International Journal of Core Engineering & Management
Volume-7, Issue-08, 2023 ISSN No: 2348-9510

PRIVACY PRESERVED APNSC-BASED SECURE DATA SOURCE DISTRIBUTION IN ACTIVE-ACTIVE DATA CENTER USING PCS-WHA AND ECD-DSA

Amaresan Venkatesan
v.amaresan@gmail.com

Abstract

Recently, a promising part of cloud computing is active-active Data Centres (DCs), allowing services for numerous applications. However, due to poor privacy preservation, the conventional schemes were ineffective. Thus, by using the Playfair Substitution Cipher Whirlpool Hashing Algorithm (PSC-WHA) and Euclidean Ceiling Division-based Digital Signature Algorithm (ECD-DSA), this paper proposes a privacy-preserved Access Policy along with Nested Smart Contract (APNSC) based reliable data source distribution in an active-active DC. Primarily, in the blockchain, the admin, cloud users, and DC are registered, followed by key generation and privacy preservation. Moreover, for the DC, the Access Policy (AP) is generated and then stored in the Inter Planetary File System (IPFS). Then, through the IPFS, the data is distributed between the DC. Likewise, the data user assigns the workflow to access the sources from the DC. Here, regarding the user's request and APNSC between the DC and IPFS and the cloud user and IPFS, the DC is selected. Moreover, to select the optimal fog, the De Bruijn Botox Optimization Algorithm (DB2OA) is used. Lastly, the APNSC is digitally signed and sent to the DC. The dominance of the proposed approach is depicted by the experimental findings.

Keywords: Cloud Computing (CC), Inter Planetary File System (IPFS), Privacy Preservation (PP), Access Policy along with Nested Smart Contract (APNSC), Playfair Substitution Cipher Whirlpool Hashing Algorithm (PSC-WHA), Euclidean Ceiling Division based Digital Signature Algorithm (ECD-DSA) and Access Policy (AP).

I. INTRODUCTION

A distributed computing platform that provides on-demand access to a shared pool of resources via the Internet is termed the CC (Ahmad et al., 2021). Also, in global networks, cloud DC plays a vital role, thus allowing services for various applications (Katal et al., 2023). Usually, cloud architectures depend on geographically distributed DC (Chen et al., 2021). So, the active-active DC is the promising factor, where multiple DCs simultaneously serve applications (Helali & Omri, 2021). But, during cloud data distribution, data breaches are a major concern. Thus, to ensure secure storage in active-active DC, the blockchain and IPFS are established (Kang et al., 2022) (Zarrin et al., 2021).

Cryptographic techniques like Advanced Encryption Standard (AES) are utilized to secure the distributed data in the existing works (Awan et al., 2020) (Sultana et al., 2020). The IPFS is introduced to ensure data confidentiality, which generates the hash code of the encrypted data uploaded by the data owners (Athanere & Thakur, 2022). Similarly, to identify the optimal fog, traditional systems employ optimization algorithms like Particle Swarm Optimization (PSO) (Morkevičius et al., 2023). But, during data distribution, none of the traditional works focused on verifying the AP privileges. Thus, by using PCSWHA-MT and ECD-DSA, this paper proposes a geographical location AP privileges-aware privacy-preserved APNSC-based reliable data source distribution in an active-active DC.

1.1 Problem statement

- None of the traditional models focused on verifying the access privilege policies of the DC and registered user during data distribution.
- (Hogade et al., 2022) failed to handle the data redundancy, affecting model performance.
- (Ulabedin & Nazir, 2021) had high transmission delay when the data was transferred across the geographically separated locations.
- The existing methods were insignificant owing to the poor preservation schemes.

1.2 Objectives

- Here, the created digital signature of the APNSC is verified in the DC before task execution.
- The proposed work introduces the Hashcode-based Merkle tree to diminish the data redundancy.
- The DB^{2}OA is employed to perform optimal fog selection, thus reducing the time delay.
- The ED^{2}S-KAnonymity is established to preserve the user's sensitive information.

The remaining part is organized as: the associated works are demonstrated in Section 2; the proposed mechanism is illustrated in Section 3; the findings of the proposed approach are discussed in Section 4; the paper is concluded in Section 5.

II. LITERATURE SURVEY

(Hogade et al., 2022) offered energy and network-aware workload monitoring for geographically distributed DC. To manage the workload among the cloud DC effectively, game theory was introduced. Here, there was a minimum DC queuing delay. But, this approach failed to handle data redundancy.

(Ulabedin & Nazir, 2021) examined the replication as well as data management-centric workflow scheduling approach for a multi-cloud DC platform. To perform efficient workflow allocation, the Replication-centric Partial Critical Path (R-PCP) was established. But, when the data was transferred across the different locations, this framework had high time complexity.
(Sayadnavard H et al., 2022) established a multi-objective model for energy-efficient as well as secure Virtual Machine (VM) allocation in cloud DC. Here, to perform VM allocation significantly, the multi-objective artificial bee colony optimization was employed. However, due to the limited bandwidth, it was less effective.

(Huang et al., 2021) combined a VM allocation strategy regarding user requirements for cloud DC. This work focused on VM allocation as well as Physical Machine (PM) shutdown. There was a high throughput and better performance. However, this model failed to handle the fault tolerance of the VMs.

(Saxena et al., 2021) intended a multi-objective model-based resource management at the cloud DC. Here, to manage the resources efficiently, an online VM prediction-centric multi-objective load balancing was introduced. This scheme significantly reduced the network traffic. However, owing to the increased size of the DC, it had high VM migration costs.

III. PROPOSED METHODOLOGY FOR SECURE DATA SOURCE DISTRIBUTION IN ACTIVE-ACTIVE DATA CENTER

Here, by using PCSWHA-MT and ECD-DSA, geographical location AP privileges-aware privacy preserved APNSC-based secure data source distribution in active-active DC is implemented. In Figure 1, the structural representation of the research model is given.

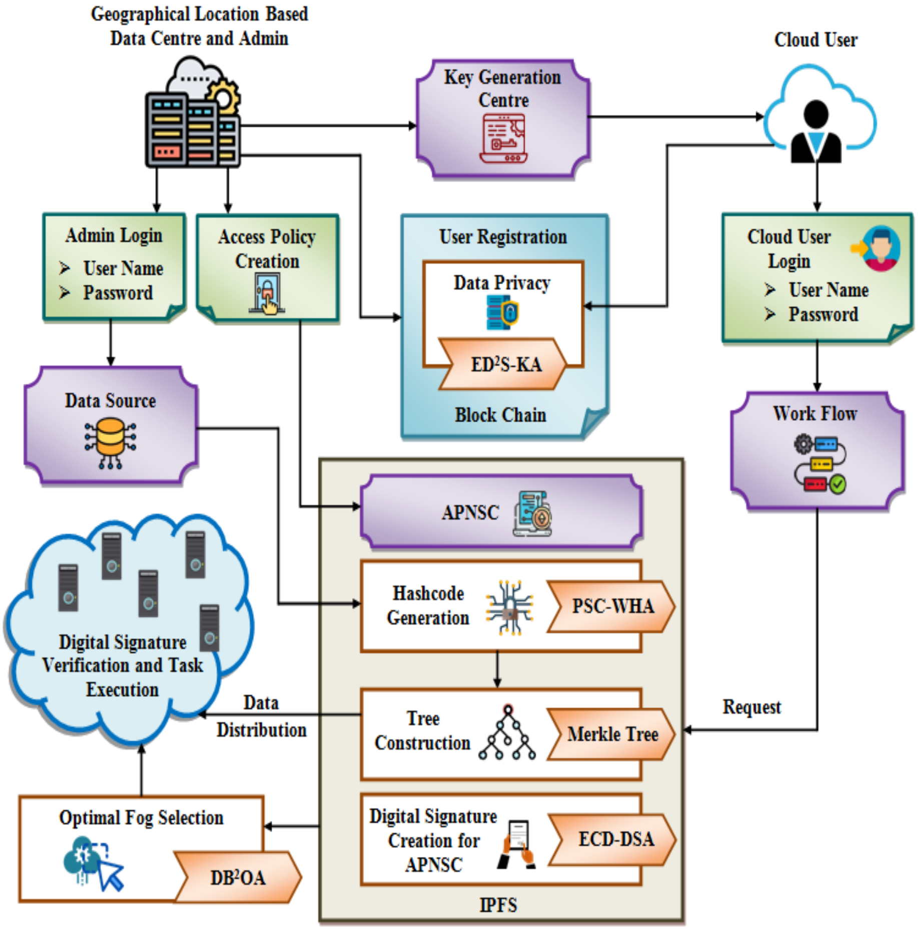


Figure 1: The pictorial depiction of the proposed approach

3.1 User Registration

Initially, by using the user name and password, the geographical location-based DC$\left(\partial_c\right)$, admin $\left(\alpha_n\right)$, and cloud user$\left(\delta_d\right)$ register into the blockchain. A private key $\left(\varphi_{pv}\right)$and a public key $\left(\phi_{pb}\right)$are created during registration for the registered users in the key generation center via the ECD-DSA, which is explained in Section 3.3. The user registration $\left(\Re_{users}\right)$is represented as,

$$\Re_{users} = \left\| \partial_c\left(\varphi_{pv}, \phi_{pb}\right), \alpha_n\left(\varphi_{pv}, \phi_{pb}\right), \delta_d\left(\varphi_{pv}, \phi_{pb}\right) \right\| \tag{1}$$

Afterward, the sensitive data $\left(\mathrm{Z}_{\Xi}\right)$of the registered users is subjected to privacy preservation, which is derived further,

3.1.1 Data privacy

Here, to preserve the user's sensitive data, the Exhaustion Divergence De-swinging based K-Anonymity (ED²S-KA) is established, thus ensuring data privacy. K-Anonymity effectively helps to prevent data breaches. But, owing to the single points of failure, it was ineffective. So, to increase the security level, the exhaustion divergence-based de-swinging technique is included.

Initially, to select the quasi-identifier (attributes), the proposed work introduces the divergence-based de-swinging technique, which is given as,

$$\tau t = \mathrm{Z}_{\Xi}\left(\frac{\partial^{'}}{\partial^{'} \tau t_{pre}} \left(2\tau t_{pre} - \tau t_{cur}\right) + \frac{\partial^{'}}{\partial^{'} \tau t_{pre}} \left(2\tau t_{pre}{}^{2} - \tau t_{cur}\right) \right) \tag{2}$$

Several attributes of the user's sensitive information are preserved in suppression$\left(\gamma_{spr}\right)$ by substituting the asterisk "*".

$$\gamma_{spr} = \mathrm{Z}_{\Xi}\left(\tau t\right) \xrightarrow{replace} "*" \tag{3}$$

Similarly, certain attributes of the sensitive information of the registered users are determined in a specific range category. Therefore, the generalization $\left(\lambda_{gen}\right)$is done as,

$$\lambda_{gen} = \mathrm{Z}_{\Xi} \rightarrow \varsigma < \tau t \leq \varsigma^{'} \tag{4}$$

Here, τt exemplifies the attributes, and ς and specify the random values. So, the privacy-preserved data is mentioned as$\left(\wp_{data}\right)$.

3.1.2 Access policy creation

The AP is created for the DC, which has different kinds of geographical locations to assure privacy. A set of rules that define which individual can access specific resources within a network is termed an AP. Thus, the generated AP $(\rho o\ell)$ is shown as,

$$\rho o\ell = \zeta_{\infty} \cdot \left|\partial_{c}, \alpha_{n}\right| \xrightarrow{send} \Im_{PFS} \tag{5}$$

Where, ζ_{∞} specifies the user's details. Besides, the $\rho o\ell$ is stored in the IPFS$\left(\Im_{PFS}\right)$. The AP along with the nested smart contract between the DC and IPFS and cloud user and IPFS is created in IPFS as,

$$\alpha\rho_{\eta sc} \rightarrow \rho o\ell \oplus \left\langle \zeta_{con}\left(\partial_{c}, \Im_{PFS}\right) \cdot \zeta_{con}\left(\delta_{d}, \Im_{PFS}\right)\right\rangle \tag{6}$$

Here, $\alpha\rho_{\eta sc}$ signifies the generated APNSC.

3.1.3 Admin login

Now, by using the username and password, the admin of the DC logs in to the network. The admin uploads the data source in the DC and then the uploaded data is shared with the IPFS,

$$\alpha_{n} = \Psi_{g} \xrightarrow{upload} \partial_{c} \xrightarrow{share} \Im_{PFS} \tag{7}$$

Here, $g = 1 to G$ portray the number of uploaded data Ψ_{g} .

3.2 IPFS

Usually, a peer-to-peer distributed file system that aims to connect each computing devices with the same system of files is termed an IPFS. Here, the hash code of the uploaded data is created and then added to the Merkle tree.

3.2.1 Hash code generation

Likewise, based on PSC-WHA, the Ψ_{g} is subjected to hash code generation. The WHA is more robust to cryptographic attacks. However, the round keys may compromise, which leads to security issues. Thus, based on Playfair substitution cipher, the round keys are generated, thus elevating data privacy.

- Principally, the initialization vector is determined, which is specifically a 512-bit value. Next, to change the length of the Ψ_{g} as multiple of 512 bits, the Ψ_{g} is padded. So, the padded data is illustrated as $\left(\mathrm{P}_{512}\right)$.

- Then, the padded message is categorized into 512-bit blocks. Next, by using the compression function, each block is processed, which involves a series of permutations. To generate the succeeding hash value, the compression function is implemented iteratively to every block along with the current hash value.

- The padded data is represented in the Playfair cipher substitution-based state matrix (Py) in the proposed work. Next, between the round key and the ciphered state matrix, the XOR operation is performed.

$$XOR = R_{key} \cdot \pi(Py \cdot (\Psi_g)) \tag{8}$$

- Finally, the final hash value is provided as the outcome. The 512-bit whirlpool hash of the original input data is illustrated by the final hash value (Δ_{hash}). The proposed PSC-WHA's pseudo-code is given below,

```
Input: Uploaded data $\Psi_g$
Output: Final hash value $\Delta_{hash}$
Begin
        Initialize $Py$ and $\Psi_g$
                Determine initialization vector
                Perform padding
                Apply XOR
                        $XOR = R_{key} \cdot \pi(Py \cdot (\Psi_g))$
Return $\Delta_{hash}$
End
```

3.2.2 Tree construction

Then, based on Merkle tree, the tree is constructed from the Δ_{hash}. The Merkle root, non-leaf nodes, and leaf nodes are encompassed in a Merkle tree. In Figure 2, The Merkle tree's structure is given.

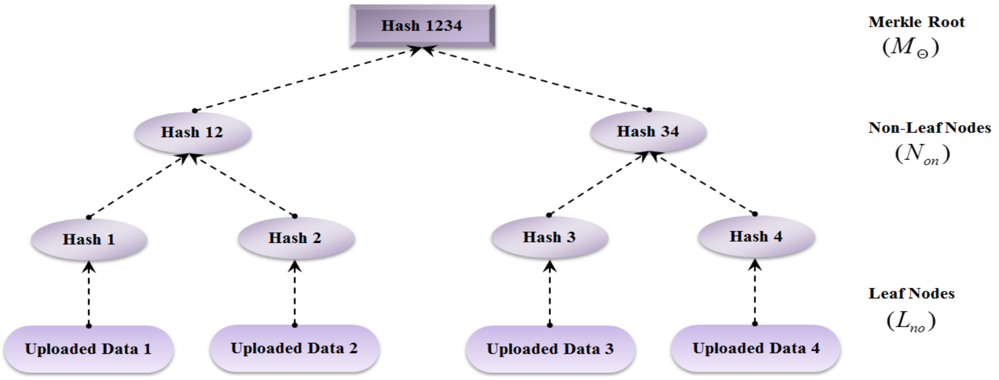


Figure 2: The Merkle tree's structure

Initially, the leaf nodes are paired together and then their hashes are concatenated.

$$L_{no} = \left(\Delta_{hash}\right) \rightarrow \begin{cases} \Delta_1 \xrightarrow{hash} \left(\Psi_1\right) \\ \Delta_2 \xrightarrow{hash} \left(\Psi_2\right) \\ \Delta_3 \xrightarrow{hash} \left(\Psi_3\right) \\ \Delta_4 \xrightarrow{hash} \left(\Psi_4\right) \end{cases} \tag{9}$$

Here, L_{no} specifies the concatenated hashes. Now, to form the parent nodes $\left(N_{on}\right)$, the concatenated hashes are utilized.

$$N_{on} = L_{no} \left| \begin{matrix} \Delta_{1,2} \xrightarrow{hash} \left(\Psi_1 \,\|\, \Psi_2\right) \\ \Delta_{3,4} \xrightarrow{hash} \left(\Psi_3 \,\|\, \Psi_4\right) \end{matrix} \right. \tag{10}$$

Likewise, until a single hash (Merkle root) is obtained, the above-mentioned steps are repeated iteratively.

$$M_{\Theta}\left(1234\right) = \Delta_{1,2} \,\|\, \Delta_{3,4} \xrightarrow{hash} \left(\Psi_{1,2} \,\|\, \Psi_{3,4}\right) \tag{11}$$

Here, M_{Θ} signifies the Merkle root. Then, by verifying whether the newly generated hash code is already presented in the Merkle tree or not, the tree is updated. Then, in the blockchain network, the uploaded data is distributed among the available active DC.

3.3 Cloud user login

By using the user name and password, the cloud user logs in to the network. Now, to access the data sources from the DC, the cloud user assigns the workflow. According to the cloud user request and APNSC, the DC is selected. Also, by using the ECD-DSA, the APNSC $\alpha\rho_{\eta sc}$ is digitally signed, and then the workflow is sent to the corresponding DC. The DSA is more effective in generating digital signatures with smaller lengths. But, during parameter generation, the modulus operator is utilized, thus increasing the time complexity. So, to improve the computational efficiency, Euclidean ceiling division is employed. Mostly, the large prime number and the prime divisor are initialized as,

$$\left\langle P_{\nabla}, D^{\Omega} \right\rangle \rightarrow \left| 1024\, bits, 160\, bits \right| \tag{12}$$

Then, the generator (G_Φ) is chosen as,

$$G_\Phi = j^{(P_\nabla - 1)/D^\Omega} \bmod P_\nabla \tag{13}$$

Here, j signifies the integer. Next, the φ_{pv} and ϕ_{pb} are created as,

$$\varphi_{pv} = \left|0 < \varphi_{pv} < D^\Omega\right| \tag{14}$$

$$\phi_{pb} = G_\Phi^{\varphi_{pv}} \bmod P_\nabla \tag{15}$$

Then, to generate a hash value (ηh_{acc}), the $\alpha\rho_{\eta sc}$ is hashed using the cryptographic hash function.

$$Q_1 = \left(G_\Phi^{w} \bmod P_\nabla\right) \bmod D^\Omega \tag{16}$$

The Euclidean ceiling division is employed by the proposed method to generate the parameter as,

$$R_1 = \left(w^{-1}(\eta h_{acc}) + \frac{\varphi_{pv}}{Q_1} \right) \bmod D^\Omega \tag{17}$$

Where, Q_1 and R_1 designate the parameters and w represent the random number. Next, the Q_1 and R_1 are paired to create the signature as below,

$$Sig = (Q_1, R_1) \cdot \alpha\rho_{\eta sc} \tag{18}$$

The signature pair is sent along with the message. The signature is valid If $\left(0 < Q_1 < D^\Omega\right)$ and $\left(0 < R_1 < D^\Omega\right)$ is true. Else, the signature is invalid. Next,

$$a = R_1^{-1} \bmod D^\Omega \tag{19}$$

$$b = (\eta h_{acc} \cdot a) \bmod D^\Omega \tag{20}$$

$$c = (Q_1 \cdot a) \bmod D^\Omega \tag{21}$$

$$x = \left(\left(G_\Phi^{b} \cdot G_\Phi^{c}\right) \bmod P_\nabla\right) \bmod D^\Omega \tag{22}$$

Here, a, b, c and x designate the parameters.

$$\begin{cases} IF(x == Q_1), & valid \\ ELSE, & invalid \end{cases} \tag{23}$$

The signature is valid if x is equal to Q_1. Else, the signature is invalid.

3.4 Optimal fog selection

Then, to diminish the transmission delay, the optimal fog is selected by using the De Bruijn Botox Optimization Algorithm (DB^{2}OA). A meta-heuristic optimization, which is inspired by the botox mechanisms, is termed the BOA. Through an iterative process, the BOA algorithm proficiently produces the optimal solution. However, because the position is updated in a random way, it had a local optimal solution. Therefore, to update the new position, the De Bruijn sequence is employed, thus upgrading the optimization quality. Here, the number of fog nodes (members) is assumed as the individuals seeking Botox injections. Primarily, the $k=1,2,\ldots K$ number of fog nodes F_k is initialized as,

$$F_{k,l} = C_l + \Re d_{k,l} * \left(B_l - C_l\right); \quad l = 1 to L \tag{24}$$

Where, L characterizes the number of decision variables. Besides, by considering the minimum response time, the fitness value $(\aleph it)$ is calculated.

$$\aleph it = \min(r_t, ly) \tag{25}$$

Then, the number of muscles requiring Botox injection (decision variables) is updated as,

$$l^{'} = \left(1 + \frac{l}{itr}\right) \le L \tag{26}$$

Here, $l^{'}$ signifies the updated number of muscles requiring botox injection and itr specifies the iteration counter. Likewise, based on patient needs, the amount of Botox injection for all population members is computed,

$$\beta x_o = \begin{cases} F_{k,l}^{\ mean} - F_{k,l}, & IF\left(itr < \dfrac{itr^{\max}}{2}\right) \\ F_{k,l}^{\ best} - F_{k,l}, & ELSE \end{cases} \tag{27}$$

Where, $itr^{\max}$ portrays the maximum iterations, $F_{k,l}^{\ mean}$ signifies the mean population position, $F_{k,l}^{\ best}$ signifies the best member, and βx_o shows the $o = 1 to O$ amount of botox injection. Afterward, according to the simulation of botox injection to the facial muscles, the position of the member is updated. To update the new position, the proposed method introduces the De Bruijn sequence $\left(\Re d_{k,l}\right)$.

$$\Re d_{k,l} = \frac{\left(F_k\right)^{k^K - 1}}{K} \tag{28}$$

$$F_{k,l}^{\ new\Theta} = F_{k,l} + \Re d_{k,l} \cdot \beta x_o \tag{29}$$

$$F_{k,l} = \begin{cases} F_{k,l}^{\ new\Theta}, & IF\left(\aleph it_{new} < \aleph it\right) \\ F_{k,l}, & ELSE \end{cases} \tag{30}$$

Lastly, the selected optimal fog is exemplified as (Opt). The proposed DB²OA's pseudo-code is given below,

Input: Fog nodes F_k

Output: Optimal fog Opt

```
Begin
    Initialize $F_k$, $Opt$ and $F_{k,l}^{best}$
        Perform $F_{k,l} = C_l + \Re d_{k,l} * (B_l - C_l); \quad l = 1 to L$
        Calculate $\aleph it = \min(r_t, ly)$
        Determine $\Re d_{k,l} = \frac{(F_k)^{k^K - 1}}{K}$
        Update position,
            $F_{k,l}^{new\Theta} = F_{k,l} + \Re d_{k,l} \cdot \beta x_o$
Return $Opt$
End
```

3.5 Task execution

Finally, by verifying the geographical location-based access policies and authorized users, the task is executed in the cloud DC. Therefore, the created digital signature is verified and the task execution is done.

IV. RESULTS AND DISCUSSION

Here, to showcase the proposed model's efficacy, the performance analysis is done and it is implemented in the working platform of PYTHON.

4.1 Performance analysis

The research methodology's performance is evaluated by analogizing it with prevailing techniques.

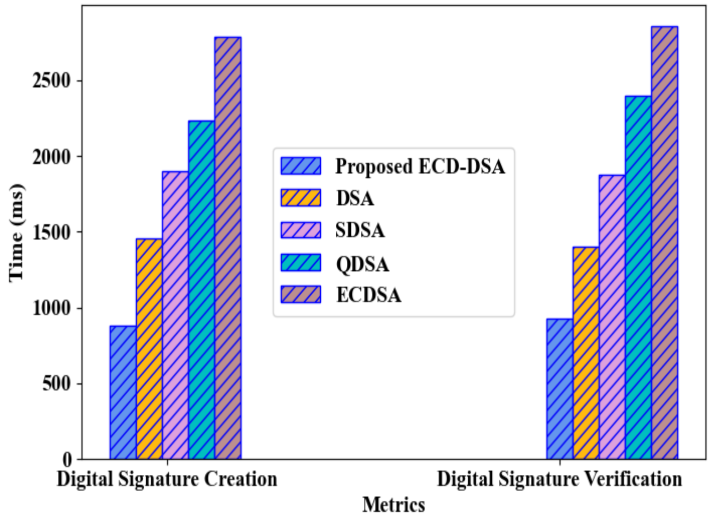
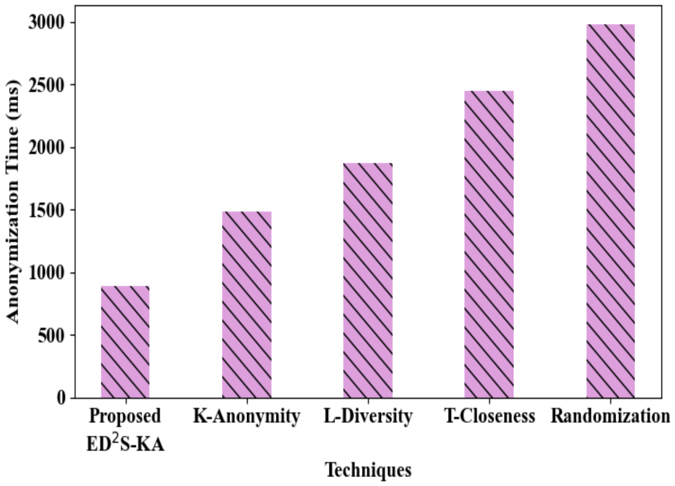
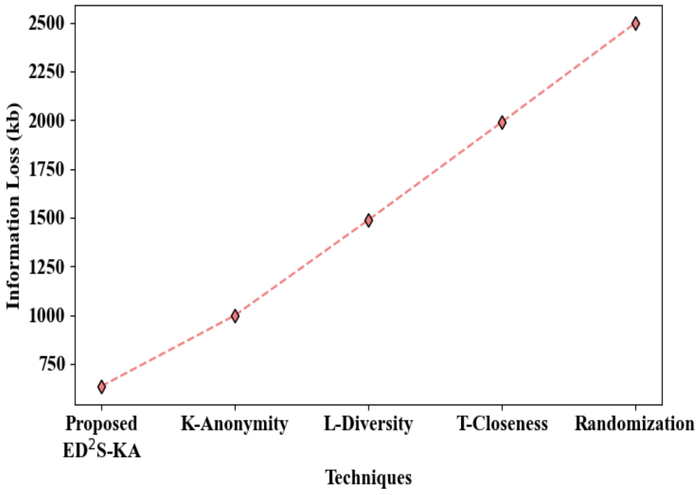


Figure 3: Performance analysis of the proposed ECD-DSA

In Figure 3, regarding Digital Signature Creation Time (DSCT) and Digital Signature Verification Time (DSVT), the performance of the proposed ECD-DSA and existing methods like DSA, Simple DSA (SDSA), Qualified DSA (QDSA), and Elliptic Curve DSA (ECDSA) is validated. Due to the Euclidean ceiling division, the proposed work's performance is improved. For DSCT and DSVT, the ECD-DSA obtained 876ms and 924ms; while, the prevailing techniques attained 2093ms and 2130ms, respectively. Thus, the proposed method had low time complexity.



(a)



(b)

Figure 4: Performance analysis of the proposed ED²S-KA based on (a) anonymization time and (b) information loss

Owing to the exhaustion divergence-based de-swinging technique, the proposed work had better efficiency. In Figure 4, the anonymization time and information loss of the proposed ED²S-KA and traditional algorithms like K-Anonymity, L-Diversity, T-Closeness, as well as randomization are depicted. For anonymization time and information loss, the ED²S-KA achieved 892ms and 635kb;

but, the traditional works had limited outcomes. Thus, the proposed work has a high-security level.

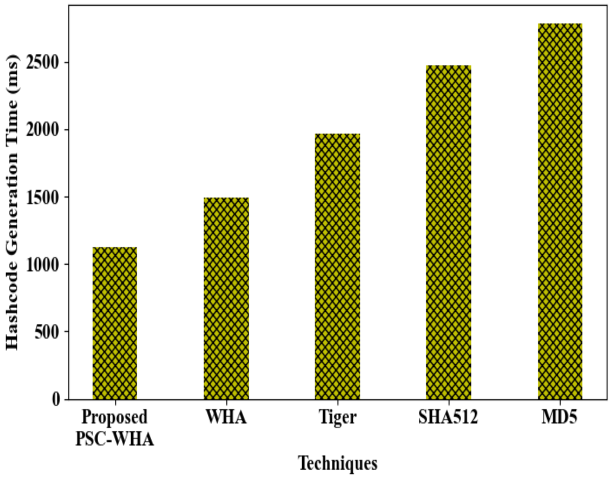
Table 1: Privacy-preserving rate

Methods	Privacy-Preserving Rate (%)
Proposed ED^{2}S-KA	98.4876
K-Anonymity	95.1249
L-Diversity	93.6321
T-Closeness	90.2954
Randomization	86.5478

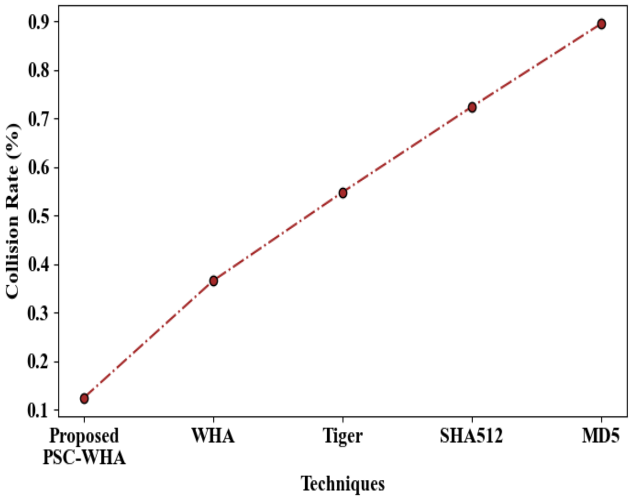
Table 2: Disclosure risk

Methods	Disclosure Risk (%)
Proposed ED^{2}S-KA	9.5678
K-Anonymity	14.2147
L-Diversity	18.6365
T-Closeness	22.4448
Randomization	27.3257

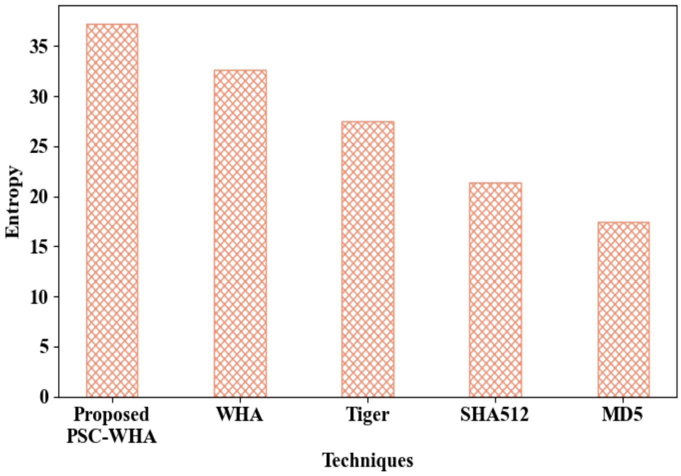
Regarding Privacy Preservation Rate (PPR) and Disclosure Risk (DR), the performance analysis of the proposed ED^{2}S-KA and traditional works are done, which is depicted in Tables 1 and 2. For PPR and DR, the ED^{2}S-KA attained 98.48% and 9.56%; likewise, the existing approaches obtained 91.40% and 20.65%, respectively. Thus, in privacy preservation, the proposed methodology had higher dominance.


Hashcode Generation Time (ms)
0
500
1000
1500
2000
2500
Proposed PSC-WHA
WHA
Tiger
SHA512
MD5
Techniques

(a)


Collision Rate (%)
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Proposed PSC-WHA
WHA
Tiger
SHA512
MD5
Techniques

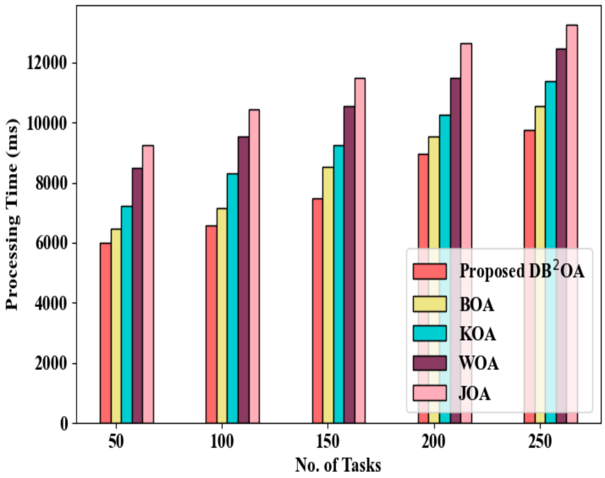
(b)



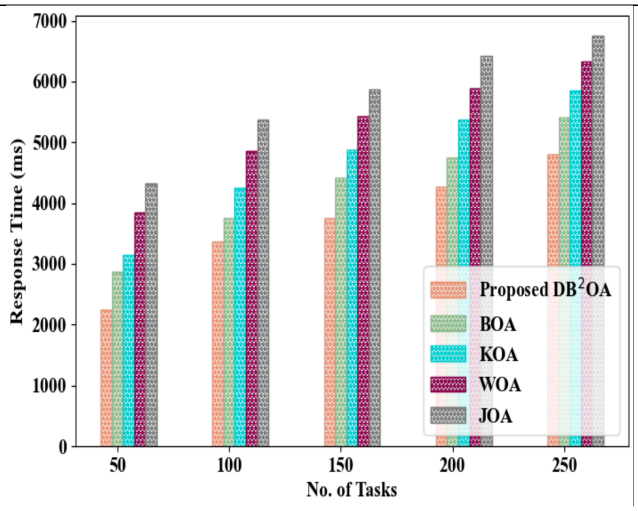
(c)

Figure 5: Performance assessment of the proposed PSC-WHA regarding (a) hash code generation time, (b) collision rate, and (c) entropy

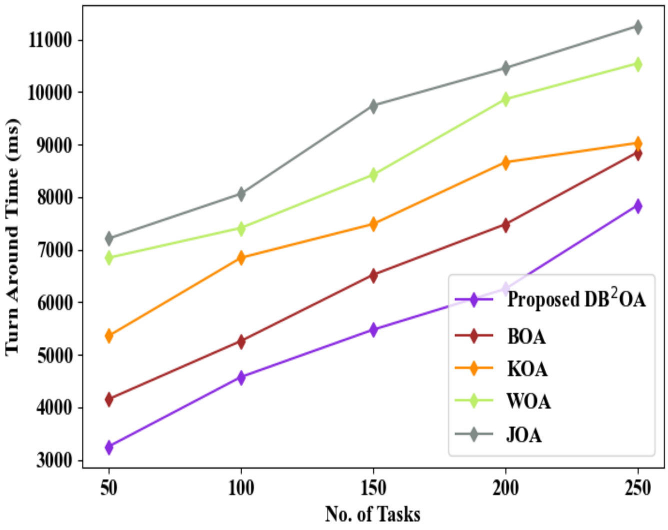
In Figure 5, the hash code of the proposed PSC-WHA and conventional algorithms like WHA, Tiger, SHA512, and Message Digest 5 (MD5) are evaluated. To increase the model's performance, the PSC is employed. But, when analogized to the proposed approach, the traditional method had poorer outcomes. Hash code generation time, collision rate, and entropy of 1124ms, 0.1248%, and 37.21 were acquired by the proposed work. Lastly, the proposed work had high supremacy.



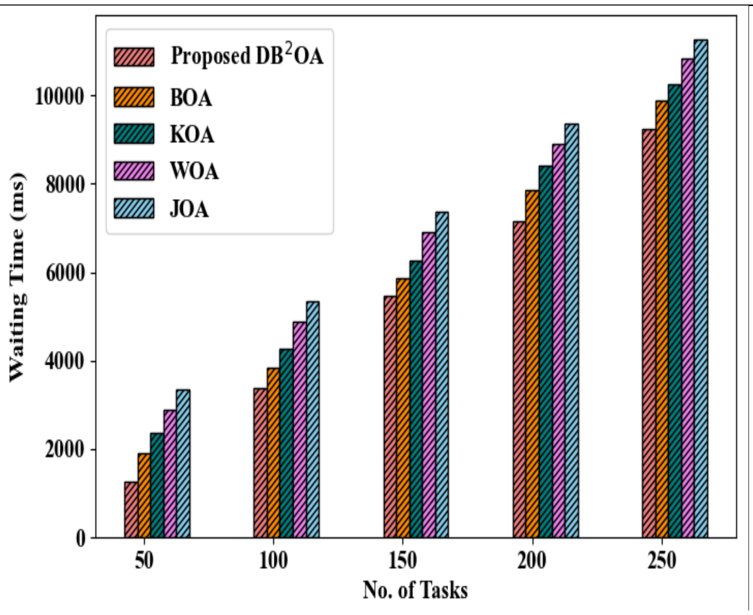
(a)


Response Time (ms)
7000
6000
5000
4000
3000
2000
1000
0
50
100
150
200
250
No. of Tasks
Proposed DB^{2}OA
BOA
KOA
WOA
JOA

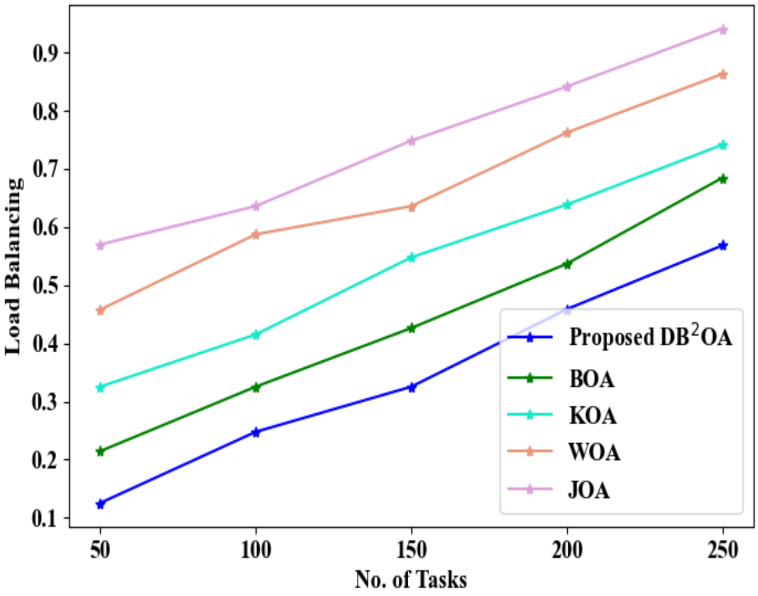
(b)


Turn Around Time (ms)
11000
10000
9000
8000
7000
6000
5000
4000
3000
50
100
150
200
250
No. of Tasks
Proposed DB^{2}OA
BOA
KOA
WOA
JOA

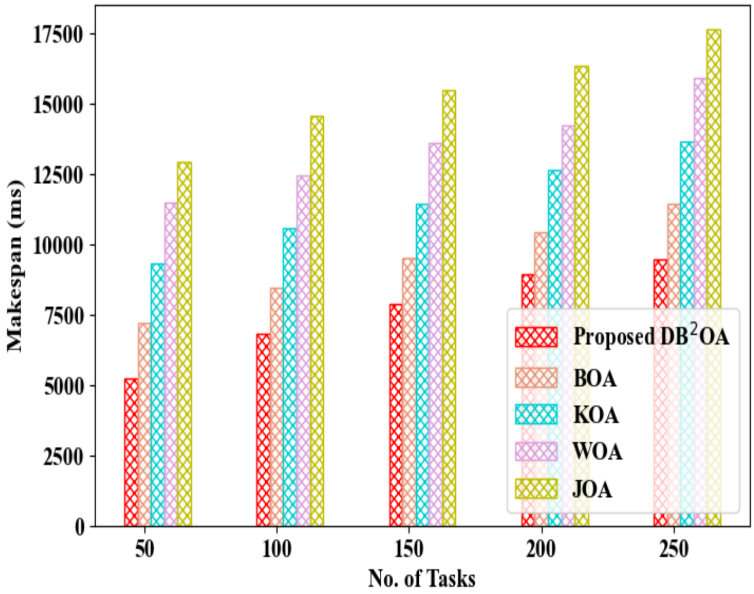
(c)


Proposed DB^{2}OA
BOA
KOA
WOA
JOA
Waiting Time (ms)
0
2000
4000
6000
8000
10000
50
100
150
200
250
No. of Tasks

(d)


Load Balancing
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
50
100
150
200
250
No. of Tasks
Proposed DB^{2}OA
BOA
KOA
WOA
JOA

(e)



(f)

Figure 6: Performance assessment of the proposed DB²OA regarding (a) processing time, (b) response time, (c) turnaround time, (d) waiting time, (e) load balancing, and (f) makespan

The performance of the proposed DB²OA and existing methods like BOA, Kookaburra Optimization Algorithm (KOA), Whale Optimization Algorithm (WOA), and Jaya Optimization Algorithm (JOA) are validated in Figure 6. For processing time, response time, turnaround time, waiting time, load balancing, and makespan, the DB²OA achieved 5986ms, 2254ms, 3256ms, 1254ms, 0.1254, and 5263ms, respectively for 50 tasks. Thus, in optimal fog selection, the proposed work has impressive outcomes.

4.2 Comparative assessment

To prove the model's consistency, the comparative analysis is carried out.

Table 3: Comparative validation

Author's name	Algorithm	Response time (ms)	Makespan (ms)
Proposed method	DB²OA	2254	5263
(Praveenchandar & Tamilarasi, 2020)	Preference-based task scheduling	25000	1500000
(Alsadie, 2021)	Non-dominated sorting genetic algorithm	-	4000000
(Negi et al., 2021)	Improved-PSO	-	41000

(Jena et al., 2022)	Modified-PSO		9671
(Sohani & Jain, 2021)	Predictive Priority-based Modified Heterogeneous Earliest Finish Time (PMHEFT) algorithm	-	500000

The performance of the proposed approach and associated frameworks are compared in Table 3. For response time and makespan, the DB^{2}OA achieved 2254ms and 5263ms. To perform task allocation in cloud DC, the prevailing PMHEFT, PSO, and genetic algorithm were implemented. But, due to the random position updating process, the traditional schemes were ineffective. Lastly, high superiority was attained by the proposed approach.

V. CONCLUSION

Here, by using PCS-WHA and ECD-DSA, a privacy-preserved APNSC-based reliable data source distribution in an active-active DC is proposed. To ensure secure data distribution, the digital signature verification of the APNSC was done. Moreover, the DB^{2}OA-based optimal fog selection helps to diminish the time delay. Also, as per the experimental results, a processing time of 5986ms is attained by the proposed DB^{2}OA. Likewise, a PPR of 98.48% is attained by the proposed ED^{2}S-KA, thus showing high trustworthiness. Thus, in active-active DC management, the proposed framework achieved higher significance. However, during data distribution and task execution, the proposed work only focused on security.

Future scope: To increase the model's consistency, the uploaded data and requested workflow will be encrypted in the future.

REFERENCES

1. Ahmad, Z., Jehangiri, A. I., Ala'anzy, M. A., Othman, M., Latip, R., Zaman, S. K. U., & Umar, A. I. (2021). Scientific Workflows Management and Scheduling in Cloud Computing: Taxonomy, Prospects, and Challenges. *IEEE Access*, *9*, 53491–53508. https://doi.org/10.1109/ACCESS.2021.3070785
2. Alsadie, D. (2021). A Metaheuristic Framework for Dynamic Virtual Machine Allocation with Optimized Task Scheduling in Cloud Data Centers. *IEEE Access*, *9*, 74218–74233. https://doi.org/10.1109/ACCESS.2021.3077901
3. Athanere, S., & Thakur, R. (2022). Blockchain based hierarchical semi-decentralized approach using IPFS for secure and efficient data sharing. *Journal of King Saud University - Computer and Information Sciences*, *34*(4), 1523–1534. https://doi.org/10.1016/j.jksuci.2022.01.019
4. Awan, I. A., Shiraz, M., Hashmi, M. U., Shaheen, Q., Akhtar, R., & Ditta, A. (2020). Secure Framework Enhancing AES Algorithm in Cloud Computing. *Security and Communication Networks*, *2020*(1), 1–16. https://doi.org/10.1155/2020/8863345
5. Chen, S., Li, P., Ji, H., Yu, H., Yan, J., Wu, J., & Wang, C. (2021). Operational flexibility of

active distribution networks with the potential from data centers. *Applied Energy*, *293*, 1–10. https://doi.org/10.1016/j.apenergy.2021.116935

6. Helali, L., & Omri, M. N. (2021). A survey of data center consolidation in cloud computing systems. *Computer Science Review*, *39*, 1–28. https://doi.org/10.1016/j.cosrev.2021.100366
7. Hogade, N., Pasricha, S., & Siegel, H. J. (2022). Energy and Network Aware Workload Management for Geographically Distributed Data Centers. *IEEE Transactions on Sustainable Computing*, *7*(2), 400–413. https://doi.org/10.1109/TSUSC.2021.3086087
8. Huang, Y., Xu, H., Gao, H., Ma, X., & Hussain, W. (2021). SSUR: An Approach to Optimizing Virtual Machine Allocation Strategy Based on User Requirements for Cloud Data Center. *IEEE Transactions on Green Communications and Networking*, *5*(2), 670–681. https://doi.org/10.1109/TGCN.2021.3067374
9. Jena, U. K., Das, P. K., & Kabat, M. R. (2022). Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment. *Journal of King Saud University - Computer and Information Sciences*, *34*(6), 2332–2342. https://doi.org/10.1016/j.jksuci.2020.01.012
10. Kang, P., Yang, W., & Zheng, J. (2022). Blockchain Private File Storage-Sharing Method Based on IPFS. *Sensors*, *22*(14), 1–12. https://doi.org/10.3390/s22145100
11. Katal, A., Dahiya, S., & Choudhury, T. (2023). Energy efficiency in cloud computing data centers: a survey on software technologies. In *Cluster Computing* (Vol. 26, Issue 3). Springer US. https://doi.org/10.1007/s10586-022-03713-0
12. Morkevičius, N., Liutkevičius, A., & Venčkauskas, A. (2023). Multi-Objective Path Optimization in Fog Architectures Using the Particle Swarm Optimization Approach. *Sensors*, *23*(6), 1–20. https://doi.org/10.3390/s23063110
13. Negi, S., Rauthan, M. M. S., Vaisla, K. S., & Panwar, N. (2021). CMODLB: an efficient load balancing approach in cloud computing environment. In *Journal of Supercomputing* (Vol. 77, Issue 8). https://doi.org/10.1007/s11227-020-03601-7
14. Praveenchandar, J., & Tamilarasi, A. (2020). Dynamic resource allocation with optimized task scheduling and improved power management in cloud computing. Journal of Ambient Intelligence and Humanized Computing, 12(3), 4147–4159. https://doi.org/10.1007/s12652-020-01794-6
15. Saxena, D., Singh, A. K., & Buyya, R. (2021). OP-MLB: An Online VM Prediction-Based Multi-Objective Load Balancing Framework for Resource Management at Cloud Data Center. *IEEE Transactions on Cloud Computing*, *10*(4), 2804–2816. https://doi.org/10.1109/TCC.2021.3059096
16. Sayadnavard H, M., Toroghi Haghighat, A., & Rahmani, A. M. (2022). A multi-objective approach for energy-efficient and reliable dynamic VM consolidation in cloud data centers. *Engineering Science and Technology, an International Journal*, *26*, 1–13. https://doi.org/10.1016/j.jestch.2021.04.014
17. Sohani, M., & Jain, S. C. (2021). A Predictive Priority-Based Dynamic Resource Provisioning Scheme with Load Balancing in Heterogeneous Cloud Computing. *IEEE Access*, *9*, 62653–62664. https://doi.org/10.1109/ACCESS.2021.3074833
18. Sultana, T., Almogren, A., Akbar, M., Zuair, M., Ullah, I., & Javaid, N. (2020). Data sharing system integrating access control mechanism using blockchain-based smart contracts for IoT devices. *Applied Sciences (Switzerland)*, *10*(2), 1–21. https://doi.org/10.3390/app10020488

19. Ulabedin, Z., & Nazir, B. (2021). Replication and data management-based workflow scheduling algorithm for multi-cloud data centre platform. *Journal of Supercomputing*, *77*(10), 10743–10772. https://doi.org/10.1007/s11227-020-03541-2
20. Zarrin, J., Wen Phang, H., Babu Saheer, L., & Zarrin, B. (2021). Blockchain for decentralization of internet: prospects, trends, and challenges. *Cluster Computing*, *24*(4), 2841–2866. https://doi.org/10.1007/s10586-021-03301-8