permissions is an interesting area to provide flexible policies based on the user and application attributes. The access control based on these policies enables the structured usage of SDN-IoT network, however, the legitimate users also perform several unauthorized activities which also affect the security of the network. In order to mitigate the vulnerabilities caused by the authorized users, various approaches performed computation of trust for both the users and applications [14]. The trust computation was based on the behaviour of the user and applications in performing their operation.

### *A. Motivation & Objectives*

The major aim of this study is to achieve overall security and QoS of the SDN-IoT network by performing authentication and access control of both users and applications. The flexibility in provisioning of the access is provided through automatic generation of policies. The trust management of users and applications is executed based on which the updation of individual policies is performed. We are motivated by several problems addressed from the existing works which are described as follows,

- **Lack of dynamic policy update:** The policies are generated to provide access control of the users and applications but the lack of updation of these policies based on the behaviour results in providing access to an inappropriate authorized user.
- **Controller scalability:** The existing approaches performed access control in the controller but the increased number of incoming users and applications in the network results in overloading of controller which causes service delay.
- **Trust management:** The computation of trust values for the entities in the network was performed by the existing approaches but the lack of consideration of feedbacks and other behaviour-related attributes affects the effective management of trust in the network.

### *B. Research Contribution*

The secure and trusted SDMAC framework is proposed to addresses the problems faced by the existing works in terms of trust management, scalability in controller, and policy updation. The proposed contributions as follows,

- The SDN controller generates policies only for the authenticated users to ensure security based on their attributes, temporal features, and actions permitted by using Soft Actor Critic (SAC) algorithm.
- The generated policies of users and applications are computed for trust management in SDN controller which computes direct and indirect trust for the user and applications by using Non-cooperative game model.

### *C.* PROBLEM STATEMENT

The security of the SDN IoT network is a major threat which was focused by the existing approaches however these approaches faced several issues such as conflicts of policies caused by improper enforcement of policies. Additionally, there are several other issues encountered by these approaches which are mentioned as follows, Authors [15] proposed trust management system for both SDN controller and applications. The major problem of this research are listed as follows,

- The authentication of applications was performed by means of generating the tokens but the lack of authentication of users result in increased number of malicious requests which thereby increases the complexity of access control.
- The authorization of applications was carried out by considering the application attributes such as buffer, port etc. but the sensitivity of the applications are not utilized which provides static access control to all types of applications.
- The computation of trust of each application was executed and the trust state was categorized into three classes namely trusted, untrusted and uncertain, however determination of trust values only based on application state resulted in inefficient updation of trust.

## II. Literature Survey

### Schemes for access control

Authors in [16] introduced a role based reputed access control with the help of the certification process and resource provider. The certification process was initiated with the cloud service provider, by accessing information like a device identifier, physical address, and request time. Then tuples like

ID, UCN, TS, DS, V, and CPN. Then depend upon the threshold the resource provided was changed. By using hash generation and signature based authentication, resource providing operation was performed sufficiently. It was mainly used to avoid congestion. Then several steps like an authenticated message by using hyperelliptic curved cryptography were used between a service provider and user to achieve authentication. Here, the static threshold was used for all types of users this will lead to security threats and congestion while fetching resources thereby causing overhead.

Authors in [17] proposed an access control scheme to cloud based sensor IoT devices for achieving security and reliability. The time-stamp parameter was utilized as the main parameter to determine the authenticity of the device. The major aim of this paper was to avoid the impersonation and man-in-the-middle attack. This work used timestamp as the major parameter in which the attacker tries to tamper the time stamp information from the user and perform impersonation and man in the middle attack, this was overcome by using the ECC algorithm that hashes the data and provides reliable access control to the users. Here, the ECC algorithm is utilized to hash the data which was efficient however the ECC was not suitable for large IoT data.

Authors in [18] introduced an access control scheme for multiple authorities to IoT-cloud systems. This work comprises six entities namely Certificate Authority, Attribute Authorities, Attribute Authority Management module, Cloud Server, Data Owner, and Data user. Initially, security was ensured for IoT users and owners by certificate authority based on attributes in which private keys are provided. These keys are stored in the supervision module of the attribute authority. The user or owner needs an appropriate private key for accessing the data. The data in the supervision module are encrypted using the RSA algorithm. Here, RSA algorithm for the encryption process. However, the performance of the RSA algorithm is greatly affected by the size of the data resulting in increased encryption and decryption time, and credentials present in the AAM module were not secured and this attracts the attackers to compromise the AAM module to get access to the user attributes.

Authors in [19] introduced a secure edge storage system approach for IoT. This paper used Local Reconstruction Code (TLRC) and Trust Oriented Data Access Control (TODA) Schemes for securing the IoT edge storage systems. If one data server failed to transfer data to the user means another server was taking responsible and recovering the full data and transferring it to the user. TLRC makes the whole message into segments. It was useful because instead of recovering the full data the particular segment recovery was enough. They used Reputation Center (RA) to perform the trust management when the data owner was offline. Then by using TODA each segment was encrypted and generated individual key to decrypt it. Here, RA is used for trust management when the data owner was offline however there is a possibility to bribe the RA for false registering and illegal access to the protected data.

Authors in [20] introduced the anonymous decentralized attribute based access control for cloud assisted IoT. The limitations of existing approaches such as partial security to the user attributes were considered in this approach. To overcome this problem anonymous access control was proposed. In this technique by using zero-knowledge proof, they used four entities. They were Setup, Encryption, KeyGen, and Decryption. Setup was used to create the public and private keys. In the encryption method, the public key was uploaded to the public platform and the data was under encryption and hash generation. KeyGen was used to create the key for accessing the data under hash generation. In Decryption, they used a technique named blind decryption i.e., partial decryption to reduce the time.

Here partial decryption is used to increase the performance. But due to the execution of partial or blind decryption, there is more possibility for failure of decryption.

Authors in [21] introduced a attribute based provisioning of access control was proposed in this paper. The limitations of the existing approaches in providing integrity and confidentiality of the users were considered and an effective access control approach based on extensive security attributes was performed. The entity attributes such as weight, range, ID, and characteristics were utilized to design the access control of the entities. The operations such as read, write, read and write and execution was given access in this approach. Based on the access control, the path planning of the flow was performed by utilizing particle swarm optimization to achieve overall security. The determination of path for access flow was carried out by the PSO algorithm but the limitations of the algorithm such as slow convergence rate and local optima affect the efficiency of this approach.

Authors in [22] introduced provisioning of services based on the service contexts were proposed in this paper. The limitations of conventional service provisioning approaches based on contexts were studied and a reconfigurable approach was proposed. The agent based provisioning of services was utilized to achieve system adaptability in the ever-changing IoT environment. The system model comprised of entities such as data collector, feature handler, service inference, and activity inference which were integrated to achieve effective provisioning of services. The extraction of features was improved by utilizing machine learning based approaches. The provisioning of services was carried out by utilizing contextual information but the lack of consideration of QoS and SLA constraints affects the effective provisioning of services.

Authors in [23] introduced the trustworthy communication between IoT based smart buildings. This work consists of the three methods such as direct trust calculation, indirect trust calculation and total trust computation. The direct trust was computed by utilizing naïve Bayes approach in which the trust of a server was classified into ten classes between 0 and 1. Then the cosine similarity based determination of indirect trust was performed and based on both the direct and indirect trust, the total trust of a server was calculated. Finally, best optimal solution will be found out for the trustworthy communication among smart buildings. Here, context based trust computation utilized in the smart building application cannot be adopted for other real time application due to the variation of application contexts and parameters.

Authors in [24] introduced the peer to peer computation of trust of the nodes in the IoT environment was proposed in this paper. The limitations of existing trust computation approaches in determining the trust value of a strange node was addressed in this approach. The nodes in the environment were categorized into three classes namely reference holders, requesters, and providers. The reference holders are the nodes present in a device which are responsible for possessing the trust values of a device. The TruSD approach comprises of three protocols namely reference query, service provisioning and feedback report through which the computation of trust values was performed. Table II represents the research gaps in state of the works.

Authors in [25] proposed policy enforcement system to protect and manage the network using SDN controller. The main problems of this research are the access control of users and applications was carried out by the centralized SDN controller placed in the control plane but the increase in the number of incoming requests resulted in reduced throughput and increased latency. The restriction of incoming requests was performed in order to avoid overloading of controller but the lack of consideration of

application sensitivity resulted in the ignorance of highly sensitive applications. The policies generated for access control of the users and applications was performed in a manual way which is prune to human errors thereby lead to increased number of policy conflicts.

Authors in [26] proposed automatic permission framework to secure SDN controller by using VOGUE method. Effective application permission framework is proposed for providing the access control for SDN [27]. The major problems of this research are listed as follows, the automatic generation of policies was carried out outside the SDN controller but the computation of large number of permission requests resulted in increased complexity and time consumption. The management of policies based on the application behavior and user feedback was not carried out in an efficient manner which affected the overall security of the SDN network.The generation of policies and provisioning of permissions was performed based on attributes such as resource type and actions but the lack of consideration of time factor limits the security provided by the approach. The provisioning of policy was executed based on the risk analysis but the management of policies based on the application behaviour and feedback was not performed which provided only the static security.

Secure application system is proposed for providing the access control for SDN assisted cloud environment [28]. The major issues of this research are listed as follows, The NTRU algorithm was utilized for the purpose of encryption and decryption purposes but the proposed algorithm has major disadvantage of possibility to failure of decryption which affects the efficiency of the SEAPP approach. The authentication and authorization of application was carried out but the management of authority of the application was not performed by this approach which affects the overall security of the network. The application level security was provided by the SEAPP approach but the increased number of incoming application requests limited the effective computation of permissions resulting in overloading of SDN controller. Table 1 summarizes some of the other existing schemes of Access control for securing SDN-IoT Network.

**Table 1: Schemes for Access Control**

| References | Objectives | Methods or Algorithms Used | Limitations |
|---|---|---|---|
| [29] | Access control method for mitigating malicious impacts in IoT environment | Reputed access control method based on role | • High congestion rate which leads to overhead |
| [30] | Access control scheme for IoT-sensor-cloud systems | Improved Access Control and Lightweight Scheme for IoT devices | • Consume more time for encryption and decryption process |
| [31] | Secure access for IoT devices | Secure and efficient multi authority access control for IoT | • User attributes gets easily attacked |

| | | | |
|---|---|---|---|
| | | | • Consume more time for encryption and decryption process |
| [32] | Edge based access control methods | Total local reconstruction code and Trust oriented data access | • Easily bribed as the reputation center was not secured |
| [33] | Anonyms Access control scheme for IoT-cloud | Attribute based encryption method | • Decryption failure by using partial decryption method |
| [34] | Access control methods based on dynamicity for software defined networks | Particle Swarm Optimization | • Low converge by utilizing particle swarm optimization for access control |
| [35] | Service provisioning for IoT smart cities | Context Aware Service Provisioning | • Not considering QoS and SLA requirements leads to ineffective provisioning |

### *A. Research Solutions:*

The above issues are addressed by the following solutions.

(1) The authentication of both users and applications is performed in order to achieve high security in the SDN-IoT network, moreover the complexity of access control is reduced by performing it in an edge based decentralized manner.

(2) The application and user access control is provided based on the user and application attributes and expiry time which improves the security of the SDN controller.

(3) The updation of individual policies was carried out based on the trust computation of each application in which the dynamic trust level of the application is mapped to the predefined levels of trust.

## III PROPOSED MODEL

The proposed study focuses on ensuring the security of the SDN controller by mitigating the security threats caused by unauthorized users and applications connected through the northbound and southbound interfaces of the controller. The proposed SDMAC which is shown in Figure 1, includes $n$ number of IoT users($\mathrm{U}_i$), applications ($\tilde{\mathrm{A}}_i$) edge gateways($E_i$), SDN controller ($SC_i$)and single Trusted Authority ($TA$) and Cloud server. In which the IoT devices and applications are provide the request to access the network. Then SDN performs policy enforcement and trust management for both applications and users by considering user attributes ($U_a$)and application attributes ($A_a$) respectively. After completed policy enforcement and trust management the Edge gateway provide the access to the users and applications by verifying the policies. Finally, services are provided to the users by the controller. Table 2 summarizes

the goals of the SDMAC model. Table 3 summarizes the notations used for designing Automatic policy enforcement and Dynamic Trust Management algorithms.

The proposed system includes four planes which are explained as follows,

*(i)* ***IoT User plane-*** This plane includes multiple IoT users. The requests are collected from IoT users to access the server for getting services from the SDN controller. Here, only authenticated users can access the network.

*(ii)* ***Infrastructure plane-***It includes multiple numbers of edge gateways. It is used to collect the policies from the users and send it to the SDN controller. It provides the access control to the users and application based on their policies and trust values.

*(iii)* ***Control Plane***-It includes multiple numbers of SDN controller and virtual SDN controller. It is used to manage the network through controller. SDN controller performs policy enforcement and trust management in order to improve the security of the network. If the SDN controller is overloaded then we employed virtual SDN controller based on the load of the SDN.

*(iv)* ***Application Plane-*** It includes n number of applications and single cloud server which is used to store and retrieve the application through SDN controller. Table II describes the goals of SDMAC model. Fig 2 illustrates the architecture of SDMAC system model.

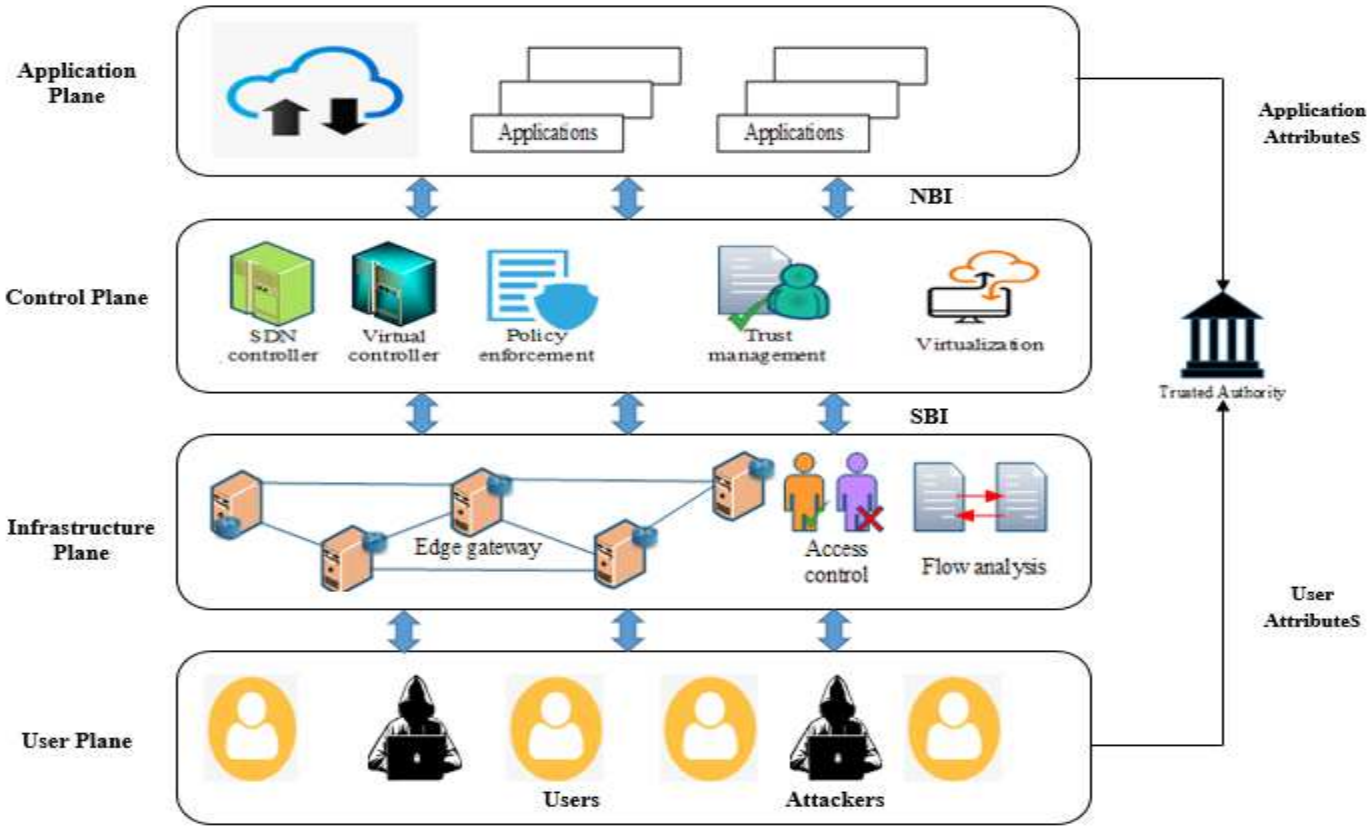


**Figure 1: Proposed SDMAC Architecture**

**TABLE 2: GOALS OF SDMAC MODEL**

| Process | Algorithms | Goals |
|---|---|---|
| Policy Enforcement | Soft Actor Critic | • Increased flexibility<br>• Avoid conflicts in policies |
| Dynamic Trust Management | Non-Cooperative Game Model | • Increases security<br>• Reduce false positive |

**Table 3: NOTATION Table**

| Notation | Description |
|---|---|
| $\mathfrak{U}, \tilde{\mathfrak{A}}$ | Users and applications |
| $U_a, A_a$ | User attributes and application attributes |
| $\lambda, \Psi$ | Private and public matrix |
| $SC_i$ | SDN controller |
| $TA$ | Single trusted authority |
| $\mathfrak{G}, \mathfrak{F}, \mathfrak{K}$ | State, action, reward |
| $Q_\vartheta(\mathfrak{G}, \mathfrak{F}),\ V_\tau(\mathfrak{F})$ | Q-function and state value |
| $\mathfrak{Q}, \mathfrak{Z}, u$ | Participants, strategies, and utility function |
| $\mathfrak{Z}^*_{\mathfrak{U}_j}, \mathfrak{Z}^*_{\tilde{\mathfrak{A}}_j}$ | Strategies of equilibrium |
| $\mathfrak{a}$ | User and application alternatives |
| $D^c$ | Decision making norms |
| $\exists^{irh}$ | Set of policy ratings |
| $\overline{DF}\left(\overline{UA}_{ih}{}^{D}\right)$ | Defuzzified normalized matrix |
| $L_{U_i}, L_{A_j}$ | Request provided by the users and applications |
| $\Pr(L_{U1_i}), \Pr(L_{A1_i}),$ | Probabilities of user and applications |

### *3.1 Automatic Policy Enforcement*

The generation of policies for the authorized users and applications is a significant process in protecting the security of the network. Once the users and applications are registered, the policies are generated in

an automatic manner to provide access. The generation of policies is based on both the user attributes and application attributes. The generated policy considers three elements such as attributes, actions permitted, and temporal features (date and time). The policies generated are validated by the administrator and are stored in a database for further generation of policies; this provides flexibility and avoids conflicts in policies. The automatic generation of policies is performed by implementing Soft Actor Critic (SAC) algorithm which executes the optimal action based on the current state.

Initially, Markov decision method is created by considering action ($\mathfrak{F}$), state ($\mathfrak{G}$) and reward ($\mathfrak{K}$). The SAC policy is denoted as $\mathbb{P}_{\emptyset}(\mathfrak{G}|\mathfrak{F})$. The Q function of SAC is represented as $Q_{\vartheta}(\mathfrak{G},\mathfrak{F})$, the value of state is represented as $V_{\tau}(\mathfrak{F})$. The SAC network parameters are $\theta, \vartheta, \tau$. Here, we have considered two states such as $U_a$ and $A_a$. Based on the current state SAC takes the actions such as policy generation and verification. The SAC provides the reward such as permission and prohibition. The residual errors are minimized by using soft value function that is represented as follows,

$$F_V(\tau) = \mathbb{E}_{\mathfrak{G}\sim d}\left[\frac{1}{2}V_{\tau}(\mathfrak{G}) - E_{\mathfrak{F}\sim\mathbb{P}_{\emptyset}}[Q_{\vartheta}(\mathfrak{G},\mathfrak{F}) - \log\mathbb{P}_{\emptyset}(\mathfrak{F}|\mathfrak{G})]^2\right] \quad (1)$$

Where, d represents the distribution of existing state and the gradient function evaluation is defined as follows,

$$\nabla_{\tau}F_V(\tau) = \nabla_{\tau}V_{\tau}(\mathfrak{G}) - (V_{\tau}(\mathfrak{G}) - Q_{\vartheta}(\mathfrak{G},\mathfrak{F}) + \log\mathbb{P}_{\emptyset}(\mathfrak{F}|\mathfrak{G})) \quad (2)$$

SAC takes the action based on the current policy, and then it updates the soft Q value to optimize the stochastic gradient function which is represented as follows,

$$\nabla_{\vartheta}F_Q(\vartheta) = \nabla_{\vartheta}Q_{\vartheta}(\mathfrak{F},\mathfrak{G})(Q_{\vartheta}(\mathfrak{G},\mathfrak{F}) - \mathfrak{K}(\mathfrak{G},\mathfrak{F}) - \beta V_{\tau}(+1)) \quad (3)$$

The policy parameter learning is used to obtain optimal policy that is defined as follows,

$$F_{\mathbb{P}}(\emptyset) = \mathbb{E}_{\mathfrak{G}\sim d, b_t\sim m}[\log\mathbb{P}_{\emptyset}(l_{\emptyset}(b_t;\mathbb{S}t)|\mathbb{S}t) - Q_{\theta}(\mathbb{S}t,\mathbb{P}_{\emptyset}(b_t;\mathfrak{G}))] \quad (4)$$

Where $l_{\emptyset}(b_t;\mathfrak{G})$ represent the action of the current network, in this way we can perform policy enforcement. Figure 2 illustrates the Automatic policy enforcement using SAC algorithm in which optimal reward is generated based on the actions. The pseudocode is provided for the automatic policy enforcement by SAC.

**Algorithm**

Input: State attributes $(U_a,\ A_a)$
Output: Policy enforcement for $\mathfrak{U}_j, \tilde{\mathfrak{A}}_j$
Begin
Initialize network parameters $(\theta, \phi, \tau)$
For every episode do
  Set initial state $I_0 = 0$
  For every time step do
    Perform action $\mathfrak{F}$ from policy $\mathfrak{P}_{\emptyset}(\mathfrak{F}|\mathfrak{G})$
    Change to next state $\mathfrak{G}_{t+1}$
    Create reward $\mathfrak{K}$ based on $\mathfrak{F}$
    Collect $(\mathfrak{G}_t, \mathfrak{F}, \mathfrak{K}, \mathfrak{G}_{t+1})$ in the replay buffer
  End for
  For every gradient step do
    $\tau \leftarrow \tau - \alpha_v \hat{\nabla}_{\tau} h_v(\tau)$
    $\vartheta_n \leftarrow \vartheta_n - \alpha_Q \hat{\nabla}_{\vartheta_n} h_Q(\vartheta_n)$
    $\phi \leftarrow \phi - \alpha_{\pi} \hat{\nabla}_{\phi} h_{\pi}(\phi)$
    $\delta \leftarrow \varepsilon\delta + (1-\varepsilon)\ \delta$
  End for
End for
End

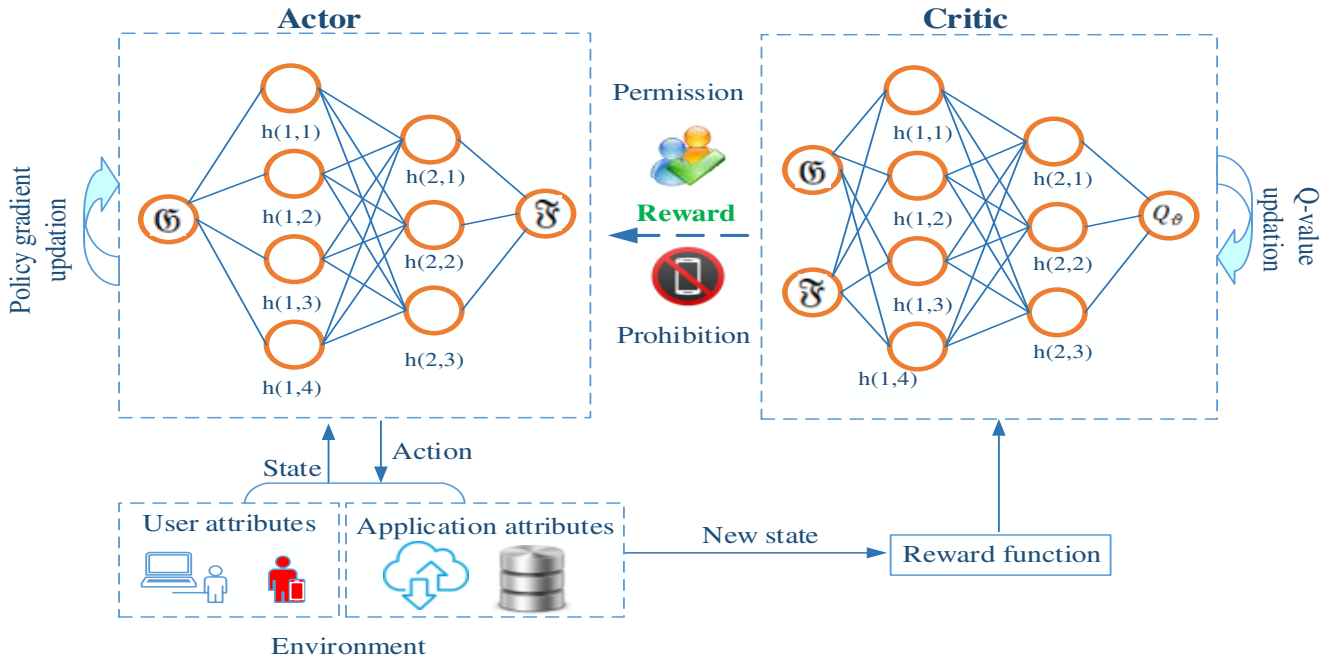


**Figure 2: SAC based automatic policy generation**

### *3.2 Dynamic Trust Management*

Both the policy enforcement and trust management are carried out by the SDN controller based on which the access control for users is performed. The policies generated in the SDN controller are constructed into a policy graph by utilizing in which the first level of nodes form the general policies and second level of nodes form the individual polices respectively. The general policies provide the permission based on the role and type of the user and applications whereas the individual policies provide the permissions based on the individual trust of the user and applications. The trust computation is based on the behavior of user and applications and the feedback provided by the other users. The dynamic computation of trust is carried out by utilizing the Non-cooperative game model in which the users and applications achieve trust based on their activities. The non-cooperative game model includes three entities such as set of participants($\mathfrak{Q}$), game strategies ($\mathfrak{Z}$) that denotes the set of decisions which makes by the participants, and set of utility functions which is used to assess the participants profit and correct the game strategies of the participants. The game model is defined as follows,

$$G = \{\mathfrak{Q}, \mathfrak{Z}, u\} = \{\mathfrak{U}_j, , \tilde{\mathfrak{A}}_j;\ \mathfrak{Z}_{\mathfrak{U}_j}, \mathfrak{Z}_{\tilde{\mathfrak{A}}_j};\ C_{\mathfrak{U}_j}, C_{\tilde{\mathfrak{A}}_j}\} \qquad (4)$$

The proposed Nash Equilibrium is one of the non-cooperative game models, in which both application and users are participate in the game for evaluating the trust values. The strategies of the equilibrium is defined as $\mathfrak{Z}^*_{\mathfrak{U}_j}, \mathfrak{Z}^*_{\tilde{\mathfrak{A}}_j}$, in which the outcome of the inputs are reach maximum value then the participants must follows the conditions,

$$u_i(\mathfrak{Z}^*_1, \dots, \mathfrak{Z}^*_{i-1}, \mathfrak{Z}^*_i, \mathfrak{Z}^*_{i+1}, \dots, \mathfrak{Z}^*_n) \qquad (5)$$

$$\geq u_i(\mathfrak{Z}^*_1, \dots, \mathfrak{Z}^*_{i-1}, \mathfrak{Z}^*_i, \mathfrak{Z}^*_{i+1}, \dots, \mathfrak{Z}^*_n) \qquad (6)$$

Before solving the Nash equilibrium, we need to solve the theorems which are defined as follows,

***Theorem I:*** For any strategy game $G = \{\mathfrak{Q}, \mathfrak{Z}, u\}$, there should be a solution for equilibrium.
***Theorem II:*** For any strategy restricted game $G = \{\mathfrak{Q}, \mathfrak{Z}, u\}$ where $s_i \in r^n$, $s_i \neq \emptyset$, satisfies the condition that $u_i(s_i)$ is constant non-convex then it exist exclusive equilibrium.

Analysing the trust management characteristics, this can be generated continuously and dynamically. Hence, the finite game meets the theorem I requirements which should be a solution of equilibrium. The Theorem I classified into two classes, one is the unique equilibrium solution and second one is the multiple equilibrium solutions. To validate the existing equilibrium solution, we need to concatenate Theorem II for explaining the constant non-convex utility property. Both user and application trust are evaluated by considering the role and feedbacks. Here, neighbouring nodes are plays the strategies for providing the feedbacks of the user and application. Based on the strategies of the neighbour nodes in the network, the SDN makes the decisions. In this way, SDN classifies the trust into two classes namely trusted, non-trusted. For instance, if the type of the application is sensitive and the incoming requests of this application is higher than the frequency threshold then the trust of the particular element will be reduced and the policy of the individual is altered.

## IV EXPERIMENTAL RESULTS

The performance of the proposed SDMAC approach is validated in this section. The experimental result proves that the proposed SDMAC method attains high security. This section consists of three sub-sections such as simulation setup, application scenario, comparative analysis, and research summary.

### A. Simulation Setup

The proposed SDMAC method is performed by integrating the IoT environment with SDN and edge to improve security. The proposed work is simulated by the simulation tool namely IFogSim simulator with several entities such as IoT devices, Edge-based gateways, SDN controllers, trusted authority, and cloud. The simulation is executed by an Intel® Core i5-4590s processor at 3.00 GHz with 8 GB of RAM and 500GB of ROM. The operating system (OS) used to perform simulation is Windows 10 pro 64 bits. Table 4, illustrates the system parameters of the proposed SDMAC method.

**TABLE 4: SIMULATION PARAMETERS**

| Parameter | Values |
|---|---|
| Number of IoT devices | 50 |
| Number of Edge gateways | 4 |
| Number of SDN controllers | 4 |
| Number of Trusted authority | 1 |
| Number of cloud server | 1 |
| Communication range | 110 m |
| Delay | $3\mu s$ |
| Number of packets | 1500 |
| Simulation time | 250 ms |
| Response Time | Max 1.3 s |
| Energy consumption | Max 10j |
| Number of Open Flow switches | 4 |
| Controller Processing latency | $0.4\mu s$ |
| Request size | ~1Mb |

### *B. Application Scenario (Smart Office)*

In recent days, the applications of SDN-IoT are rapidly expanded to improve security by providing access control, especially in offices or companies. Figure 3 illustrates the application scenario of smart office in SDN-IoT environment. Various processes are performed in the smart office application scenario such as data transfer, cloud printing, data management, and connectivity, etc. However, numerous issues are addressed in the applications regarding security-based on misbehavior of legitimate users. So, we proposed a SDMAC approach and applied it in this smart office scenario to overcome these issues. Authentication of users is performed based on attributes by the trusted authority to increase the legitimacy of the users before entering the office. Then the attributes are stored in a hashed manner at the SDN controller to improve privacy of the legitimate user. Policy generation is performed for the

legitimate (i.e. authorized) users and applications based on several elements to increase the network protection.

Trust management is performed dynamically by the SDN controller by constructing a policy graph with two levels to identify the role, type, and individual trust of the applications and users. Policy generation and trust management are performed to improve secure data transfer and data management. Access control is provided for the users and applications through edge gateway only by verifying the global and local policies for providing access only to the authorized users to enter into the data management area. After providing access, a matching response is provided for the respective request effectively with low time consumption to increase the QoS. This proposed SDMAC method provides safe cloud printing, efficient access control, high secure connectivity between the users, robust data transfer, and data management in the smart office scenario with better QoS.

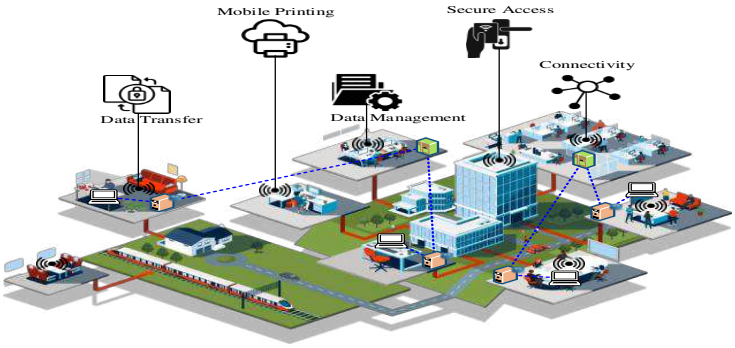


**Figure 3: Use-Case Scenario**

## V COMPARATIVE ANALYSIS

This section summarizes impact of different parameters have been considered to analyse the efficiency of proposed methodology by comparing it with different methods. Latency is defined as the amount of delay time taken between the users' request and the applications' response. In general, latency is caused due to policy generation and trust management. Low latency provides efficient network. The comparison of latency is based on number of tasks and number of edge gateways.

### *5.1 Impact of Latency*

Figure 4, shows the comparison of latency for the proposed SDMAC method and several existing methods in terms of number of tasks. Increasing the number of tasks increases the latency. Centralized SDN controller is used in LPE-SDN method for performing access control however, it increases the latency when more number of tasks is arrived that results in high latency.

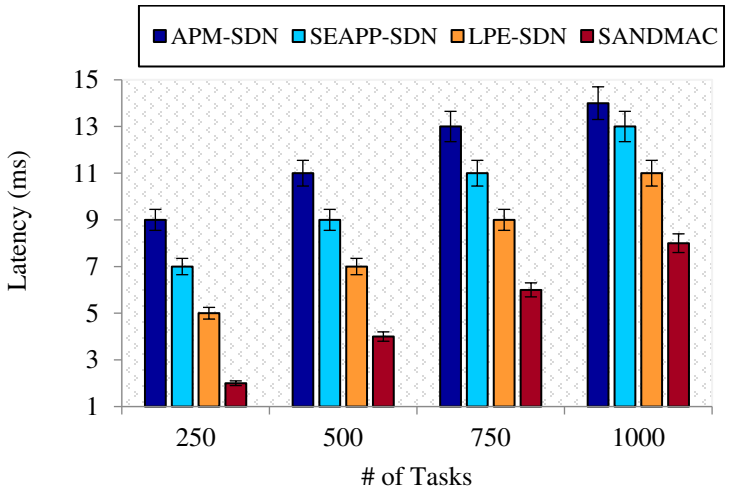


**Figure 4: Latency vs. # of Tasks**

In the proposed SDMAC method, decentralized SDN controller is used to perform access control which reduces the latency. In addition, virtual controller is also implemented to avoid overloading thereby reducing the latency in an efficient manner. The graphical result shows that the proposed work achieves low latency with an average of 8ms which is 3ms faster than LPE-SDN method, 5ms faster than SEAPP-SDN method and 6ms faster than APM-SDN method. Similarly, figure 5 represents the comparison of latency with respect to the number of edge gateways. This figure shows that the proposed SDMAC method attains an average latency of 18ms which is 5ms faster than LPE-SDN method, 9ms faster than SEAPP-SDN method, and 10ms faster than APM-SDN method. Table 5, illustrates the analysis of latency with respect to the number of tasks and edge gateways between proposed and existing methods.

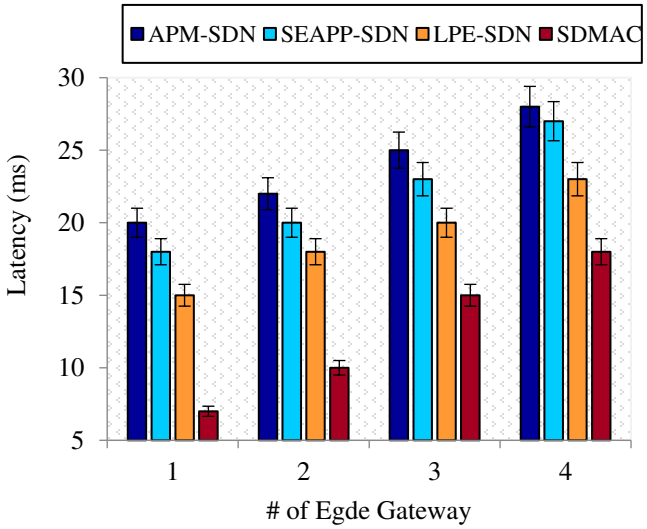


**Figure 5: Latency vs. # of Edge Gateway**

**TABLE 5 : ANALYSIS OF LATENCY**

| Methods | Latency (ms) |
|---|---|

| | # of Tasks | # of Edge Gateways |
|---|---|---|
| APM-SDN | 11.75±0.4 | 23.75±0.5 |
| SEAPP-SDN | 10.2±0.3 | 22.5±0.4 |
| LPE-SDN | 8.3±0.2 | 19.3±0.3 |
| SDMAC | 5.1±0.1 | 12.5±0.1 |

### 5.2 Impact of Trust Values

This metric is used to measure the trust calculation for the compromised users (i.e. malicious users) from the total calculation of trust. Efficient trust calculation of compromised users increases the integrity of the network. Figure 6, illustrates the comparison of trust values of compromised between the proposed SDMAC method and other existing works with respect to time. Trust value is decreased by increase of time. The trust calculation of the proposed SDMAC method is performed efficiently by generating the policy based on attributes, temporal features and actions permitted using SAC algorithm and trust computation is performed based on feedback of the user or applications from the neighbours. In trust management, the general and individual policies are verified to provide trust value for the users and applications. This increases the trust value of the proposed SDMAC method when compared with previous works. The proposed SDMAC method achieves trust value of 5 at 250ms. The trust value difference between the proposed SDMAC approach and LPE-SDN method is 1, for SEAPP-SDN method is 2 and for APM-SDN method is 3 with 250ms of time. Table 6 explains the numerical analysis of trust value.

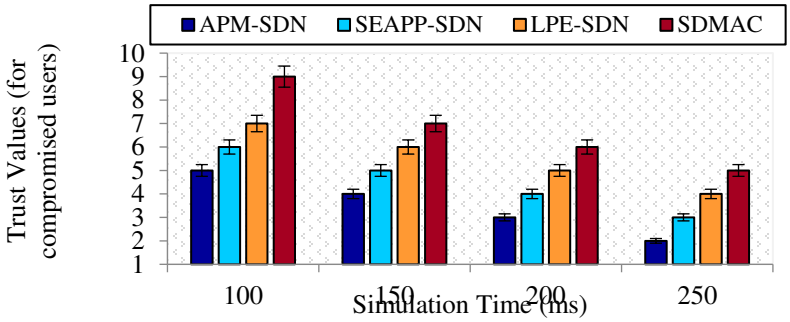


**Figure 6: Trust Values vs. Time**

**TABLE 6: ANALYSIS OF TRUST VALUE**

| Methods | Trust Value |
|---|---|
| APM-SDN | 3.5±0.5 |
| SEAPP-SDN | 4.5±0.4 |
| LPE-SDN | 5.5±0.3 |
| SDMAC | 6.75±0.1 |

**TABLE 7**

**COMPARISON OF PROPOSED VS EXISTING WORKS**

The SDN controller is the core component, providing network services for upper layer applications and managing the fundamental network resources for establishing ground work for proposed model. Due to the combination of SDN and IoT, they manage network resources more flexibly. Below Table 7 provides the summary of the comparison of existing approaches with the proposed SDMAC approaches by considering all the parameters of Authentication, Policy Enforcement, Trust Management and Access Control. Table 8 gives the overview of the Security analysis with the proposed models.

| **Reference** | **Concept/ Objective** | **Authentication** | **Policy Enforcement** | **Trust Management** | **Access Control** | **Problems** | **Proposed Approaches** |
|---|---|---|---|---|---|---|---|
| [36] | Trust management for SDN network | ✓ | ✗ | ✓ | ✓ | • High complexity<br>• Inefficient trust management<br>• Poor access control | • Deployment of edge gateways<br>• Dynamic trust management<br>• Multiple attributes based access control |
| [37] | Resource protection for SDN network | ✗ | ✓ | ✗ | ✓ | • High latency<br>• Low throughput<br>• Increased policy conflicts | • Edge gateways deployment<br>• Automatic policy enforcement |
| [38] | Provide access control for control plane of SDN | ✗ | ✓ | ✗ | ✓ | • Static security<br>• High overloading<br>• Poor trust management | • Generating policies automatically<br>• Deployment of virtual controller<br>• Dynamic trust management |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| [39] | Provide permission for application in SDN | ✗ | ✓ | ✗ | ✓ | • Inefficient policy generation<br>• High time consumption | • Time factor based policy generation<br>• Edge gateway based access control |
| [40] | Secure access control SDN environment | ✓ | ✗ | ✗ | ✓ | • Inefficient authentication<br>• High controller overload | • Effective authentication<br>• Deployment of virtual controller |

**TABLE 8 : SECURITY ANALYSIS**

| **Attacks** | **Solutions** | **Algorithms Used** |
|---|---|---|
| DDoS Attack | DDoS attack is mitigated by performing policy enforcement for both users and applications based on permitted action, attributes and temporal features. | Soft Actor Critic (SAC) Algorithm |
| XSS Attack | XSS attack is mitigated by performing dynamic trust management based on individual and general policies with feedback. | Non-cooperative game model |

**Conclusion**

The secure and trusted SDMAC framework is proposed for improve the performance in terms of security for the SDN-IoT network. All the users and applications in the SDN-IoT network are registered with their attributes to the TA and stored to SDN controller in hashed manner based on the registration, authentication is performed by using Bliss-B algorithm to ensure legitimacy of user and applications. The automatic policy enforcement of authenticated users is carried out by using SAC which considers three elements namely attributes, actions permitted, and temporal features. Based on the generated policies, the access control for the users and application is carried out in edge gate way by using IFDA which mitigates the several cyber-attacks.

**Declarations:**

-**Ethics Approval:** Not Applicable

-**Conflict of Interest:** No conflict of Interest

-**Data Availability:** Not Applicable

-**Author Contribution:** Both the authors are equally contributed in all the levels of design, implementation, and results

-**Funding:** Not Applicable

-**Consent to publish:** Both the authors give consent for the publication

**References**

[1] K. Thimmaraju , B. Shastry , T. Fiebig , F. Hetzelt , J.-P. Seifert , A. Feldmann , S. Schmid , Taking control of SDN-based cloud systems via the data plane, in: Proceedings of the Symposium on SDN Research, ACM, 2018, p. 1.

[2] C. Yoon , S. Shin , P. A , V. Yegneswaran , H. Kang , M.W. Fong , A security-mode for carrier-grade sdn controllers., ACSAC, 2017.

[3] Nife, F. and Kotulski, Z. (2018) New SDN-Oriented Authentication and Access Control Mechanism. International Conference on Computer Networks.

[4] C. Yoonet al., "A security-mode for carrier-grade sdn controllers,"inProceedings of the 33rd Annual Computer Security ApplicationsConference. ACM, 2017, pp. 461–473.

[5] Fang, L., Li, Y., Yun, X., Wen, Z., Ji, S., Meng, W., Cao, Z., & Tanveer, M. (2020). THP: A Novel Authentication Scheme to Prevent Multiple Attacks in SDN-Based IoT Network. *IEEE Internet of Things Journal, 7*, 5745-5759.

[6] Li, J., Jin, J., Lyu, L., Yuan, D., Yang, Y., Gao, L., & Shen, C. (2021). A Fast and Scalable Authentication Scheme in IoT for Smart Living. *Future Gener. Comput. Syst., 117*, 125-137.

[7] Menard, P., & Bott, G.J. (2020). Analyzing IOT users' mobile device privacy concerns: Extracting privacy permissions using a disclosure experiment. *Comput. Secur., 95*, 101856.'

[8] Khalid, W., & Yu, H. (2021). Security Improvement With QoS Provisioning Using Service Priority and Power Allocation for NOMA-IoT Networks. *IEEE Access, 9*, 9937-9948.

[9] Badawy, M.M., Ali, Z.H., & Ali, H.A. (2019). QoS provisioning framework for service-oriented internet of things (IoT). *Cluster Computing*, 1-17.

[10] Wadhwa, H., & Aron, R. (2021). Resource Utilization for IoT Oriented Framework Using Zero Hour Policy. *Wireless Personal Communications*.

[11] Das, R.K., Ahmed, N., Pohrmen, F.H., Maji, A.K., & Saha, G. (2020). 6LE-SDN: An Edge-Based Software-Defined Network for Internet of Things. *IEEE Internet of Things Journal, 7*, 7725-7733.

[12] Lv, Z., & Xiu, W. (2020). Interaction of Edge-Cloud Computing Based on SDN and NFV for Next Generation IoT. *IEEE Internet of Things Journal, 7*, 5706-5712.

[13] Mavromatis, A., Colman-Meixner, C., Silva, A.P., Vasilakos, X., Nejabati, R., & Simeonidou, D. (2020). A Software-Defined IoT Device Management Framework for Edge and Cloud Computing. *IEEE Internet of Things Journal, 7*, 1718-1735.
[14] Lv, Z., & Xiu, W. (2020). Interaction of Edge-Cloud Computing Based on SDN and NFV for Next Generation IoT. *IEEE Internet of Things Journal, 7*, 5706-5712.

[15] Aliyu, A.L., Aneiba, A., Patwary, M., & Bull, P. (2020). A trust management framework for Software Defined Network (SDN) controller and network applications. Comput. Networks, 181, 107421.

[16] Amoon, M., Altameem, T., & Altameem, A. (2020). RRAC: Role based reputed access control method for mitigating malicious impact in intelligent IoT platforms. Comput. Commun., 151, 238-246.

[17]Chaudhry, S.A., Yahya, K., Al-turjman, F., & Yang, M. (2020). A Secure and Reliable Device Access Control Scheme for IoT Based Sensor Cloud Systems. IEEE Access, 8, 139244-139254.

[18] Xiong, S., Ni, Q., Wang, L., & Wang, Q. (2020). SEM-ACSIT: Secure and Efficient Multiauthority Access Control for IoT Cloud Storage. IEEE Internet of Things Journal, 7, 2914-2927.

[19] Xia, J., Cheng, G., Gu, S., & Guo, D. (2020). Secure and Trust-Oriented Edge Storage for Internet of Things. IEEE Internet of Things Journal, 7, 4049-4060.

[20] Nasiraee, H., & Ashouri-Talouki, M. (2020). Anonymous decentralized attribute-based access control for cloud-assisted IoT. Future Gener. Comput. Syst., 110, 45-56.

[21] Chang, D., Sun, W., Yang, Y., & Wang, T. (2019). A Dynamic Access Control Method for SDN. Journal of Computational Chemistry, 07, 105-115.

[22] Lu, C. (2020). Context-Aware Service Provisioning via Agentized and Reconfigurable Multimodel Cooperation for Real-Life IoT-Enabled Smart Home Systems. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 50, 2914-2925.

[23] Altaf, A., Abbas, H., Iqbal, F., Khan, M.M., & Daneshmand, M. (2021). Robust, Secure, and Adaptive Trust-Oriented Service Selection in IoT-Based Smart Buildings. IEEE Internet of Things Journal, 8, 7497-7509.

[24] Kalkan, K., & Rasmussen, K.B. (2020). TruSD: Trust framework for service discovery among IoT devices. Comput. Networks, 178, 107318.

[25] Leng, X., Hou, K., Chen, Y., Bu, K., Song, L., & Li, Y. (2019). A lightweight policy enforcement system for resource protection and management in the SDN-based cloud. Comput. Networks, 161, 68-81.

[26] Kang, H., Yegneswaran, V., Ghosh, S., Porras, P.A., & Shin, S. (2020). Automated Permission Model Generation for Securing SDN Control-Plane. IEEE Transactions on Information Forensics and Security, 15, 1668-1682.

[27] Kang, H., Yoon, C., & Shin, S. (2019). Astraea: Towards an effective and usable application permission system for SDN. Comput. Networks, 155, 1-14.

[28] Hu, T., Zhang, Z., Yi, P., Liang, D., Li, Z., Ren, Q., Hu, Y., & Lan, J. (2021). SEAPP: A secure application management framework based on REST API access control in SDN-enabled cloud environment. J. Parallel Distributed Comput., 147, 108-123.

[29] Amoon, M., Altameem, T., & Altameem, A. (2020). RRAC: Role based reputed access control method for mitigating malicious impact in intelligent IoT platforms. Comput. Commun., 151, 238-246.

[30] Chaudhry, S.A., Yahya, K., Al-turjman, F., & Yang, M. (2020). A Secure and Reliable Device Access Control Scheme for IoT Based Sensor Cloud Systems. IEEE Access, 8, 139244-139254.

[31] Xiong, S., Ni, Q., Wang, L., & Wang, Q. (2020). SEM-ACSIT: Secure and Efficient Multiauthority Access Control for IoT Cloud Storage. IEEE Internet of Things Journal, 7, 2914-2927.

[32] Xia, J., Cheng, G., Gu, S., & Guo, D. (2020). Secure and Trust-Oriented Edge Storage for Internet of Things. IEEE Internet of Things Journal, 7, 4049-4060.

[33] Nasiraee, H., & Ashouri-Talouki, M. (2020). Anonymous decentralized attribute-based access control for cloud-assisted IoT. Future Gener. Comput. Syst., 110, 45-56.

[34] Chang, D., Sun, W., Yang, Y., & Wang, T. (2019). A Dynamic Access Control Method for SDN. Journal of Computational Chemistry, 07, 105-115.

[35] Lu, C. (2020). Context-Aware Service Provisioning via Agentized and Reconfigurable Multimodel Cooperation for Real-Life IoT-Enabled Smart Home Systems. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 50, 2914-2925.

[36] Aliyu, A.L., Aneiba, A., Patwary, M., & Bull, P. (2020). A trust management framework for Software Defined Network (SDN) controller and network applications. Comput. Networks, 181, 107421.

[37]Leng, X., Hou, K., Chen, Y., Bu, K., Song, L., & Li, Y. (2019). A lightweight policy enforcement system for resource protection and management in the SDN-based cloud. Comput. Networks, 161, 68-81.

[38] Kang, H., Yegneswaran, V., Ghosh, S., Porras, P.A., & Shin, S. (2020). Automated Permission Model Generation for Securing SDN Control-Plane. IEEE Transactions on Information Forensics and Security, 15, 1668-1682.

[39] Kang, H., Yoon, C., & Shin, S. (2019). Astraea: Towards an effective and usable application permission system for SDN. Comput. Networks, 155, 1-14.

[40] Hu, T., Zhang, Z., Yi, P., Liang, D., Li, Z., Ren, Q., Hu, Y., & Lan, J. (2021). SEAPP: A secure application management framework based on REST API access control in SDN-enabled cloud environment. J. Parallel Distributed Comput., 147, 108-123.