Heliyon 10 (2024) e26312

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Slide-block: End-to-end amplified security to improve DevOps resilience through pattern-based authentication

Gopalakrishnan Sriraman[a,*], Shriram R[a]

[a] *Department of Computing Science and Engineering, VIT Bhopal University, Sehore, MP, India*

ARTICLE INFO

Index Terms:
Security in DevOps-Cloud
Vulnerability management
Pattern-based authentication
Tri-level access control
Secure data storage
Attack detection and mitigation

ABSTRACT

DevOps represents the fusion of cultural philosophies, tools, and practices that rapidly enhance an organization's capacity to deploy services and applications. Cloud-based tools, a subset of DevOps services, facilitate collaboration between development and operations teams within an organization. However, persistent challenges such as inadequate security management, substantial leakage of sensitive data, and system/service unavailability pose significant threats to sustainability. We propose an end-to-end enhanced security framework to fortify DevOps resilience by implementing authentication and vulnerability management through the Slide-Block methodology. Our approach comprises four sequential processes: pattern-based authentication, tri-level access control, privacy-focused data storage, and vulnerability management and correction. Initially, we establish candidate legitimacy through pattern-based authentication using the Magnificent Chacha-Poly 1305 algorithm. Subsequently, we devise effective access policies using the Enhanced Deep Deterministic Policy Gradient (EDDPG) algorithm, employing tri-level access control based on trust value, attributes, and roles for optimal user and developer selection via the African Vulture Optimization Algorithm (AVOA). Moreover, we encrypt data in transit and at rest using Mcha-Poly 1305, considering sensitivity, and store it in a blockchain to enhance data privacy. Our approach incorporates a sliding window blockchain for secure data transmission and storage. Finally, we identify and address attack and application-based issues using the Tweak Naive Bayes (Tweak-NB) algorithm and Intruder Vulnerability Scanner (IVS). Our Slide-Block framework demonstrates superior performance in detection rate, authentication time, packet loss, security strengthening, communication overhead, and latency compared to existing models.

1. Introduction

In an Information Technology (IT) environment, "DevOps" is one of the software development cultures that is meant for integrating the software and development teams to boost up development activities with higher convey rates [1–3]. By utilizing DevOps-based software development we can achieve continuous deployment, continuous monitoring, continuous development, security, flexibility, and expandability [4]. To be more specific, DevOps helps to deliver the developed software either on the customer side or server side in a continuous pace in short and fast delivery cycles [5]. Cloud computing is another paradigm that provides on-demand services and infrastructure [6,7].

The combined adoption of cloud computing and DevOps enhances production speed and resilience [8–10]. Specifically,

* Corresponding author.
E-mail addresses: gopalakrishnan.sriraman2019@vitbhopal.ac.in (G. Sriraman), shriram.r@vitbhopal.ac.in (S. R).

https://doi.org/10.1016/j.heliyon.2024.e26312
Received 9 July 2023; Received in revised form 6 February 2024; Accepted 9 February 2024
Available online 14 February 2024

incorporating cloud computing in software development empowers developers to manage their tools and offers additional components for continuous application automation, building, testing, and monitoring [11,12]. Despite these benefits, careful consideration and research are necessary to address potential drawbacks and design improved models. Security emerges as a primary concern, as the lack of empirical research in DevOps has led the research community to hesitate in integrating security practices [13].

Only limited works in recent years to provide a recommendation of security practices but it doesn't provide end-to-end security practices as research articles [14]. Authentication and access control are basic security operation that restricts unauthorized access [15]. However, the existing works on authentication and access control provide limits with poor metrics as they only consider limited metrics to authenticate and access the users, developers, and operators [16,17]. The existing works lack security during continuous monitoring, leading to several cyber security attacks [18]. Machine learning and data engineering are emerging technologies that enable intelligence [19,20]. Thus, the proposed work addressed the challenges and research gaps that are faced in the former works further in this work, the betterment solution is to accommodate for secure and practical application in a cloud environment with DevOps.

1.1. Research aim & objectives

The main aim of this research is to enhance the security for the betterment of the resilience of DevOps by integrating cloud computing, machine learning and data engineering. In addition, the research also identifies the problems of improper security management, inefficacy authorization, enormous sensitive data leakage and system and service unavailability. Moreover, the main objective of this research is to enhance security for the betterment of the resilience of DevOps by integrating cloud computing through effective vulnerability management based on continuous monitoring.

- We have integrated an effective authentication mechanism to enhance secret management, allowing only legitimate candidates access to the applications. We have developed significant access control policies to improve authorization and implement trust-level-based access control.
- To enhance data privacy and prevent sensitive data leakage, we have implemented two categories of data encryption. Continuous monitoring and rectification, aided by a security monitoring agent, have been employed to address application-based issues and amplify system and service availability.
- In order to strengthen security, we have adapted attack detection and enhanced blockchain to improve Quality of Service (QoS)

1.2. Research Motivations

To enhance the security of cloud-based applications with integrated DevOps, most of the existing works perform continuous monitoring for attack detection. However, the current works are limited by improper security management, inefficacy authorization, enormous sensitive data leakage and system and service unavailability.

I. **Improper Security Management:** In most of the existing works, insufficient metrics and ineffective authentication mechanisms were utilized, which led to improper security management. In addition to that, in several earlier works, only the application users were authenticated; however, the lack of authenticating the developers and resource owners leads to improper security management.

II. **Inefficacy authorization:** In most of the existing works, the access control policies are only provided for application users, and several previous works don't fabricate effective or secure access policies where this inefficacy authorization leads to high-security breaches. Furthermore, the access control policies are stored without any security measures where attackers can easily tamper and modify the policies that affect the security level of cloud-based applications with integrated DevOps.

III. **Enormous Sensitive Data Leakage:** In most of the previous works, the data privacy and sensitive data were not secured effectively, and the communication between the cloud and users was performed without any privacy concerns that led to high data leakage. Moreover, the rest of the data, which is sensitive data like organization policies, access control policies, passwords, and other sensitive data, were stored in the cloud without encryption or any other security measure where the attackers can effortlessly access the data.

IV. **System and Service Unavailability:** In several existing works, the application users' sides are continuously monitored where the inconsideration of application settings, bugs, data errors, etc.., leads to high system and service unavailability. Besides, the bugs and data errors are continuously monitored manually, and immediate remediation is not taken where this ineffective continuous monitoring leads to high system and service unavailability.

1.3. Research contributions

Improving the resilience of DevOps thereby enhancing security is the primary target of our research. To achieve that, we have proposed several requisite contributions that are explained below as follows.

- All candidates undergo pattern-based authentication to secure the application and network, with password encryption using the MCha-Poly 1305 encryption algorithm. This algorithm operates efficiently, allowing only legitimate candidates access to the service, and their credentials are stored securely in the blockchain.

- We generate tri-level access control policies using EDDPG, assigning them to candidates based on their attributes, roles, and trust values. AVOA is employed to optimize access control.
- To ensure communication and transaction certainty, we implement a privacy-focused data storage mechanism that encrypts data using the MCha-Poly 1305 encryption algorithm, minimizing the risk of highly sensitive data leakage.
- Continuous monitoring amplifies system and service availability, identifying attacks and application-based issues through an intruder vulnerability scanner and the Tweak NB algorithm.

1.4. *Paper organization*

The rest of the paper is organized as follows: Section II illustrates state-of-the-art research with its gaps. Section III delineates the foremost problem statement which is faced by cloud-based applications. Section IV articulates the proposed Slides-Block framework, encompassing an appropriate diagram, mathematical equations, algorithm, and pseudocode. Section V exemplifies the experimental analysis of simulation setup, comparative analysis, and research summary. Finally, Section VI concludes the proposed Slides-Block framework.

2. Literature survey

In this section, we have briefly described the state-of-arts with its limitations faced in DevOps-cloud-based applications. Furthermore, this section is divided into three sub-sections which are defined below,

2.1. *Analysis of attack & malware detection*

The paper [21] introduced an auto-encoder entrenched-based mechanism for detecting advanced persistent threat (APT) attacks. Users underwent authentication through a two-factor authentication system based on the OTP scheme. We employed the auto-encoder neural network to analyze informative features derived from unsupervised network traffic. We performed feature extraction and dimension reduction using Principal Component Analysis (PCA). Subsequently, we added a softmax regression layer to the top layer of the auto-encoder network to classify APT attacks. The detection of the attack prompted the strengthening of cloud-based security.

In a different study [22], the author introduced an approach to detect DDoS attacks in a cloud computing environment to reduce misclassification errors during DDoS attack detection. Initially, the study applied feature selection schemes, including mutual information (MI) and random forest feature importance (RFFI). Subsequently, the study utilized various algorithms for classifying DDoS attacks, such as random forest, gradient boosting, weighted voting ensemble (WVE), k nearest neighbour, and logistic regression. Ultimately, the random forest algorithm demonstrated superior performance compared to the others [23]. implemented a secure SaaS approach for detecting and mitigating attacks. The Deep Belief Network (DBN) is utilized for attack detection, with the weight and activation function fine-tuned through the Median Fitness-oriented Sea Lion Optimization Algorithm (MFSLnO). Upon detecting an attack, the system transitioned control to a lightweight bait mechanism, ensuring the reliable mitigation of the most common attack nodes without disrupting routine connections. The evaluation of the proposed work focused on assessing the packet loss ratio and throughput.

In [24], researchers proposed a method for real-time detection of attacks in the cloud environment. The study initially identified attacks in the application layer by employing multiple machine learning algorithms, such as the multi-layer perceptron (MLP) and random forest (RF), utilizing the Scikit ML library and big data architecture. The researchers optimized the model's performance to decrease prediction time, achieving superior accuracy with the random forest algorithm compared to other approaches.

The author offered an intelligent behavior-based malware detection framework in a cloud environment [25]. Multiple virtual

Table 1
Limitations of attack & malware detection.

Reference	Objective	Methods/Algorithms	Limitations
[21]	To utilize auto-encoder based mechanism for APT attack detection.	Auto-encoder neural network & PCA	However, this algorithm misinterprets the significant variable that leads to ineffective attack detection.
[22]	To perform DDoS attack detection in cloud computing using MI.	RFFI, MI, GB, WVE, KNN, RF & LR	Anyhow a MI and RF method selects the redundant and irrelevant feature that leads to high false positive rate.
[23]	To design a method for attack detection and mitigation using secure SaaS approach.	DBN, MFSLnO & lightweight bait mechanism (LBM)	Once intrusion is detected the LBM was performed for mitigation, but ineffective attack mitigation and countermeasure affects the security.
[24]	To method for real time attack detection in the cloud environment.	MLP & RF	Anyhow, RF generates numerous of trees while classification that increase complexity.
[25]	To offer an intelligent behavior-based malware detection framework.	RF	However, lack of ensuring user legitimacy affects the network security.
[26]	To perform an effective IDS CSS algorithm entrenched DBN for detecting suspicious intrusion.	Chronological salp swarm (CSS), DBN & Fuzzy entropy	Anyhow, consideration of inadequate features for performing IDS affects the detection rate.
[27]	To introduce hybrid DL approach for an efficient intrusion detection system.	Cu-LSTMGRU & Pearson's correlation coefficient (PCC)	PCC this algorithm cannot effectively differentiate among dependent and independent variables.

machines initially collect malware data, examining distinctive features and selecting effective ones. Then, the selected features are fed into learning-based and rule-based detection agents to detect whether the data is normal or malware using several machine learning algorithms effectively. The proposed methodology can detect both known attacks and unknown attacks effectively. Finally, the proposed work enhances security by effectively attack detection using random forest. An effective IDS chronological salp swarm algorithm entrenched deep belief network was designed to detect suspicious intrusion in the cloud [26]. Initially, we designed this method by integrating the chronological concept with the Salp swarm algorithm. We established a fitness function to seek an optimal solution that accepts a low error value. Subsequently, we optimally tuned the weights using this method to identify an efficient solution for detecting intruders. Finally, the designed chronological salp swarm algorithm entrenched deep belief network acquired enhanced performance by the exploitation and exploration facility in search space. Researchers employed a hybrid deep learning approach to create an efficient intrusion detection system [27]. Initially, this work focused on enhancing the efficiency of the Intrusion Detection System (IDS) in analyzing abnormal network traffic. The Pearson correlation feature selection algorithm was utilized for efficient feature selection. The intrusion detection process involved utilizing a recurrent neural network embedded with gated recurrent units (GRU) and enhanced long short-term memory (LSTM), forming Cu-LSTMGRU. Subsequently, the system effectively classified network flows as either malicious or benign. The limitations of attack and malware detection are outlined in Table 1.

2.2. *Analysis of access control mechanism & secure data sharing*

A blockchain-based multi-authority access control mechanism (BMAC) for secure data sharing in a cloud environment was incorporated [28]. Initially, the Shamir secret sharing (S3) technique and permission blockchain were utilized to execute individual attributes and jointly supervised by several authorities. Then it adapts the smart contract to generate tokens for attributes handled over several management domains that minimize the communication and computation overhead on data users. The blockchain aids in recording the process of access control in an auditable and secure way. Finally, the security of the proposed algorithm was examined. Several mechanisms are combined to provide fine-grained access control and secure data sharing [29]. Here, the blockchain, ciphertext-policy attribute-based encryption (CP-ABE), and interplanetary file system (IPFS) – BSSPD were utilized for secure personal data sharing. Initially, the user-centric approach was employed where the data owner encrypts the sharing data and stores it in IPS, which increases the approach’s decentralization. The decryption key and address of the transmitted data were encrypted using CP-ABE as per the specific access policy, the data owner adapted the blockchain to publish his data-correlated information and distribute the keys to data users. The data user whose attributes are eligible for access policy can download and decrypt data. Finally, the ciphertext keyword search was utilized to protect the data privacy of user’s while retrieving the data. Blockchain an entrenched access control technique in a cloud computing environment, was introduced [30]. Initially, in this environment, the data owner (DO) handles an access matrix which was stored in blockchain to illustrate the access policy. Then, the public keys of entire nodes and the access matrix are also stored in blockchain, to assure the security of this system. Here, the DO will encrypt the files that are largely shared once utilizing a symmetric key in a long time. Also, public key to authorized users is encrypted with the symmetric key in parallel within a minimal time. Finally, the proposed mechanism enhanced the security, thereby reducing computation overhead.

The author introduced the Blockchain-based Multi-Authority Access Approach (BMAC) to enhance secure data sharing [31]. Initially, they utilized the Shamir secret sharing approach and the permissioned blockchain Hyperledger Fabric to execute individual attributes. This execution was a collaborative effort supervised by multiple authorities, effectively avoiding a single point of failure. Moreover, blockchain technology established trust between authorities and tokens for attributes generated from smart contracts. These contracts were overseen across various management domains, helping to reduce computation and communication overhead on the data user side. The access control approach was audibly and securely recorded.

In the realm of cloud-based applications, the author implemented a secure data securing mechanism by introducing access control [32]. The Hyperledger Fabric and Attribute-Based Access Control (Fabric-ABAC) were initially proposed for secure data sharing across domains. To address data security issues, a trusted central organization was implemented, and a distributed environment involving stakeholders between parties was developed. The multi-environment was integrated with intelligent contracts, constructing a unified attribute model. The proposed Fabric-ABAC achieved multi-level, auditable access control and fine-grained data security by

Table 2
Limitations of access control & secure data sharing.

Reference	Objective	Methods/Algorithms	Limitations
[28]	To develop BMAC mechanism for secure data sharing in cloud.	BMAC, Permissioned blockchain & S3 technique	BMAC mechanism was performed for secure access control, however, the usage of traditional blockchain leads to ineffective immutability.
[29]	To tangle several mechanisms for providing fine-grained access control and secure data sharing.	CP-ABE & IPFS- BSSSPD	Lack of authentication increase the malicious traffic in the network which misleads the access control.
[30]	To design blockchain entrenched access control techniques in cloud computing environment.	Encryption & Access control	Access control policies are randomly generated where inconsideration of their role and attributes leaks to high data leakage.
[31]	To develop blockchain based multi-authority access approach BMAC for secure data sharing.	S3 & BMAC	Only considering the attribute while providing access control affects the efficiency of access control.
[32]	To introduce the secure data securing mechanism by implementing access control in the cloud-based applications.	Fabric-ABAC & PRE	Anyhow the traditional blockchain suffers from scalability issues.

automatically examining permissions. Finally, intelligent contracts exploited Proxy Re-Encryption (PRE) to identify ciphertext communication without involving a third party. Table 2 outlines the limitations of access control mechanisms and secure data sharing.

2.3. *Analysis of authentication mechanism & secure data sharing*

A mechanism for generating enhanced secure keys was explicitly designed for encrypting data in a cloud environment [33]. To begin, security keys are generated using segments of an identity bit string to enable an enhanced identity-based encryption approach. This method ensures that the user's identity remains concealed, preventing any possible adversary or attacker from decoding the key or decrypting the data. The key benefit of this method is that it leverages a polynomial interpolation function consisting of a Lagrange coefficient to hide the user's identity. Additionally, the system's security depends on the computing complexity of the bilinear Diffie-Hellman problem. Ultimately, this mechanism efficiently performs the encryption and decryption processes, thereby reducing latency. Another secure authentication scheme based on blockchain was introduced in cloud computing [34]. Initially, all users were registered with the authentication server (AS) and obtained their secret key through AS using Harmony search optimization (HSO). The elliptic curve integrated encryption scheme (ECIES) was then utilized to encrypt the data packets in mobile nodes and transfer them to a cloud server.

The SDN controller oversees the blockchain to protect evidence gathered from the users' signatures and data, which are embedded in the cryptographic hash algorithm of the SHA-256. The authorized investigator then conducts various processes such as identification, evidence gathering, examination, and report generation using the Logical Graph of Evidence (LGoE). A searchable encryption technique has been included for authentication and authorization in cloud computing [35]. This work consists of three components: classic user authentication (based on username, password, and a message with a code sent via SMS), a searchable encryption scheme, and biometric authentication. The first two components comprise two-factor authentication (2FA), with the second component illustrating the initialization process of the searchable encryption technique. Special attention has been given to the trapdoor function, which generates a value that can be used to execute the search process and function. Table 3 outlines the limitations of the authentication mechanism and secure data sharing.

3. Problem statement

Network traffic analysis for anomaly detection in the integrated environments of cloud computing and DevOps was introduced [36]. Initially, the weight agnostic neural networks (WANNs) framework was designed to automate the detection of malicious intent through darknet traffic examination and network management. Then it was utilized as an intelligent forensics tool for analyzing network traffic, and clarification of malware traffic, and the traffic identification was encrypted in real-time. After that, features are extracted, and feature selection is performed using the predictive power score (PPS) method. Then, the automated searching neural-net scheme was implemented to detect zero-day attacks. Finally, the employed process of malicious intent detection, the most critical asset of many organizations, was protected effectively by reducing effort barriers. The major limitations of this proposed work are described below as follows,

- Here, all the users are considered legitimate users and developers, further permitting them to access the cloud application that leads to high complexity and communication overhead in both the network and application due to the presence of high number of illegitimate users.
- In this work, even though intrusion detection was performed to enhance the security in DevOps, permission for accessing the applications was provided to all the users and developers, leading to the high leakage of sensitive data.
- The features are extracted, and appropriate features are selected by feature selection using predictive power score which executes effectively; however, this method consumes a considerable time for performing calculations that leads to high latency.
- Here, even though the intrusion detection was implemented using WANNs to minimize the security threats in the DevOps, lack of effective and secure data storage affects the data privacy and leads to security breaches.

Efficient feature extraction is incorporated for intrusion detection in a cloud computing environment [37]. Initially, the set significant features is selected using a univariate ensemble feature selection mechanism. It utilizes five dissimilar filter feature selection

Table 3
Limitations of authentication & secure data sharing.

Reference	Objective	Methods/Algorithms	Limitations
[33]	To work, I proposed an enhanced secure key generation mechanism to encryption in cloud environment.	Lagrange coefficient & Bilinear Diffle-Hellman	Here, the insecure channel selection for critical transformation leads to greater risk of disclosure.
[34]	To propose blockchain based secure authentication scheme in cloud computing.	HSO, ECIES, SHA-256 & LGoE	Consideration of insufficient credentials for authentication and lack of ensuring developers and resource owners legitimacy increases security breaches.
[35]	To implement a searchable encryption technique for authentication and authorization in cloud computing.	SMA, 2FA, special attention & trapdoor function	2FA approach was utilized which enhance the security but this approach can turn against the users due to factors get losing that limit with QoS.

mechanisms for acquiring the subset of optimal features from the collected data. Then the feature map is generated from the set of filtered features for classification. Finally, the ensemble majority voting is performed using the ensemble of several machine learning algorithms which classify into two classes: intruder and normal. The major issues of the work are delineated,

- All the users in this work are considered legitimate candidates and granted access to the cloud-based applications without any limitations and permission policies that lead to high-security breaches.
- The feature extraction employed five different filter selection mechanisms, while an ensemble of machine learning algorithms was used for intrusion detection, leading to increased system complexity.
- Here, the intruder was detected using an ensemble learning algorithm, which improves security; the lack of countermeasures and other technical issues (i.e., bugs, network settings.., etc.) led to system and service unavailability in the cloud environment.

A hybrid optimized deep learning approach was developed to improve security in DevOps by detecting attacks. The approach consists of two phases: feature extraction and classification. Initially, network traffic is monitored, and data from individual applications are processed to extract features. These extracted features are then fed into a the classification model for attack detection. To execute the classification model, an optimized deep belief network (DBN) algorithm was proposed. Finally, the activation function was optimized using the hybrid optimization algorithm called Firefly Alpha-Evaluated Grey Wolf Optimization Algorithm (FAE-GWO). The limitations of this approach are further explained below,

- In this work, communication and data sharing were implemented without performing any privacy concerns that lead to the leakage of user's sensitive information and organizational policies.
- Here, the network traffic is monitored, data flows are collected, and statistical features are extracted for attack classifications. However, the consideration of limited features leads to ineffective attack detection.
- The attack detection was accomplished by an optimized deep belief network where the complexity of this algorithm was minimized by optimizing the activation method, however, the traditional drawback of this algorithm was its robust nature which is unsuitable for handling large data that leads to increases in high latency.
- Even though the attack detection was performed to enhance the security of DevOps effectively, anyhow, lack of consideration of the bugs, network settings, etc …, leads to service unavailability thereby limiting QoS.

Fast and continuous monitoring (F&CM) effectively improves system availability and security in DevOps, as stated in Ref. [39]. Initially, this mechanism was executed using the software and system process engineering metamodel (SPEM). The real-time scenario demonstrated that the execution of F&CM availability mechanism helps teams to detect and remedy outage problems and attacks better. By promptly detecting and identifying outage problems and attacks, teams can quickly and effectively apply the necessary remediation. However, some drawbacks of this research are mentioned below,

- Continuous monitoring was carried out in this study to monitor system outages and attacks, which was effective. However, improper monitoring and remediation failed to meet QoS requirements. DevOps enhanced system availability and security through fast and continuous monitoring. Nonetheless, the lack of secure data communication and information sharing resulted in high leakage of organizational policies and sensitive application information.
- Continuous monitoring was employed to detect attacks and outage issues in a cloud-based application. However, the packet features were not considered, leading to ineffective attack detection. The Software and System Process Engineering Metamodel (SPEM) was used for continuous monitoring, but no intelligence was adapted to detect outages and attacks, resulting in security issues.

A method of fine-grained access control using an attributed-based searchable encryption approach was proposed in Ref. [40]. The framework includes an attribute-entrenched searchable encryption scheme allowing precise access control. The data owner stores the access rules with a searchable encryption service provider (SESP). When a user requests access, the SESP returns the encrypted search results using the SHA algorithm within a specified timeframe. If the user has any disputes, they can initiate an arbitration request. The blockchain handles such requests but only arbitrates on entrenched details. The main issues addressed by this research are also defined. The Neural network approach as suggested by Liu et al. [41] leveraging mixed mode-dependent time-delays were also found interesting.

In a cloud environment, all users are considered legitimate. However, malicious users increase the amount of malicious traffic, which negatively affects security. Access control is provided using a searchable encryption scheme, which can be complicated to implement and may not be fully effective due to limited consideration for this attribute. Searchable encryption service providers (SESPs) preserve data and privacy during transmission to address this issue. While this method is effective, traditional blockchain technology may need to provide more confidentiality and scalability. In this work, SESP stores and provides user requests through encryption using the SHA algorithm, which is effective but can result in high latency due to its time consumption.

3.1. Research solutions

We have proposed an end-to-end security amplified framework to overcome the disputes. Initially, the users, developers, and resource owners are candidates authenticated by TCA using the Mcha-Poly 1305 algorithm based on their credentials. After that, the access control policies are effectively generated by EDDPG. Based on trust value evaluated by SMA, attribute, and role, the optimal user and developer are selected using AVOA, and access control is provided, thus minimizing high data sensitive leakage. Then, data privacy in transmission and rest is enhanced by performing encryption before transmission using the Mcha-Poly 1305 algorithm.

Moreover, based on the sensitive data, it is decided to store it in blockchain and cloud servers, improving the application's security.

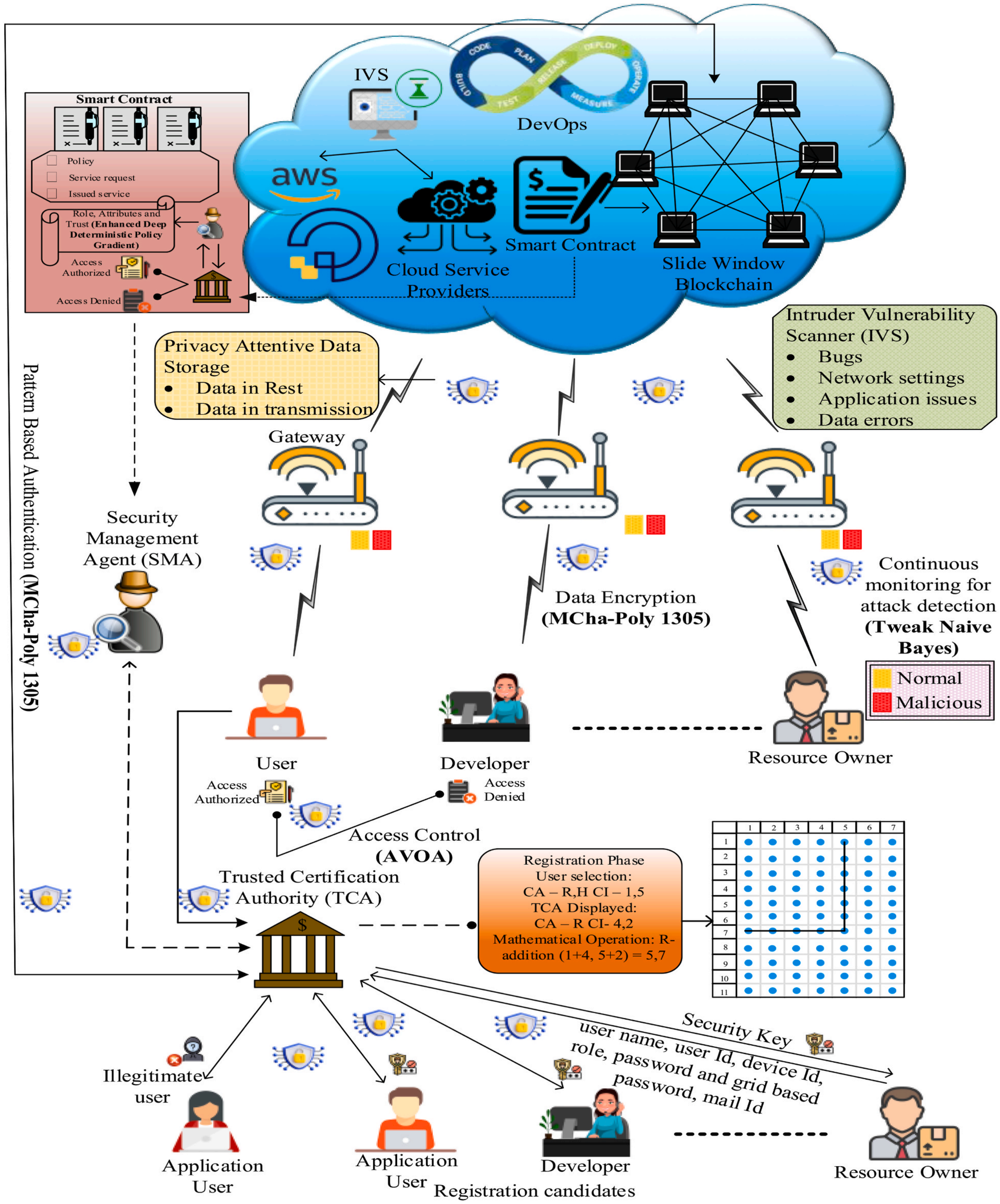


Fig. 1. Overall architecture of proposed slide-block framework.

System and service unavailability are the foremost issues; we have proposed vulnerability management and revision to address this. In this process, we have identified and mitigated both application-based issues and attacks using IVS and Tweak-NB algorithms, respectively, in terms of considering several parameters.

4. Proposed work

In this research, we concentrate on amplifying the security of cloud-based applications by integrating DevOps. In addition, data engineering, DevOps, and machine learning are combined to ensure the automation of the DevOps cycle in the production environment. The slide window blockchain is employed to increase data privacy and immutability. Fig. 1 represents the architectural flow of the proposed Slide-Block framework. The proposed work consists of several entities, which are elaborated on below,

- **Users:** Users in the physical layer (fundamental layer) are seeking cloud applications, performing data collection, and storing their data in a cloud server from several sensors. The users can access cloud applications from any location through laptops, mobile phones, computers, etc.
- **Developers & Resource Owners:** Developers in the network were responsible for developing the cloud-based application. The resource owners are the ones who own the cloud-based applications.
- **Trust Certification Authority (TCA):** Trust Certification Authority deployed in the physical layer is one of the blockchain nodes for accommodating authenticity to users by analysing their credentials and affording them with security keys.
- **Edge Server:** The edge layer encompasses several servers accountable for collecting the network traffic. Furthermore, it continuously monitors the network for attack detection to strengthen the security and privacy of users, developers, and resource owners.
- **Cloud Server:** Cloud layer composed of blockchain to upsurge network security and minimize computational burden by providing adequate access control. Moreover, it is responsible for accommodating servers for users.
- **Security Management Agent (SMA): A** Security Management Agent is deployed in the network to amplify network security. This agent monitors and maintains the candidate records in the blockchain based on the historical data of the candidate and is constantly monitoring. Furthermore, it also evaluates trust value for individual users while affording access control and transmitting the application-based issues if they occur.

4.1. Pattern-based authentication

Initially, the users (ε), developers (γ) and resource owners (δ) legitimacy are ensured through performing authentication. For that, the ε, γ and δ are known as regitration candidates they are register by providing their credentials such as user name (Ɲ), user Id (α), device Id ($\mathfrak{T}$), role ($\mathfrak{r}$), password (Ƿ), mail Id (Ɯ) and grid-based pattern selection (Ơ) to the Trust Certification Authority (TCA) which sends the candidate credentials to blockchain to improve network security. This scheme consists of two stages, where the registration and authentication phase are detailed below as follows,

4.1.1. Registration phase

- **Step 1**: At first, the ε is registered to TCA by providing their credentials (Ɲ), (α), ($\mathfrak{T}$), ($\mathfrak{r}$), (Ƿ), (Ɯ), and (Ơ) which can be formulated as,

$$\text{TCA}\leftarrow\text{Reg}\,\{(\text{Ɲ}),(\alpha),(\mathfrak{T}),(\mathfrak{r}),(\text{Ƿ}),(\text{Ɯ}),(\text{Ơ})\}$$

where, Reg $\{(\text{Ɲ}),(\alpha),(\mathfrak{T}),(\mathfrak{r}),(\text{Ƿ}),(\text{Ɯ}),(\text{Ơ})\}$ denotes the registration of ε with parameters (Ɲ), (α), ($\mathfrak{T}$), ($\mathfrak{r}$), (Ƿ), (Ɯ), and (Ơ) respectively.

- **Step 2**: Once, the ε is registered in this stage, the TCA display the registration Compute Alphabets (CA) where that ε selects two Compute Alphabets (CA) which comprises of hide mathematical operations.

$$TCA\leftarrow dis(\text{CA})$$

In that, the first alphabet denotes the addition, and the second alphabet defines subtraction that will only be acknowledgeable for registration ε.

- **Step 3:** Then the ε must select the computer integer (CI) among 0 to 5 where the CA and CI generates new password pattern each time.

$$\varepsilon(sel)\rightarrow CI_{1\leftrightarrow 5}$$

where *sel* defines the ε user selected CI within the range of 0–5 that is denoted as $CI_{1\leftrightarrow 5}$.

4.1.2. Authentication phase

- **Step 4:** In the second stage, that is the authentication phase the registered ε must enter the N, α and p that can be expressed as,

$$\varepsilon \leftarrow Ent(\mathsf{N}, \alpha, p)$$

- **Step 5:** After that, the authenticate pointer (AP) display single letter which is selected by the ε during registration phase and display two random numbers among 1–6.

$$\mathrm{AP} \leftarrow \mathrm{dis}(CA_1, CI_{1\leftrightarrow 6})$$

where CA_1 denotes the displayed single letter and $CI_{1\leftrightarrow 6}$ denotes the displayed random two numbers from range of $1 \leftrightarrow 6$.

- **Step 6:** Furthermore, the ε executes the mathematical operation which is hidden in the AP of CA displayed letter i.e., either addition or subtraction between the candidate selected CI and the digits displayed by TCA.
- **Step 7:** Then the ε obtains two numbers by performing the mathematical operation. After that, the TCA displays the 7 × 11 grid where the candidate must draw the pattern through obtained numbers (i.e., the first number is considered as column and second number as row).
- **Step 8:** During, this the behavioural features of the ε person such as finger velocity and stroke time features are also extracted for authenticating the person.

$$TCA \leftarrow ext(fin_{vel}, str_{time})$$

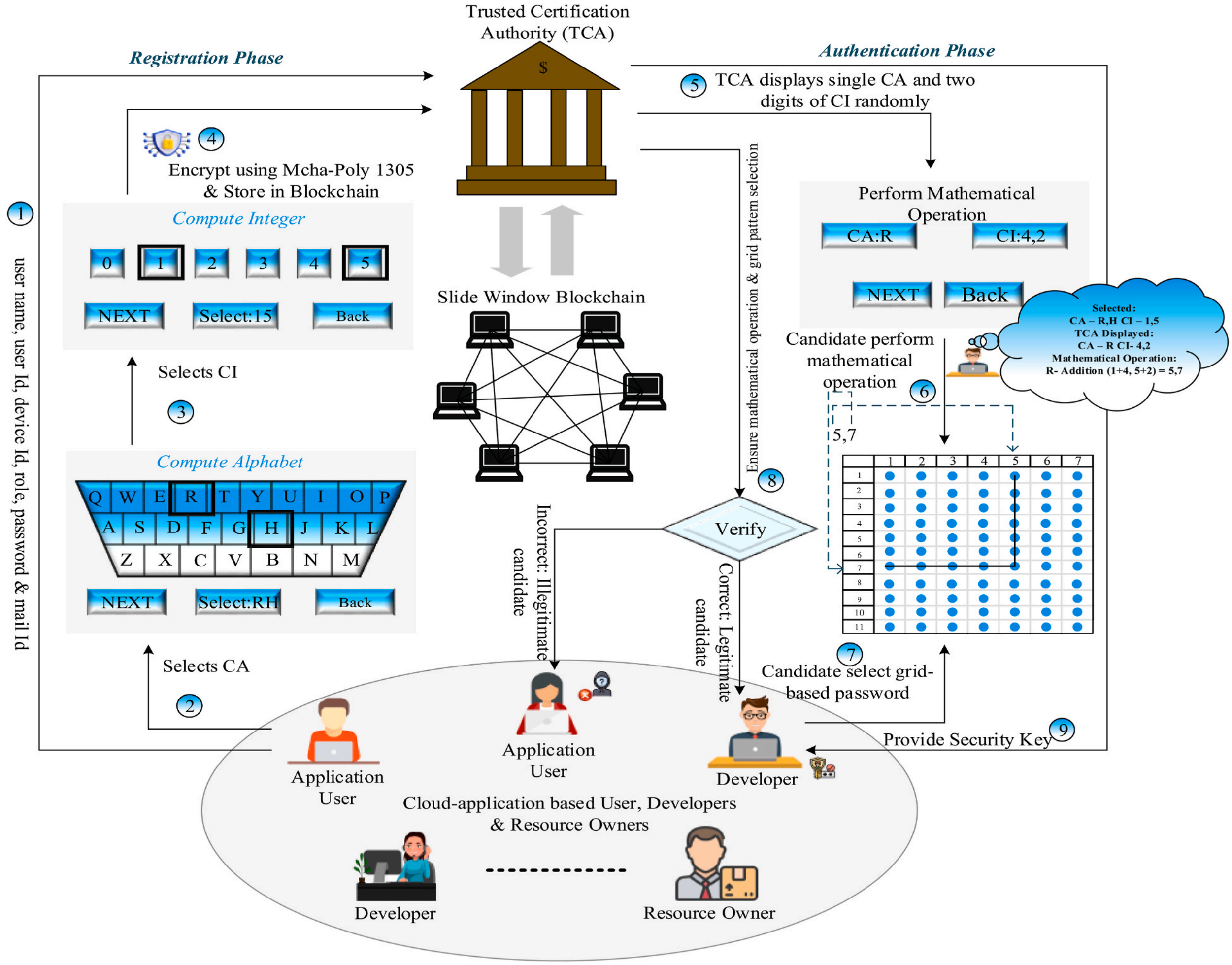

Trusted Certification Authority (TCA)
Registration Phase
Authentication Phase
5 TCA displays single CA and two digits of CI randomly
4 Encrypt using Mcha-Poly 1305 & Store in Blockchain
1
user name, user Id, device Id, role, password & mail Id
Compute Integer
0 1 2 3 4 5
NEXT Select:15 Back
Selects CI
3
Compute Alphabet
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
NEXT Select:RH Back
Selects CA
2
Slide Window Blockchain
Ensure mathematical operation & grid pattern selection
8
Perform Mathematical Operation
CA:R CI:4,2
NEXT Back
Selected:
CA – R,H CI – 1,5
TCA Displayed:
CA – R CI- 4,2
Mathematical Operation:
R- Addition (1+4, 5+2) = 5,7
Candidate perform mathematical operation
6
5,7
1 2 3 4 5 6 7
1 2 3 4 5 6 7 8 9 10 11
Verify
Incorrect: Illegitimate candidate
Correct: Legitimate candidate
7 Candidate select grid-based password
9 Provide Security Key
Application User
Application User
Developer
Cloud-application based User, Developers & Resource Owners
Developer
Resource Owner

Fig. 2. Workflow of pattern-based authentication.

where the parameters fin_{vel} and str_{time} refers to the finger velocity and stroke time features of ε.

- **Step 9:** Here, the password, ε selected CA and CI are encrypted by TCA, and it provides the security key to the legitimate ε, then the encrypted data is stored in blockchain using MCha-Poly 1305 algorithm.

$$TCA \leftarrow sk[(\mathsf{N}, \alpha, p, CA, CI)]$$

where sk represents security key affords to legitimate ε.

Likewise, the registration and authentication are performed for both γ and δ. The employed MCha-Poly 1305 [detailed in section C] algorithm improves the resistance to cryptanalysis with low complexity. By performing this pattern-based authentication, the shield against shoulder surfing attack and smudge attacks are resisted by TCA where the AP is displayed while the candidate touches the screen and disappears while taking their finger from the screen enhancing security. Fig. 2 illustrates the workflow of pattern-based authentication using the MCha-Poly 1305 algorithm.

4.2. Tri-level access control

After performing successful authentication, access control is implemented using the Enhanced Deep Deterministic Policy gradient (EDDPG) reinforcement algorithm where this algorithm effectively fabricates the access policies. Based on the actor-critic structure, the DDPG algorithm is developed with a dual deep neural network (DNN). To be more specific, critic network $\mathscr{Q}$ and policy network μ. Here, the policy network act as an actor to map the composition of state-space to continuous action $\forall$, while the value network act as critic, that timely estimates the performance of policy function and provides feedback for enhancement. Target networks $\mathscr{Q}^{'}$ and $\mu^{'}$ are utilized for tracking the original network of $\mathscr{Q}$ and μ, hence for mitigating the impact of incorrect estimation. The determination of DDPG action in certain timestamps t contemplatedboth inherent policy and exploration, which is mathematically expressed as,

$$\mathfrak{A}_t = \mu(\mathfrak{S}_t|\theta^{\mu}) + \omega$$

where $\mathfrak{S}_t$ refers to the state space, θ^{μ} is the parameter of μ and ω refers to the Gaussian noise that occurs only in training phase. Subsequently, the policy is evaluated in training phase; the ideologies of offline training are illuminated hereafter. Then, the policy estimation is executed by means of Bellman's principles as,

$$\mathscr{Q}^{*}(\mathfrak{S}_t, \mathfrak{A}_t) = \mathfrak{E}\left[\mathfrak{R}(\mathfrak{S}_t, \mathfrak{A}_t) + \Upsilon \, arg\max_{\mathfrak{A}_t}(\mathscr{Q}^{*}(\mathfrak{S}_{t+1}, \mathfrak{A}_{t+1}))\right]$$

where $\mathscr{Q}^{*}$ represents the function of optimal value, $\mathfrak{R}$ is the single-step reward and Υ known as discount factor. From the equation, it is clears that the optimal estimation of current composition of $\mathfrak{S}$ and $\mathfrak{A}$ can be acquired repeatedly. It is anticipated that deep networks $\mathscr{Q}$ and $\mathscr{Q}^{'}$ might indefinitely repetition task precisely. To recognize it, the updating error of critic network $\mathscr{Q}$ can be estimated by.

$$\mathfrak{L}_3\left(t|\theta^3\right) = \left[\left(\mathfrak{R}(\mathfrak{S}_t, \mathfrak{A}_t) + \Upsilon \mathscr{Q}^{'}\left(\mathfrak{S}_{t+1}, \mathfrak{A}_{t+1}\middle|\theta^{3^{'}}\right) - \mathscr{Q}\left(\mathfrak{S}_t, \mathfrak{A}_t\middle|\theta^3\right)\right)\right]$$

$$\mathfrak{A}_t = \mu^{'}\left(\mathfrak{S}_t\middle|\theta^{3^{'}}\right)$$

where the first two terms in (9) represents the anticipated $\mathscr{Q}$ value denoting to (10), and the last term represents the actual output of current critic network. By this way, the squared error is acquired, and the updating method of gradient-descent is performed for enhancing the ability of policy evaluation. An ideal critic network is anticipated to generate effective policy, henceforth that the actor network can modify its policies corresponding to abandon the action with worst $\mathscr{Q}$ value feedback. Thus, the performance of objective policy network can be represented as $\nexists$,

$$\nexists\left(\theta_{\mu}\right) = \mathfrak{E}[-\mathscr{Q}(\mathfrak{S}_t, \mu(\mathfrak{S}_t))]$$

where $\mathfrak{E}(\bullet)$ defines the expectation operator. Next, the policy network repeats updating autonomously towards the promoting direction of performance objective. Consequently, the updating error expressed as objective gradient regarding network μ can be expressed as,

$$\mathfrak{L}_{\mu}(t|\theta^{\mu}) = \nabla_{\theta_{\mu}} \nexists(\theta^{\mu}) = \nabla_{\mathfrak{A}} \mathscr{Q}\left(\mathfrak{S}_i, \mu(\mathfrak{S}_i)\middle|\theta^{\mathscr{Q}}\right) \nabla_{\theta_{\mu}} \mu(\mathfrak{S}_t|\theta^{\mu})$$

A strategy of soft updating is employed for target networks $\mathscr{Q}^{'}$ and $\mu^{'}$ can expressed as,

$$\theta^{\mathscr{Q}} \leftarrow \tau\theta^{\mathscr{Q}} + (1-\tau)\theta^{\mathscr{Q}}$$

$$\theta^{\mu} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu}$$

Here, the method of experience method is exploited for evade the back-forth correlations while training which enhance the stability and efficiency of learning. The probability of appraised experience j can be defined as,

$$Pro_j = D_j^{\partial} \Big/ \left(\sum\nolimits_k D_k^{\partial} \right)$$

$$D_j = {}^{1}/_{rank(j)}$$

where $\sum_k(\bullet)$ is the total index of experience pool and ∂ denotes the hyperparameter to compute degree of priority which ranges from 0 to 1. Lower ∂ leads to uniform conventional sampling DDPG, $rank(\bullet)$ is the prominence degree of set experience that can be estimated by,

$$rank(j) = \sqrt{\mathfrak{L}_{\mathscr{Q}}(i)}$$

By exploiting the replay experience, those experiences initiating huge important variations to policy estimations will be allocated huge weights, and therefore, are mostly selected and replay in training process. Once, the policies are generated, the access policies are provided steps involved are articulated as follows. Initially, the legitimate user (users and developers) initiates the request to TCA for accessing the cloud based application. Here, the TCA is responsible for generating the access policies and providing the access control based on their attribute role and trust value using smart contract. Once the TCA received the request, it redirects the request to the security Management agent (SMA) where this agent monitoring and maintain the candidate records in the blockchain based on the historical data of the candidate, historical access behaviour and constant monitoring which can be represented as,

$Can_{rec}(d,a,m) = \{Can_{rec_1}, Can_{rec_2}, \dots, Can_{rec_n}\}$ Furthermore, it will calculate the trust value based on the information stored in blockchain which can be evaluated as,

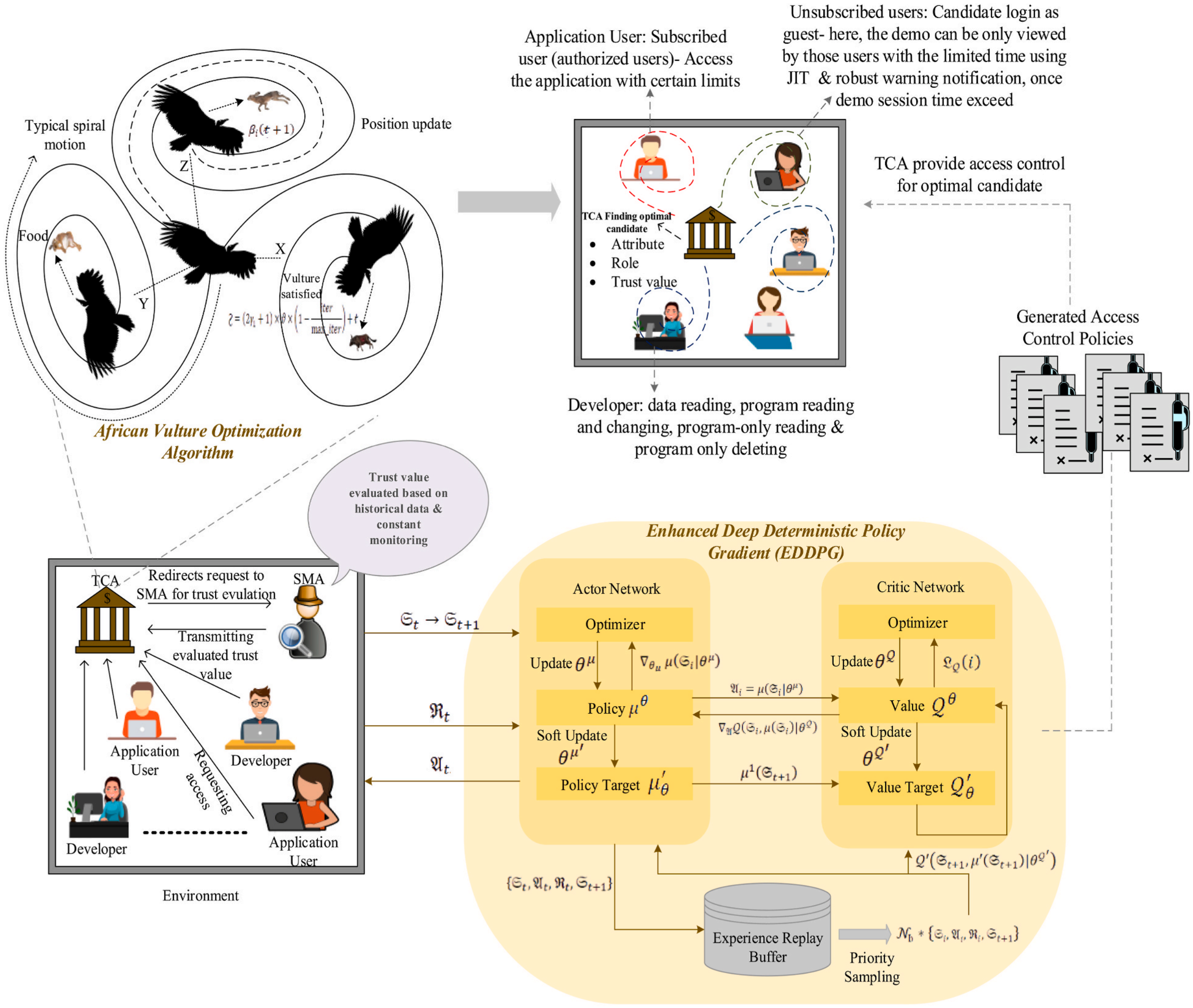

Typical spiral motion
Position update
$\beta_i(t+1)$
Z
Food
X
Y
Vulture satisfied
$\zeta = (2\eta_1 + 1) \times \vartheta \times \left(1 - \frac{iter}{max\_iter}\right) + t$
African Vulture Optimization Algorithm
Application User: Subscribed user (authorized users)- Access the application with certain limits
Unsubscribed users: Candidate login as guest- here, the demo can be only viewed by those users with the limited time using JIT & robust warning notification, once demo session time exceed
TCA Finding optimal candidate
Attribute
Role
Trust value
TCA provide access control for optimal candidate
Developer: data reading, program reading and changing, program-only reading & program only deleting
Generated Access Control Policies
Trust value evaluated based on historical data & constant monitoring
TCA
Redirects request to SMA for trust evulation
SMA
Transmitting evaluated trust value
Application User
Requesting access
Developer
Developer
Application User
Environment
$\mathfrak{S}_t \rightarrow \mathfrak{S}_{t+1}$
$\mathfrak{R}_t$
$\mathfrak{A}_t$
Enhanced Deep Deterministic Policy Gradient (EDDPG)
Actor Network
Optimizer
Update θ^{μ}
$\nabla_{\theta_u} \mu(\mathfrak{S}_i|\theta^{\mu})$
Policy μ^{θ}
Soft Update $\theta^{\mu'}$
Policy Target μ'_{θ}
$\mathfrak{A}_i = \mu(\mathfrak{S}_i|\theta^{\mu})$
$\nabla_{\mathfrak{A}} Q(\mathfrak{S}_i, \mu(\mathfrak{S}_i)|\theta^{Q})$
$\mu^1(\mathfrak{S}_{t+1})$
Critic Network
Optimizer
Update θ^{Q}
$\mathfrak{L}_Q(i)$
Value Q^{θ}
Soft Update $\theta^{Q'}$
Value Target Q'_{θ}
$Q'(\mathfrak{S}_{t+1}, \mu'(\mathfrak{S}_{t+1})|\theta^{Q'})$
$\{\mathfrak{S}_t, \mathfrak{A}_t, \mathfrak{R}_t, \mathfrak{S}_{t+1}\}$
Experience Replay Buffer
$\mathcal{N}_b * \{\mathfrak{S}_i, \mathfrak{A}_i, \mathfrak{R}_i, \mathfrak{S}_{t+1}\}$
Priority Sampling

Fig. 3. Tri-level access contract.

$$Tru_{can}^{\mathfrak{t}} = \sum_{Can_{rec}} \mathscr{P}(Can_{rec_i}) \times \mathfrak{F}(d, a, Can_{rec_i})$$

where $\mathfrak{t}$ is the current time period and $\mathfrak{F}(d, a, Can_{rec_i})$ signifies the trust estimation of candidate records Can_{rec_i}, in which, $\mathscr{P}(Can_{rec})$ can be expressed as,

$$\mathscr{P}(Can_{rec}) = \mathfrak{p}(Can_{rec_i}) \Big/ \sum\nolimits_{\mathfrak{p}} (Can_{rec_j})$$

where the equation defines the proportion of candidate records set Can_{rec}, and $\mathfrak{p}(Can_{rec_i})$ is decrease with time. Here, the $\mathscr{P}$ denotes the permission of an entity collection of degree and scope of candidate operations on resources which includes of writing, deleting, data reading etc. Then, the SMA sends the candidate's trust value and historical data to TCA. Based on the attribute, role and trust value of the candidate, the TCA assigns access control for the candidate through resource owner. The resource owner will sign the token to prove that the resource is issued by the owner. Fig. 3 demonstrates the tri-level access control based on EDDPG and AVOA. For providing access control optimally, the African Vulture Optimization Algorithm (AVOA) is employed which is illustrated below.

4.2.1. *Preliminary stage*

At first, the primary members are moulded, and then the appositeness of entire solution is determined. Here, there are two kinds of candidate solutions, individual group's optimal solution is determined. In our work, the TCA is act as a vulture which search for optimal candidate.

4.2.2. *Starvation rate*

For providing access control, vulture keeps searching for food (i.e., optimal candidate) which become violent while starvation that can be represented as,

$$\mathscr{t} = h \times \left(sin^{\mathscr{w}}\left(\frac{\pi}{2} \times \frac{iter}{\max\_iter}\right) + cos\left(\frac{\pi}{2} \times \frac{iter}{\max\_iter}\right) - 1 \right)$$

$$\text{Ƨ} = (2\gamma_1 + 1) \times \vartheta \times \left(1 - \frac{iter}{\max\_iter}\right) + \mathscr{t}$$

where Ƨ defines the eagles satisfied status, the parameters *iter* and $\max\_iter$ is the present and maximum iterations respectively, ϑ is the random number which ranges [-1,1], h is random number which ranges [-2,2] and γ_1 is also an random number among 0 and 1. The eagle is predicted as starved, if the value of ϑ is minimal than 0, and the eagle is satisfied if the value gets increased. Here, the $\mathscr{w}$ is fixed number that represents the exploration and exploitation phase in AVOA.

4.2.3. *Exploration phase*

In this stage, the eagle scan diverse area arbitrarily that can be attained by two dissimilar strategies. In AVOA, β_1 is the parameter that is adopted to choose any one of those two strategies, the β_1 value is among 0 and 1, the strategy is chosen utilizing following equation,

$$\beta_1(\mathscr{t}+1) = \begin{cases} \mathscr{Z}(i) - |\mathfrak{X} - \mathscr{Z}(i) - \beta(i)| \times F\beta_1 \geq \gamma_{\beta_1} \\ \mathscr{Z}(i) - F + \gamma_2 \times ((\mathfrak{u}_{\mathfrak{b}} - \mathfrak{l}_{\mathfrak{b}}) \times \gamma_3 + \mathfrak{l}_{\mathfrak{b}})\beta_1 < \gamma_{\beta_1} \end{cases}$$

where $\mathscr{Z}(i)$ designates one of the finest eagles, $\mathfrak{X}$ is the distance that the eagle moves to shield food from others γ_2 and γ_3 which is an random number [0,1], F is the fitness value and the parameters $\mathfrak{u}_{\mathfrak{b}}$ and $\mathfrak{l}_{\mathfrak{b}}$ designates the upper and lower bounds in search space respectively.

4.2.4. *Exploitation stage*

In this stage, the eagle has adequate energy for searching the food. At such times, the vulture with extreme physical strengthens and it generates rotational flight to typical spiral motion. The exploitation is the first stage which can be formulated as follows,

$$\beta_i(\mathscr{t}+1) = \begin{cases} |\mathfrak{X} - \mathscr{Z}(i) - \beta(i)|(F + \gamma_4) - (\mathscr{Z}(i) - \beta(i))\beta_2 \geq \gamma_{\beta_2} \\ \mathscr{Z}(i) - \mathscr{Z}(i) \times \left(\frac{\beta(i)}{2\pi}\right)(\gamma_5 \times \cos(\beta(i)) + \gamma_6 \times \sin(\beta(i)))\beta_2 < \gamma_{\beta_2} \end{cases}$$

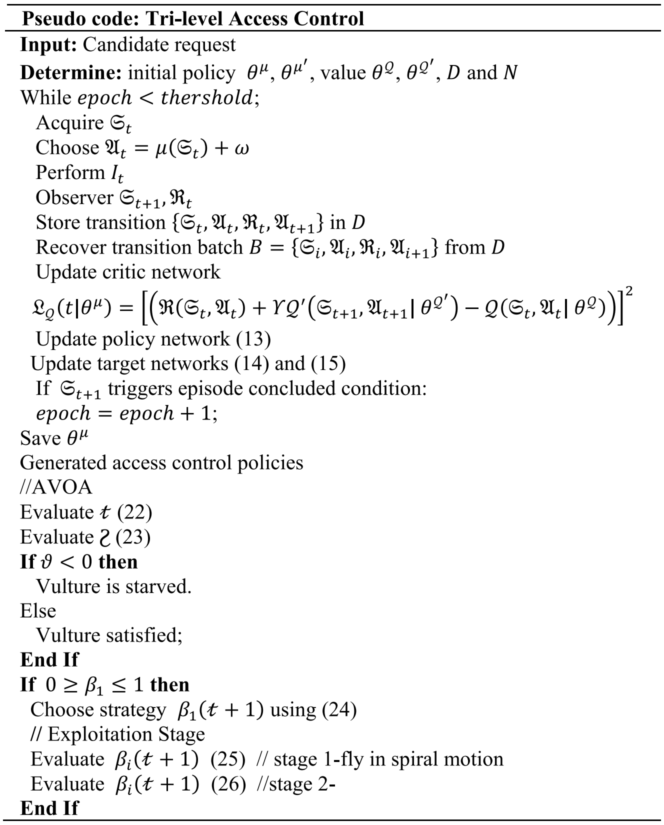
where $\gamma_4,$ γ_5 and γ_6 denotes the random numbers [0,1]. In second stage, the movements of eagle fascinate various vultures. The second stage of exploitation phase is mathematically formulated as,

$$\beta_i(t+1) = \begin{cases} 0.5\left(Best_{Eagle} - \left(\frac{Best_{Eagle}(i) \times \beta(i)}{Best_{Eagle}(i) \times \beta(i)^2}\right) \times F\right) \\ \mathscr{Z}(i) - \Big|\mathscr{Z}(i) - \beta(i)| \times F \times Levy(\mathscr{Z}(i) - \beta(i))\beta_3 < \gamma_{\beta_3} \end{cases}$$

where *Levy* denotes the function of levy flight. After performing, the AVOV for optimal candidate selection, if the candidate is not suitable for certain access or with low trust value, then the TCA terminates the request and send the notification to the specific candidate. Here, the permission level that is the degree of operations on resource is imitated with three levels as Operator (developer), subscribed and unsubscribed. Those permission levels are determined as,

- Operator: The developer who developed the program coding and implementation-data reading, program reading and changing, program-only reading, the program only deleting.
- Subscribed user (authorized users): Access the application with certain limits.
- Unsubscribed users: The candidate who is login as guest-here, the demo or basic instructions for those specific applications can be only viewed by those users with the limited time using Just-in-time (JIT) mechanism. Once the time is completed, the robust warning notification will be displayed, and their demo session will be concluded.

The access control policies are encrypted and stored in the blockchain for tamper-proof. By this, way the access control policies enhance security thereby reducing highly sensitive information linkage.



```
Pseudo code: Tri-level Access Control
Input: Candidate request
Determine: initial policy $\theta^{\mu}$, $\theta^{\mu'}$, value $\theta^{\mathcal{Q}}$, $\theta^{\mathcal{Q}'}$, $D$ and $N$
While $epoch < thershold$;
  Acquire $\mathfrak{S}_t$
  Choose $\mathfrak{A}_t = \mu(\mathfrak{S}_t) + \omega$
  Perform $I_t$
  Observer $\mathfrak{S}_{t+1}, \mathfrak{R}_t$
  Store transition $\{\mathfrak{S}_t, \mathfrak{A}_t, \mathfrak{R}_t, \mathfrak{A}_{t+1}\}$ in $D$
  Recover transition batch $B = \{\mathfrak{S}_i, \mathfrak{A}_i, \mathfrak{R}_i, \mathfrak{A}_{i+1}\}$ from $D$
  Update critic network
  $\mathfrak{L}_{\mathcal{Q}}(t|\theta^{\mu}) = \left[\left(\mathfrak{R}(\mathfrak{S}_t, \mathfrak{A}_t) + \Upsilon\mathcal{Q}'(\mathfrak{S}_{t+1}, \mathfrak{A}_{t+1}|\, \theta^{\mathcal{Q}'}) - \mathcal{Q}(\mathfrak{S}_t, \mathfrak{A}_t|\, \theta^{\mathcal{Q}})\right)\right]^2$
  Update policy network (13)
  Update target networks (14) and (15)
  If $\mathfrak{S}_{t+1}$ triggers episode concluded condition:
  $epoch = epoch + 1$;
Save $\theta^{\mu}$
Generated access control policies
//AVOA
Evaluate $t$ (22)
Evaluate $\mathfrak{z}$ (23)
If $\vartheta < 0$ then
  Vulture is starved.
Else
  Vulture satisfied;
End If
If $0 \geq \beta_1 \leq 1$ then
  Choose strategy $\beta_1(t+1)$ using (24)
  // Exploitation Stage
  Evaluate $\beta_i(t+1)$ (25) // stage 1-fly in spiral motion
  Evaluate $\beta_i(t+1)$ (26) //stage 2-
End If
```

4.3. Privacy attentive data storage

After accommodating the access control, privacy-attentive data storage and communication is executed for evading sensitive data leakage. Here, the two types of the data are encrypted and stored in blockchain. The data in transmission ($trans_{data}$) and the data in rest ($rest_{data}$) are the two categories. Initially, the $trans_{data}$ that is the data transmitted between the cloud, developers, users, and resource owners are encrypted. Then, the $rest_{data}$, which is whenever the candidate must store their data in cloud, the query is raised to whether the data ($Ɯ_{data}$) is sensitive or non-sensitive. The sensitive data are password, hard code passwords, rules and regulations, organizational policies and access control policies and are stored in the blockchain respectively. Moreover, the sensitive and non-sensitive data are evaluated based on certain threshold which can be determined as follows,

$$rest_{data} = \begin{cases} if \quad 0 > \text{Ш}_{data} \leq 0.5 \quad non - sensitive \\ \quad if \ 0.5 > \text{Ш}_{data} \leq 1 \quad sensitive \end{cases}$$

If the $rest_{data}$ is sensitive then the $rest_{data}$ is encrypted and stored in blockchain which can be illustrated as,

$$Enc(rest_{data}) \rightarrow \text{ЉC}$$

where ЉC denotes the blockchain. If the If the $rest_{data}$ is non-sensitive it is encrypted and stored in cloud server which can be exemplified as,

$$Enc(rest_{data}) \rightarrow cloud_{server}$$

For that encryption purpose, we have adapted the Magnificent Chacha algorithm where this algorithm encrypts the data effectively thereby incorporating poly 1305 algorithm. Henceforth, the proposed algorithm is known as Magnificent Chacha-poly 1305 algorithm (MCha-Poly 1305) which is employed for its randomization characteristics and rotation technique to secure the data that execute on low duty cycle meanwhile poly 1305 is employed for its confidentiality and integrity. Both the algorithms are tangled to effectively encrypt the data. MCha-Poly 1305 considers input Ш_{data}, nonce of 12-byte (Ȣn) which is depicted as,

$$rest_{data} : \{0,1\}^{256} \rightarrow \{0,1\}^{*} \times \{0,1\}^{128}$$

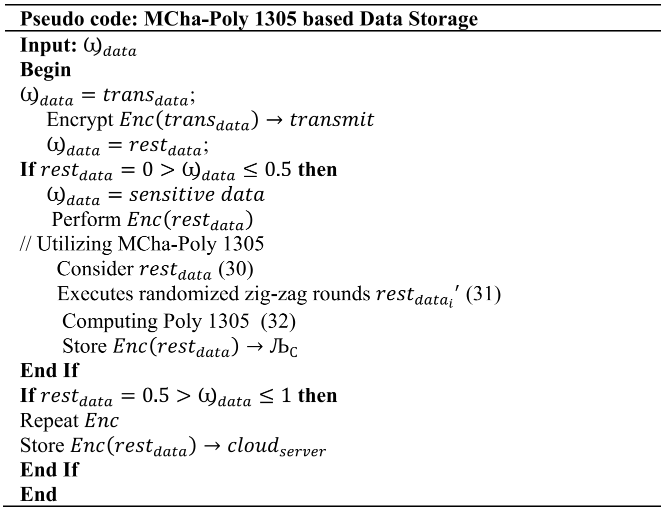
where, $\text{Ȣn} \rightarrow \{0,1\}^{256}$ and $\text{Ш}_{data} \rightarrow \{0,1\}^{128}$ which are tuple of $(\text{Ȣn}, \text{Ш}_{data})$ 256-bit. The tuple is accommodating as an input to MCha-Poly 1305 for computing the $Enc(rest_{data_i}{}')$. With 256-bit tuple, the MCha-Poly 1305 executes randomized zig-zag rounds $(i.e. \geq 10)$ that can be mathematically formulated as,

$$rest_{data_i}{}' = \{\text{MCha} - Q^{fun}(\text{ŋ}0, \text{ŋ}4, \text{ŋ}12, \text{ŋ}7), \text{MCha} - Q^{fun}(\text{ŋ}9, \text{ŋ}1, \text{ŋ}10, \text{ŋ}2), \text{MCha} - Q^{fun}(\text{ŋ}6, \text{ŋ}13, \text{ŋ}5, \text{ŋ}3), \ldots, \text{MCha} - Q^{fun}(\text{ŋ}8, \text{ŋ}12, \text{ŋ}11, \text{ŋ}15)\}$$

From the equation, MCha is the Magnificent Chacha algorithm and Q^{fun} is denoted as quartier function that implements randomized zig-zag approaches to update the input for individual rounds. By performing the update of randomized zig-zag approaches, security is strengthened where the attackers faces high complexity to tamper the data. Generated $rest_{data_i}{}'$ is accommodate as input to ploy 1305 for acquiring the ЉC. The poly 1305 is computing ЉC based on sk_i' and the polynomial co-efficient $(\text{ŋ}_i)1 \leq i \leq \mathfrak{m}$ at $\mathscr{Y}$ that can be expressed as,

$$\text{ЉC} = \left(\sum_{i=1}^{\mathfrak{m}} \text{ŋ}_i \mathscr{Y}^{\mathfrak{m}-i+1} \ mod \ 2^{130} - 5 \right) + sk_i' \ mod \ 2^{128}$$

By this way of encryption, data privacy is enhanced thereby reducing sensitive data leakage. Furthermore, the cloud-based application is secured, and the attackers can't tamper the policies and candidate's information.



```
Pseudo code: MCha-Poly 1305 based Data Storage
Input: $ω_{data}$
Begin
$ω_{data} = trans_{data}$;
    Encrypt $Enc(trans_{data}) \rightarrow transmit$
    $ω_{data} = rest_{data}$;
If $rest_{data} = 0 > ω_{data} \leq 0.5$ then
    $ω_{data} = sensitive\ data$
    Perform $Enc(rest_{data})$
// Utilizing MCha-Poly 1305
     Consider $rest_{data}$ (30)
     Executes randomized zig-zag rounds $rest_{data_i}'$ (31)
     Computing Poly 1305 (32)
     Store $Enc(rest_{data}) \rightarrow$ ЉC
End If
If $rest_{data} = 0.5 > ω_{data} \leq 1$ then
Repeat $Enc$
Store $Enc(rest_{data}) \rightarrow cloud_{server}$
End If
End
```

4.4. Vulnerability management and emendation

Once, the data is securely stored and data transmission is encrypted successfully, vulnerability management and emendation are performed. In DevOps, continuous integration, continuous delivery, and continuous monitoring are the vital processes to enhance the QoS and security for cloud based application integrated DevOps. To improve the security, continuous monitoring is established. In this process, both the application-based issues and attack is detected and rectified. Here, the application-based issues such as bugs, application issues, network settings, data errors are detected using IP configuration, user privileges, security protocols, file system infrastructure and patch levels through the Intruder Vulnerability Scanner (IVS) constantly. Furthermore, the packet features and behavioural features are continuously collected from the gateway and monitored for detecting the attackers using Tweak Naive Bayes (Tweak-NB) Algorithm where this algorithm significantly analyses the packet features and detect the attackers accurately.

Here, the linear relationships among the attributes are eradicated by an orthogonal matrix to minimize their relations thereby enhancing the algorithm performance. Assume $\mathfrak{M}_c = \{\mathscr{v}_1, \mathscr{v}_2, \ldots, \mathscr{v}_m\}$ defines the set of entire samples associate to class c in network traffic $\mathfrak{M}$, the samples $\mathscr{v}_t = \{\mathscr{v}_{t1}, \mathscr{v}_{t2}, \ldots, \mathscr{v}_{tn}\}$ in $\mathfrak{M}_c$ is n-dimensional vector. The covariance matrix is determined as,

$$Matx_c = \frac{1}{m}\sum_{k=1}^{m}(\mathscr{v}_t - \Phi) - (\mathscr{v}_t - \Phi)^{\mathscr{T}}$$

$$\Phi = (\Phi_1, \Phi_2, \ldots, \Phi_n)^{\mathscr{T}}$$

$$\Phi_{\ell} = \frac{\mathscr{v}_{1\ell} + \mathscr{v}_{2\ell} + \ldots + \mathscr{v}_{m\ell}}{m}, \ell = 1, 2, \ldots, n$$

where Φ_{ℓ} is denoted as the mean in ℓ th attribute value in individual class, m is the number of samples presented in $\mathfrak{M}_c$ and $Matx_c$ is denoted as covariance matrix. $Matx_c$ is an $n \times n$ matrix, assume eigenvalues in $Matx_c$ be $\text{Ч}_1, \text{Ч}_2, \ldots, \text{Ч}_n.$, and the eigenvectors as $\varrho_1, \varrho_2, \ldots, \varrho_n$. Each eigenvectors is combined by equation (36) to acquire the basis of standard orthogonal and to fabricate the orthogonal matrix $\mathscr{O}_c$.

$$\mathscr{O}_c = (\text{ф}_1, \text{ф}_2, \ldots, \text{ф}_n)$$

From that, $\text{ф}_i = \frac{1}{|\varrho_i|}\varrho_i, i = 1, 2, \ldots, n$, where $\text{ф}_i = 1, 2, \ldots, n$ is the combined eigenvector. Following covariance, matrix $Matx_c$ is harmoniously diagonalized through orthogonal matrix $\mathscr{O}_c$, entire elements are excepting the diagonal are 0's. Specifically, the linear relationships among the attributes are eradicated, which is nearer to the conditional independence assumption of naïve bayes (NB). The m samples in $\mathfrak{M}_c$ are transmuted by $\mathscr{O}_c$, then the mean and variance of individual attribute are evaluated. After executing the abovementioned transformation, the mean is 0 and variances modifies from $\sigma_{c,i}$ to $\sigma'_{c,i}$. By means of this basis, the weight of attribute $\text{Ծ}_{c,i}$ is enabled to optimize NB, and Tweak-NB is formulated as,

$$\eta_{inb}(\mathscr{v}) = argmax_{c \in \mathscr{C}}\widehat{\mathscr{O}}(c)\prod_{i=1}^{n}\text{Ծ}_{c,i}\widehat{\mathscr{O}}(\mathfrak{y}_i|c)$$

where $\mathfrak{y} = \mathscr{O}_c^{\mathscr{T}}(\mathscr{v} - \Phi_{c,i}), i = 1, 2, \ldots, n$ is denoted as the new sample acquired by transformation of $\mathscr{O}_c$ of sample $\mathscr{v}$.

$$\mathfrak{y} = (\mathfrak{y}_1, \mathfrak{y}_2, \ldots, \mathfrak{y}_n)$$

$$\widehat{\mathscr{O}}(\mathfrak{y}_i|c) = \frac{1}{\sqrt{2\pi}\sigma'_{c,i}}exp\left(-\frac{\mathfrak{y}_1^2}{\sigma'^2_{c,i}}\right)$$

$$\text{Ծ}_{c,i} = \left[1 + exp\left(-0.3 \times \frac{\prod_{c' \in C}\mathscr{g}(\Phi_{c,i}, \Phi_{c',i})}{\sigma_{c,i}}\right)\right]^{-1}$$

$$\mathscr{g}(\Phi_{c,i}, \Phi_{c',i}) = \begin{cases} |\Phi_{c,i} - \Phi_{c',i}|, \mathfrak{M}_c = normal; \\ 1, \mathfrak{M}_c = attack \end{cases}$$

Here, the parameters c and c' are denoted as class labels, $\Phi_{c,i}$ and $\Phi_{c',i}$ are represented as the means in i th attribute value in classes c and c'. Furthermore, the $\sigma_{c,i}$ is the variance in i th attribute value in classes c and c' respectively. Among these, the attribute variance $\sigma_{c,i}$ can illustrates as network traffic concentration, imitating the classification pre-eminence of attributes under the class when minimizing the noise interference. The product of absolute value in mean difference among different classes of same attribute, specifically, $\mathscr{g}(\Phi_{c,i}, \Phi_{c',i})$, enhances the classification performance. Once the application-based issues are detected, the SMA agent notifies to specific developer to amend the issues and it will block the attackers. Furthermore, feedback from the users, and developers are collected to enhance the continuous delivery. By executing this, both the system and service unavailability and the security are amplified.

```
Pseudo code: Tweak-NB based Attack Detection
Input: $\mathfrak{M}_c = \{\mathscr{v}_1, \mathscr{v}_2, \dots, \mathscr{v}_m\}$
Begin
Compute $\mathscr{v}_t = \{\mathscr{v}_{t1}, \mathscr{v}_{t2}, \dots, \mathscr{v}_{tn}\}$
Calculate (33) &(34);
Calculate $\Phi_\ell$ (35);
Fabricate $\mathcal{O}_c$ (36);
Enable $\mho_{c,i}$ to optimize NB
Evaluate Tweak-NB $\eta_{inb}(\mathscr{v})$ (37);
Obtain $\mathfrak{y}$ of $\mathcal{O}_c$;
Evaluate $\hat{\mathcal{O}}(\mathfrak{y}_i|c)$ (39)
Classify $q(\Phi_{c,i}, \Phi_{c',i})$ (40)
End
```

5. Experimental results

In this section, we have illustrated experimental result of the proposed Slide-Block framework using artificial intelligence (AI) approach and blockchain technologies. This experimental research enfolds of three categories including implementation setup, comparison analysis and research summary. The result section describes that the proposed work attains betterment performance with compared to existing models.

5.1. Implementation setup

In implementation setup, the experimental setup of proposed Slide-Block framework is demonstrated. The server utilized for proposed Slide-Block framework is Wamp server 2.0 and MySQL 5.1.36 as backend. Furthermore, we utilize OS windows 10 and the programming language of Java is adopted with development tool kit of JDK 1.8. We proposed conducted our work on Integrated Development Kit (IDE) by means of NetBeans 8.2. The above Software/Hardware requirements are installed in PC with Central Processing Unit (CPU) of Intel (R) Core (TM) i5-4590S CPU @ 3.00 GHz 3.00 GHz for the proposed Slide-Block framework.

5.2. Comparative analysis

In this section, several metrics are considered from proposed Slide-Block framework is compared with existing methods. Here, the following are the various performance metrics which we have evaluated such as detection rate, authentication time, packet loss rate, security strengthen, latency and communication overhead. The proposed Slide-Block framework is compared with several existing works such as Darknet [36], HOD-Net [38] and SPEM [39] to prove its efficacy of proposed work.

5.2.1. Result of detection rate

Detection rate (Dec_{rate}) is the metric which utilized to evaluate the rate of attack detection in network. Generally, this is characterized as number of detected attacks in ratio to the number of candidates increasing that can be described as,

$$Dec_{rate} = \frac{dec_{att}}{\aleph}$$

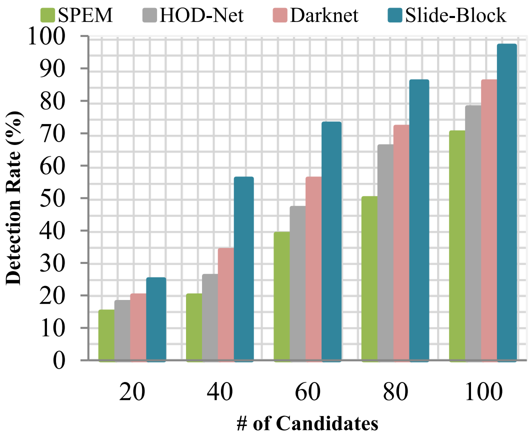


Fig. 4. # of Candidates vs Detection Rate.

where, dec_{att} designates the detected attacks and $\aleph$ designates the increasing candidates. Fig. 4 the graphical plot shows the comparison of proposed work and existing models in terms of detection rate with respect to number of candidates. Among that, it is clearly shown that the proposed work have achieved higher detection rate then the Darknet, HOD-Net and SPEM existing works. The main reason for achieving such high detection rate is due to executing vulnerability management and emendation process. In this, we have focused on both application-based issues and attacks detection through continuous monitoring. Data errors, application issues, bugs and network settings are some of the application-based issues which are detected using IVS based on user privileges, patch levels, file system infrastructure etc. Furthermore, the packet and behavioural features are considered for detecting the attacks using Tweak NB algorithm which provides effective and accurately results thus aids to increases in high detection rate. Meanwhile, in existing works lack of considering application side-based issues and manually monitoring leads to ineffective. Besides, while attack detection the adequate features are considered which are leads to low detection rate.

The numerical result shows that the Dec_{rate} of the proposed work is increased to 97% when the number of candidates increased to 100. In contrast, for the same number of candidates, the Dec_{rate} of the existing works Darknet, HOD-Net and SPEM achieves 86%, 78% and 70.3% of Dec_{rate} respectively. Overall, the Dec_{rate} of the proposed Slide-Block framework increases about 11%–27.3% than the existing works.

5.2.2. Result of authentication time

Authentication time (Au_{time}) is the vital metric utilized to estimate the amount of time occupied for implementing authentication of users, developers, and resource owners. Au_{time} is illustrated as the difference among the amount of time occupied for accommodating access to the individual request of the equivalent user to total amount of time. Au_{time} can be mathematically expressed as,

$$Au_{time} = \rho_{time} - \mho_{time}$$

where, ρ_{time} indicates the time occupied to access the request and $\mho_{time}$ is the total amount of time. Fig. 5 the graphical plot displays the comparison of proposed work and existing models in terms of authentication time with respect to number of candidates. From that, it is clearly exposed that the proposed work has attained minimal authentication time then the Darknet, HOD-Net and SPEM existing works. The main reason for attaining such minimal authentication time is due to proposing of pattern-based authentication. In the proposed Slide-Block framework, pattern-based authentication is performed for users, developers, and resource owners by contemplating several credentials, which is implemented by TCA using MCha-poly 1305 algorithm. Once, the authentication is completed the credentials are stored in slide window blockchain. Authentication with minimal time consumption is achieved by executing sliding window blockchain which occupies minimal amount of time to execute authentication while compared with other existing approaches. Whereas, in existing works the authentication is only performed for users with insufficient credentials and traditional blockchain which are tends to increase the authentication time.

The numerical result shows that the Au_{time} of the proposed work is minimized to 550 ms when the number of candidates increased to 100. Meanwhile, for the same number of candidates, the Au_{time} of the existing works Darknet, HOD-Net and SPEM reaches 820 ms, 880 ms and 920 ms of Au_{time} respectively. Overall, the Au_{time} of the proposed Slide-Block framework minimized about 270 ms–370 ms than the other existing works.

5.2.3. Result of packet loss rate

Packet loss rate (Pac_r^{los}) is the number of packets plunge tersely against the total number of packets while transmitted that can be formulated as,

$$Pac_r^{los} = \frac{Avg_r^{pac}}{Tol^{pac}}$$

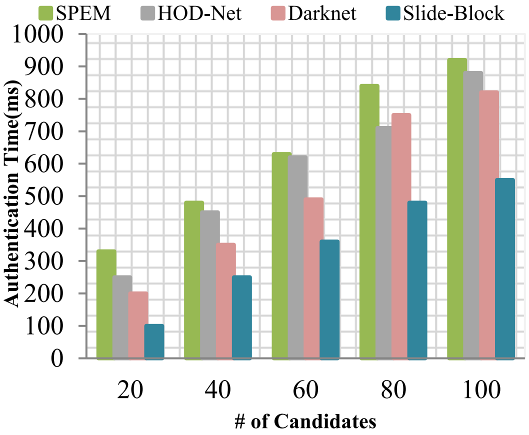


Fig. 5. # of Candidates vs Authentication Time.

where Avg_r^{pac} represents the average number of packet loss while transmission from total packets and Tol^{pac} is the transmitted packets. Fig. 6 the graphical plot indicates the comparison of proposed work and existing models in terms of packet loss rate with respect to number of candidates. From that, it is clearly visible that the proposed work has reached a lower packet loss rate then the Darknet, HOD-Net and SPEM existing works. The main reason for reaching such lower packet loss rate is due to proposing of tri-level access control and privacy attentive data storage. In our work, at first the access control policies are effectively generated using EDDPG. Once, the request initiated by the candidates to TCA, which is responsible for providing access based on attribute role and trust value by utilizing smart contract Here, the TCA redirects those candidate request to SMA, it will evaluate the trust value through constant monitoring as well as historical candidate data. Evaluated trust values are transmitted to TCA, then optimal candidates are selected by AVOA and then the access control are provided in three level permission levels as operator, subscribed users and unsubscribed users that improves the data privacy level thereby minimizing packet loss rate. Besides, in privacy attentive data storage, both data in transmission and rest are encrypted using Mcha-Poly 1305 algorithm where the packet loss rate is minimized. Meanwhile in existing works, the data communication is performed without any privacy concern that leads to high data leakage. Moreover, the sensitive are stored in clouds server without encryption which are tends to high data leakage thereby increasing high packet loss rate.

The numerical result shows that the Pac_r^{los} of the proposed work is reduced to 53.5% when the number of candidates increased to 100. Meanwhile, for the same number of candidates, the Pac_r^{los} of the existing works Darknet, HOD-Net and SPEM increased as 75%, 80% and 85% of Pac_r^{los} respectively. Overall, the Pac_r^{los} of the proposed Slide-Block framework reduced about 21.5%–31.5% than the other existing works.

5.2.4. Result of security strengthen

Security strengthens (Sec^{str}) is the vital metric utilized to estimate the security level of the cloud-based applications during authentication, access control, data storage and vulnerability analysis. High security strengthen enriches the resistance of cloud-based applications against several vulnerabilities. Sec^{str} is denoted as (%).

Fig. 7the graphical plot specifies the comparison of proposed work and existing models in terms of security strengthen with respect to number of candidates. From that, it is clearly observable that the proposed work has achieved maximum rate of security strengthen then the Darknet, HOD-Net and SPEM existing works. The main reason for achieving such maximum security strengthens is due to proposing of authentication and vulnerability analysis. In our work, pattern-based authentication is performed for users, developers, and resource owners by contemplating several credentials, which is implemented by TCA using MCha-poly 1305 algorithm. Once, the authentication is completed the credentials are stored in slide window blockchain and further, the data are encrypted and stored in blockchain as well as data is encrypted before transmission thus improves data privacy. Furthermore, the packet and behavioural features are taken into an account for detecting the attacks using Tweak NB algorithm which are increases the security. However, in existing works, the insufficient metrics and ineffective authentication, as well as lack of authenticating the resource owner leads and developers leads to improper security management. In addition to that, while attack detection the adequate features are considered which reduces the security strengthen.

The numerical result shows that the Sec^{str} of the proposed work is increased to 98% when the number of candidates increased to 100. In contrast, for the same number of candidates, the Sec^{str} of the existing works Darknet, HOD-Net and SPEM reduced as 87%, 80% and 74% of Sec^{str} respectively. Overall, the Sec^{str} of the proposed Slide-Block framework increased about 11%–24% than the other existing works.

5.2.5. Result of communication overhead

Communication overhead (Com_{ϱ}) is describes as the ratio of overhead packets when transmission to receiver which can mathematically expressed as,

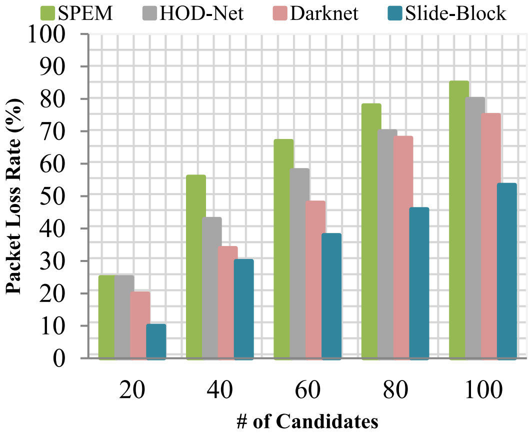


Fig. 6. # of Candidates vs Packet Loss Rate.

$$Com_{\text{ϙ}} = \frac{pac_{\text{ϙ}}}{pac_{trans}}$$

where $pac_{\text{ϙ}}$ is denoted as the overhead packets occurs while transmission and pac_{trans} is the transmitted packets. Fig. 8 the graphical plot symbolizes the comparison of proposed work and existing models in terms of communication overhead with respect to number of candidates. From that, it is clearly visible that the proposed work has achieved a slightest latency then the Darknet, HOD-Net and SPEM existing works. The main reason for achieving such slightest latency is due to performing of pattern-based authentication. In this, the legitimacy of candidate is ensured by TCA using MCha-poly 1305 algorithm based on their credentials. Furthermore, the vulnerability analysis is performed by Tweak NB algorithm by considered several constraints which are tends to minimize the malicious traffic in network thus reduce communication overhead. Whereas, in existing works the authentication is only performed for users and ineffective vulnerability analysis is performed which are leads to increases in high communication overhead.

The numerical result shows that the $Com_{\text{ϙ}}$ of the proposed work is minimized to 0.54 ms when the number of candidates increased to one hundred. In contrast, for the same number of candidates, the $Com_{\text{ϙ}}$ of the existing works Darknet, HOD-Net and SPEM increased as 0.78 ms, 0.8 ms and 0.84 ms of $Com_{\text{ϙ}}$ respectively. Overall, the $Com_{\text{ϙ}}$ of the proposed Slide-Block framework minimized about 0.24ms–0.3 ms than the other existing works.

5.2.6. Result of latency

Latency ($lat_{\text{ʎ}}$) is utilized to evaluate the amount of time occupied for performing data encryption, authentication, access control and vulnerability analysis. $lat_{\text{ʎ}}$ is defined as the difference between the amount of time taken for performing specific task from the above-mentioned tasks to the total amount of time. $lat_{\text{ʎ}}$ can be expressed as,

$$lat_{\text{ʎ}} = time_{tak} - \text{Ծ}_{time}$$

where $time_{tak}$ refers to the time taken for performing the task. Fig. 9 the graphical plot represents the comparison of proposed work and existing models in terms of latency with respect to number of candidates. From that, it is clearly noticeable that the proposed work has attained minimal latency then the Darknet, HOD-Net and SPEM existing works. The main reason for achieving such minimal latency is due to adopting appropriate algorithms and techniques in individual process. Initially, the authentication with minimal time consumption is achieved by executing sliding window blockchain which occupies minimal amount of time to execute authentication. Then, the access control policies are generated by EDDPG, and the tri-level access control is provided by TCA based on attribute and trust level estimated by SMA. Furthermore, the data in transmission rest are encrypted using MCha-Poly 1305 where this algorithm encrypts the data effectively thereby minimizing the number of rounds. In vulnerability analysis, the Tweak NB algorithm is employed for attack detection which removes linear relationship that are tends to minimize high latency. However, in existing works utilization of ineffective algorithms and techniques leads to high latency.

The numerical result shows that the $lat_{\text{ʎ}}$ of the proposed work is reduced to 4700 ms when the number of candidates increased to 100. In contrast, for the same number of candidates, the $lat_{\text{ʎ}}$ of the existing works Darknet, HOD-Net and SPEM increased as 7800 ms, 8400 ms and 9000 ms of $lat_{\text{ʎ}}$ respectively. Overall, the $lat_{\text{ʎ}}$ of the proposed Slide-Block framework reduced about 3100 ms–4300 ms than the other existing works.

5.3. Research summary

In this section, we summarize the experimental results that demonstrate the superior performance of the proposed Slide-Block framework compared to existing approaches. We evaluate the performance of our work in terms of several metrics, including detection rate, authentication time, packet loss, security strength, communication overhead, and latency, presented in Figs. 4–9.

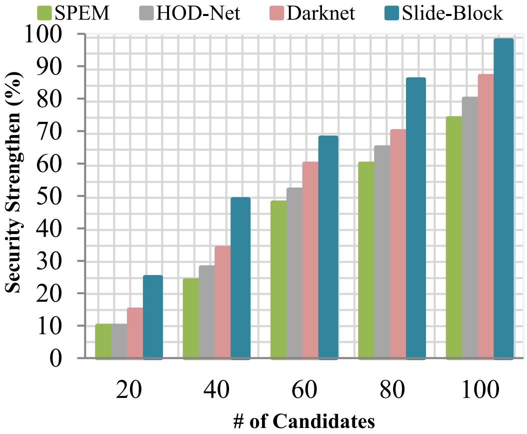


Fig. 7. # of Candidates vs Security Strengthen.

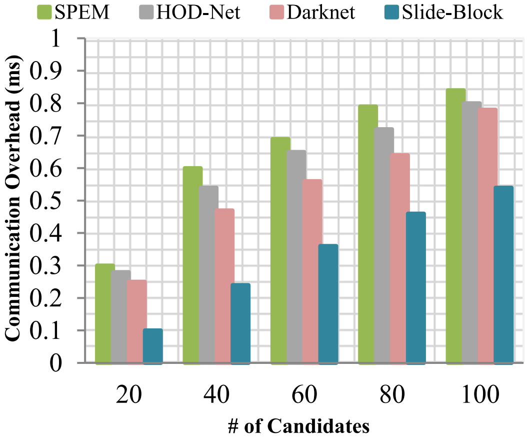


Fig. 8. # of Candidates vs Communication Overhead.

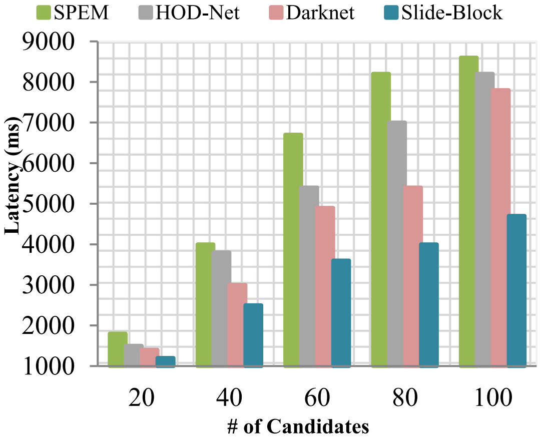


Fig. 9. # of Candidates vs Latency.

Additionally, Table 4 provides a numerical analysis of the performance metrics for proposed and existing works. Finally, we list the research highlights as follows.

- For enhancing application and network security, pattern-based authentication is performed for all candidates and the passwords are encrypted using the MCha-poly 1305 encryption algorithm. This algorithm executes effectively, which permits only legitimate candidates to access the service.
- For efficient authorization, the Tri-level access control policies are fabricated using an enhanced deep deterministic policy gradient algorithm (EDDPG) and provided to the candidate who meets the control policy. The African vulture optimization algorithm is adapted to provide optimal access policies.
- To ensure communication and transaction certainty, the privacy-attentive data storage mechanism is executed, encrypting the data using the MCha-poly 1305 encryption algorithm that minimizes highly sensitive data leakage.
- For amplifying system and service availability, continuous monitoring is established where the intruder vulnerability scanner detects the Tweak NB algorithm detects both the application-based issues and the attack.

6. Conclusion

Improper security management, enormous sensitive data leakage and system and service unavailability are the primary concerns in DevOps-based cloud applications, which are addressed and resolved by our proposed work. For that, initially, the users, developers and resource owners are candidates authenticated based on their credentials by TCA using the Mcha-Poly 1305 algorithm. After that, the access control policies are effectively engendered by EDDPG. Based on trust value evaluated by SMA, attribute and role, the optimal user and developer are selected using AVOA and access control is provided, thus minimizing high data sensitive leakage. Then, data privacy in transmission and rest is enhanced by performing encryption before transmission using the Mcha-Poly 1305 algorithm.

Furthermore, based on the sensitivity, it is decided that the data will be stored in blockchain and cloud servers, improving the

Table 4
Performance analysis of proposed & existing works.

Performance metrics (%)	SPEM	HOD-Net	Darknet	Slide-Block
Detection Rate (%)	≃ 37.86	≃ 45.4	≃ 52.6	≃ 66.4
Authentication Time (ms)	≃ 640	≃ 582	≃ 522	≃ 348
Packet Loss (%)	≃ 62.2	≃ 55.2	≃ 49	≃ 35.5
Security Strengthen (%)	≃ 43.2	≃ 47	≃ 53.2	≃ 65.2
Communication Overhead (ms)	≃ 0.644	≃ 0.598	≃ 0.54	≃ 0.34
Latency (ms)	≃ 5860	≃ 5180	≃ 4500	≃ 3200

application's security. System and service unavailability are the foremost issues; we have proposed vulnerability management and emendation to address this. In this process, we have identified and mitigated both application-based issues and attacks using IVS and Tweak-NB algorithms, respectively, in terms of considering several parameters. The proposed work is implemented by Java/JDK 1.8 to prove this efficacy in several performance metrics such as detection rate, authentication time, packet loss, security strength, communication overhead and latency, where our proposed Slide-Block achieves better performance.

Data availability statement

The data used in this research will be made available on request.

Ethics statement

Informed consent was not required for this study because no specific information from humans is used.

CRediT authorship contribution statement

Gopalakrishnan Sriraman: Writing – review & editing, Writing – original draft, Software, Methodology, Data curation, Conceptualization. **Shriram R:** Validation, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M.S. Khan, A.W. Khan, F. Khan, M.A. Khan, T.K. Whangbo, Critical challenges to adopt DevOps culture in software organizations: a systematic review, IEEE Access 10 (2022) 14339–14349.

[2] D.S. Battina, THE CHALLENGES AND MITIGATION STRATEGIES OF USING DEVOPS DURING SOFTWARE DEVELOPMENT, 2022.

[3] F.L. Almeida, J. Simões, S. Lopes, Exploring the benefits of combining DevOps and agile, Future Internet 14 (2022) 63.

[4] S. Rafi, M.A. Akbar, M. Sánchez-Gordón, R.C. Palacios, DevOps practitioners' Perceptions of the low-code trend, in: Proceedings of the 16th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, 2022.

[5] H. Altunel, B. Say, Software product system model: a customer-value oriented, adaptable, DevOps-based product model, Sn Computer Science 3 (2021).

[6] N. Azad, Understanding DevOps critical success factors and organizational practices, in: 2022 IEEE/ACM International Workshop on Software-Intensive Business (IWSiB), 2022, pp. 83–90.

[7] A. Al-Marsy, P. Chaudhary, J.A. Rodger, A model for examining challenges and opportunities in use of cloud computing for health information systems, Applied System Innovation (2021).

[8] S. Zeb, A. Mahmood, S.A. Khowaja, K. Dev, S.A. Hassan, N.M. Qureshi, M. Gidlund, P. Bellavista, Industry 5.0 is coming: a survey on intelligent NextG wireless networks as technological enablers, ArXiv, abs/2205.09084 (2022).

[9] A. Al-Marsy, P. Chaudhary, J.A. Rodger, A model for examining challenges and opportunities in use of cloud computing for health information systems, Applied System Innovation (2021).

[10] C. Camacho, P.C. Cañizares, L. Llana, A. Núñez, Chaos as a Software Product Line—a platform for improving open hybrid-cloud systems resiliency, Software Pract. Ex. 52 (2022) 1581–1614.

[11] C. Werner, Z.S. Li, D. Lowlind, O. Elazhary, N.A. Ernst, D.E. Damian, Continuously managing NFRs: opportunities and challenges in practice, IEEE Trans. Software Eng. 48 (2021) 2629–2642.

[12] O. Elazhary, C. Werner, Z.S. Li, D. Lowlind, N.A. Ernst, M.D. Storey, Uncovering the benefits and challenges of continuous integration practices, IEEE Trans. Software Eng. 48 (2021) 2570–2583.

[13] R.N. Rajapakse, M. Zahedi, M.A. Babar, An empirical analysis of practitioners' perspectives on security tool integration into DevOps, in: Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), 2021.

[14] O.H. Plant, J.V. Hillegersberg, A. Aldea, How DevOps capabilities leverage firm competitive advantage: a systematic review of empirical evidence, in: 2021 IEEE 23rd Conference on Business Informatics (CBI), 2021, pp. 141–150, 01.

[15] O.H. Plant, J.V. Hillegersberg, A. Aldea, Rethinking IT governance: designing a framework for mitigating risk and fostering internal control in a DevOps environment, Int. J. Account. Inf. Syst. 45 (2022) 100560.

[16] J. Alonso, L. Orue-Echevarria, M. Huarte, CloudOps: towards the operationalization of the cloud continuum: concepts, challenges, and a reference framework, Appl. Sci. 12 (2022) 4347, https://doi.org/10.3390/app12094347.

[17] S.R. Alam, M. Gila, M. Klein, M. Martinasso, T.C. Schulthess, Versatile software-defined HPC and cloud clusters on Alps supercomputer for diverse workflows, Int. J. High Perform. Comput. Appl. (2023).

[18] M.S. Farooq, U.M. Ali, Harnessing the potential of blockchain in DevOps: a framework for distributed integration and development, ArXiv, abs/2306.00462 (2023).
[19] L. Surya, AI and DevOps in information technology and its future in the United States, InfoSciRN: Artif. Intell. (2021).
[20] Applying azure to automate devops for small ML smart sensors, International Research Journal of Modernization in Engineering Technology and Science (2022).
[21] F.J. Abdullayeva, Advanced Persistent Threat attack detection method in cloud computing based on autoencoder and softmax regression algorithm, Array 10 (2021) 100067.
[22] M.A. Alduailij, Q.W. Khan, M. Tahir, M. Sardaraz, M.A. Alduailij, F. Malik, Machine-learning-based DDoS attack detection using mutual information and random forest feature importance method, Symmetry 14 (2022) 1095.
[23] R. SaiSindhuTheja, G.K. Shyam, A machine learning based attack detection and mitigation using a secure SaaS framework, J. King Saud Univ. Comput. Inf. Sci. 34 (2020) 4047–4061.
[24] M.J. Awan, U. Farooq, H.M. Babar, A. Yasin, H. Nobanee, M. Hussain, O. Hakeem, A.M. Zain, Real-time DDoS attack detection system using big data approach, Sustainability (2021).
[25] Ö. Aslan, M. Ozkan-Okay, D. Gupta, Intelligent behavior-based malware detection system on cloud computing environment, IEEE Access 9 (2021) 83252–83271.
[26] L. Karuppusamy, J. Ravi, M. Dabbu, S. Lakshmanan, Chronological salp swarm algorithm based deep belief network for intrusion detection in cloud using fuzzy entropy, International Journal of Numerical Modelling: Electronic Networks 35 (2021).
[27] A.S. Aldallal, Toward efficient intrusion detection system using hybrid deep learning approach, Symmetry 14 (2022) 1916.
[28] X. Qin, Y. Huang, Z. Yang, X. Li, A Blockchain-based access control scheme with multiple attribute authorities for secure cloud data sharing, J. Syst. Archit. 112 (2020) 101854.
[29] H. Gao, Z. Ma, S. Luo, Y. Xu, Z. Wu, BSSPD: a blockchain-based security sharing scheme for personal data with fine-grained access control, Wirel. Commun. Mob. Comput. 2021 (2021) 6658920, 1-6658920:20.
[30] T. Liu, J. Wu, J. Li, J. Li, Y. Li, Efficient decentralized access control for secure data sharing in cloud computing, Concurrency Comput. Pract. Ex. (2021).
[31] X. Qin, Y. Huang, Z. Yang, X. Li, A Blockchain-based access control scheme with multiple attribute authorities for secure cloud data sharing, J. Syst. Archit. 112 (2020) 101854.
[32] Y. Liu, W. Yang, Y. Wang, Y. Liu, An access control model for data security sharing cross-domain in consortium blockchain, IET Blockchain (2023).
[33] R.K. Gupta, K.K. Almuzaini, R.K. Pateriya, K.A. Shah, P.K. Shukla, R. Akwafo, An improved secure key generation using enhanced identity-based encryption for cloud computing in large-scale 5G, Wireless Commun. Mobile Comput. (2022).
[34] P. Velmurugadass, S. Dhanasekaran, S.S. Anand, V.K. Vasudevan, Enhancing Blockchain security in cloud computing with IoT environment using ECIES and cryptography hash algorithm, Mater. Today: Proc. (2020).
[35] M.I. Mihailescu, S.L. Nita, A searchable encryption scheme with biometric authentication and authorization for cloud environments, Cryptogr 6 (2022) 8.
[36] K. Demertzis, K.G. Tsiknas, D. Takezis, C. Skianis, L.S. Iliadis, Darknet traffic big-data analysis and network management to real-time automating the malicious intent detection process by a weight agnostic neural networks framework, ArXiv, abs/2102.08411 (2021).
[37] S. Krishnaveni, S. Sivamohan, S.S. Sridhar, S. Prabakaran, Efficient feature selection and classification through ensemble method for network intrusion detection on cloud computing, Cluster Comput. (2021) 1–19.
[38] S.K. Sarma, Hybrid optimised deep learning-deep belief network for attack detection in the internet of things, J. Exp. Theor. Artif. Intell. 34 (2021) 695–724.
[39] M.A. López-Peña, J. Díaz, J.E. Pérez, H. Humanes, DevOps for IoT systems: fast and continuous monitoring feedback of system availability, IEEE Internet Things J. 7 (2020) 10695–10707.
[40] H. Gao, S. Luo, Z. Ma, X. Yan, Y. Xu, BFR-SE: a blockchain-based fair and reliable searchable encryption scheme for IoT with fine-grained access control in cloud environment, Wireless Commun. Mobile Comput. (2021).
[41] Yurong Liu, Weibo Liu, Mustafa Ali Obaid, Ibrahim Atiatallah Abbas, Exponential stability of Markovian jumping Cohen–Grossberg neural networks with mixed mode-dependent time-delays, Neurocomputing 177 (2016) 409–415, https://doi.org/10.1016/j.neucom.2015.11.046. ISSN 0925-2312.