An Evolutionary Algorithmic Framework- Cloud Based Evidence Collection Architecture

Yunus Khan (yunuskhansgsits@gmail.com)  
Shri Govindram Seksaria Institute of Technology and Science

Sunita Varma  
Shri Govindram Seksaria Institute of Technology and Science

Research Article

Keywords: Evolutionary Computing, Performance Parameters, Fuzzy Logic, Graph neural network, Blockchain Technology, Evidence Collection, Neural Science, SHA-3, Smart Contracts, Coocko Algorithm, SDN Control Plane

Posted Date: July 12th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-630503/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License
An Evolutionary Algorithmic Framework-
Cloud Based Evidence Collection Architecture

1Yunus Khan, 2Sunita Varma
1yunus.khan@gsits@gmail.com, 2sunita.varma@gmail.com
1,2 Department of Computer Science & Engineering, Shri GSITS, Indore, MP

Abstract: Forensic in cloud computing is an advancement of evolutionary modern forensic science that protects against cyber criminals. Single centralize point compilation and storage of data, however, overcome the authenticity of digital evidence. In order to address this serious issue, this article suggests an evolutionary modern algorithm automated forensic platform leveraging infrastructure as a cloud service (IaaS) based on Blockchain concept. This proposed forensic structural design, evidence collection of evidence and stored on a blockchain which is circulated around several peer blocks. Secure Block Verification Mechanism (SBVM) is proposed to Safeguarding the device from unauthorised users. On the bases of level of confidentiality, all data is stored and encrypted at cloud authentication server. Confidentiality-based Algebraically Homomorphic Cryptosystems learning is presented with a fast-forwarding algorithm for encryption. A block in the SDN controller is created for every data and information is stored in the cloud service provider and the history is recorded as metadata data about data. A hash based tree is constructed in each block by Secure Hash Algorithm version - 3 of 512 bits. By implementing graph theory-based graph neural networks in Smart Contracts, our framework enables users to track their data (GNNSC). Finally, the construction of a evidence graph using blockchain data enables evidence analysis. Experiments was carried out in a Python programming and blockchain integrated cloud environment with network simulator-3.30 (for Software Defined Network). As part of result our newly designed forensic architecture using blockchain (FAuB) good results in terms of evidence response time, insertion times of cloud evidence, verification time of evidence, computational overhead of evidence, hashes calculation time, keys generations times of evidence, evidence encryption time, evidence decryptions time, and total overall change rate of evidence, according to a comprehensive comparative study.

Keywords: Evolutionary Computing, Performance Parameters, Fuzzy Logic, Graph neural network, Blockchain Technology, Evidence Collection, Neural Science, SHA-3, Smart Contracts, Cuckoo Algorithm, SDN Control Plane.

1. Introduction:
Cloud computing is a emerging technological concept that, through virtualization technology, provides users with physical resources. The cloud computing industry is growing with the benefit of allowing network accessing to a scalable and elastic combination of shared physical or virtual resources with self owned service provisioning and on demand available service. There is also an enhance in the number of cloud users using cloud computing because of these features. Security risks have begun to develop, however, with the rising cloud computing industry. Several security strategies for the cloud environment are being investigated, with virtualization technologies making it difficult to implement current digital forensic methods [1]. Access to certain system layers is restricted in Software as a Service (SaaS) and Platform as a Service (PaaS) [1] environments when the cloud environment is categorized according to the service model, access to that layer are regulated by Cloud Service Provider (CSP). It is therefore appropriate to supply the log data generated in the inaccessible layer to the CSP through agreements [2]. Investigators have complete control over the evidence in conventional digital forensics. In a cloud environment, however, data centers are geographically distributed, cloud service customers (CSP) exchange physical infrastructure, unreliable data that disappears when the instance is shut down, virtual network, load balancing and auto scaling to provide a smooth service environment. Therefore, prior to a security incident for investigation, it is important not only to record data for cloud forensics, but also to guarantee the truthfulness of the log data while it is impossible for the investigator to directly capture the data and collect the data from the remote server. Forensic architecture is suggested for software-defined networking (SDN) using Internet of Things (IoT) and blockchain. Blockchain's algebraic homomorphic encryption scheme is adapted here. Evidence data collection performed using in the presence of the Software Defined Networking policy. Digital evidence obtained or stored by DataFlow switches that adapt additional forensic techniques for forensic examination. A (PDMS) data management system of
provenance aware has been invented and builds on the existing provenance monitoring framework. The integrity management framework based on the blockchain is proposed by Mchain. Therefore, many analysis makes an attempt to distributed exploitation blockchain technology within the SDN cloud atmosphere. During this analysis, within the cloud atmosphere, we tend to use blockchain concept for cloud digital computer forensics.

1.1 Research Contribution:
In this article, the following contributions have been made to provide additional digital forensics research:

- In case of cloud environment like infrastructure as a cloud service (IaaS), the digital forensics mechanism design is constructing to collect, analyze, and release evidence. Blockchain technology is used to collect evidence.

- Evidence and information is secured against malicious users by using the Secure Block Verification Mechanism (SBVM) authentication mechanism driven by a cloud authentication server (CAS). The SBVM mechanism involves users those are completed successfully secure verification process by means of a globular logic and secret key (SK).

- Encryption based on confidentiality CB- For the generation of digital signature and encryption, the EL GAMAL algorithm is proposed. Key generation is done by cuckoo search optimization algorithm in CB-EL GAMAL to generate strong secret keys. The main contribution of the Algebraically Homomorphic Cryptosystems algorithm based on confidentiality is that the proposed algorithm is based on the data level of sensitivity and adaptive in nature.

- Block was generated by control plane SDN and distributed across the blockchain network for all data and information is stored in the cloud based server. For added security, a Secure Hashing-3 (SHA-3-512) algorithm has been proposed for blockchain accounts. By using neural network-based smart contracts (GNNSC) to graph to track data activities throughout its life cycle, the data source is preserved.

2. Background
Siva Rama Krishna Tummalapalli [15] develops and Bayesian fuzzy clustering and cluster search laid on support vector neural network based intrusion detection mechanism simulator for clustering and two-level classifiers works on cloud environment.

“A cloud forensics tamperproof framework for cloud forensics is developed by author that is available in a cloud environment that is untrusted and multi-tenancy. This framework relies on a forensic system based on the compressed multilayer counting filter, independent of daily cloud activity. No standard forensics preparedness model for cloud environments can be applied properly. A model for improving security, which can be used in a cloud environment. Forensic preparedness is a way of maximizing the potential of an organization to respond to violations [15]. The below mention graph and table shows that number of paper published in various digital libraries like ACM,IEEE, ScienceDirect, Springer and Elsevier.

| Type            | ACM  | IEEE Explore | Science Direct | Springer | Elsevier |
|-----------------|------|--------------|---------------|----------|----------|
| Journals        | 8994 | 209          | 197           | 683      | 506      |
| Book Chapters   | 469  | 3            | 17            | 6909     | 12531    |
| Conference      | 70   | 698          | 3             | 509      | 80       |
| Total Publications | 9533 | 910         | 217           | 62101    | 13137    |

Table 1: Record of Article Types in Various Libraries on this topic

Graph 1: Year-wise analysis of research papers being published in digital libraries.

Cloud log include useful data and information for the computer forensic investigation which is very important. Earlier designed logging systems have a few inconveniences to provide the cloud user with security. The existing
system gives protection and security for user files that are either saved or uploaded by the user or authenticated by the user [16]. This paper secures logging by encrypting cloud logs using encryption techniques and identifying assaults on the cloud framework from DDoS (distributed denial of service) [17].

3. Evidence Collection

In order to classify and access forensic data from different parts and sources in the cloud world, processing of evidence plays a critical role. Evidences are refusal longer housed in one only physical host and data are split into another geographical region. Therefore, after an incident occurs, the evidence is very hard to find [18]. Proofs are obtained from different forensic origins such as switches, routers, servers, virtual machines, hosts, browsers and from in-house storage content media such as hard disk drives, ram image files, physical memories, etc. The information is retrieved from multiple sources. Data collection from the cloud servers, web browser objects, and physical memory analysis also collects evidence.

3.1 Blockchain in Cloud Forensics

Blockchain is one of the overestimated breaking fields and has acquired significant consequence as an invention commonly used in numerous fields. The blockchain is known mostly as a billing book or digital distributed database[19]. The way Blockchain interface, render device costs, monitor and document transactions began to emerge as a revolutionary advance since its introduction in 2008. Blockchain can be inexpensive, removing the could do with to supervise and normalize transactions and communications between various members from the central authorities. Each move is marked cryptographically by other miners who have a record of the entire transaction history in a blockchain. This renders time records that cannot be altered one by one safe, synchronized, and collective [20]. Moreover, blockchain technology is considered to be IT, and can be used in the applications, industry and industrial industries. Figure 1 displays the blockchain design.

![Blockchain Diagram](image)

**Figure 1: Architecture of Blockchain Technology**

4. Proposed Blockchain Based Cloud-Forensic (BCF):

The proposed forensic architecture, called Blockchain-based forensics, is developed with the necessary algorithms in this section. The proposed forensic cloud uses software defined network and blockchain concept collection of evidence and investigation.

4.1 Entities of the Architecture:

The main objective of our experimental study is to acquire reliable proof or evidence in cloud environment and to maintain cloud provenance of data. The following entities comprise the overall forensic system:
1) **Cloud Users (CU):**
Cloud users (CU1, CU2, .... CUn) are included in our system 'n' number. It is permissible for cloud users to save and access evidence at the server cloud.

2) **Cloud Authentication Server (CAS):**
At the start registration of all the cloud clients with CAS to deter unwanted access by users. Key generation and authentication are major responsibilities of CAS.

3) **Cloud Service Provider (CSP):**
Cloud users store up all data outer surface of their cloud on CSP hosted cloud servers. For every piece of data stored in CSP, a Blockchain was developed.

4) **Dataflow open switches (DFSs):**
During this practice, software defined network is used to gather CSP data. We have therefore used many DFSs to relay CSP data to consumers. For data owned database flow regulations applied by the control plane to users DFSs be mainly responsible. DFSs only deploying and modifying flow rules in the software defined network control plane.

5) **Software defined networking Control plane (SDNC):**
The Software defined networking Control plane is responsible for applying network status data flow rules and for gathering all CSP evidence. The Software defined networking Control plane manages Blockchain for proof collection and a block is generated for any CSP data. The complete machine architecture as seen in Figure 2.

Our forensic architecture's principal objective is to capture and conserve appropriate CSP data. We initially developed an efficient verification design to secure the device beginning unlicensed users. Data saved on the CSP is encrypted to ensure secrecy within the cloud setting. Decentralized data processing was planned on the basis of blockchain technologies.

Smart contracts can be used for the motto of recording and storing data history. For successful proof analyses, the graph-based research approach is recommended.

### 4.2 Cloud User Authentication:
CAS is first registered with all cloud customers. User ID and password are user credentials that are taken into account when logging (PW). CAS produces a secret key (SK) for each documented CU by means of the cuckoo algorithm. Both users are valid at any where using the circular theorem's secret code (SC), SK, ID, and P.

---

**Figure 2: Blockchain Based Cloud-Forensic (BCF) architecture**

1) Key Creation and Generation with the help of Cuckoo Algorithm:
Cuckoo Search Algorithm is a newly invented meta heuristic search optimization algorithm used to solve problems of optimization. This is a meta heuristic nature inspired algorithm focused on the brood parasitism of certain cuckoo birds, as well as spontaneous Levy flight walking. It has been carried out in a number of areas. The Cuckoo algorithm is used in this research meant for the main generation of cryptography process.

The ECC equation is usually defined as,

$$ y^2 = x^3 + ax + b $$  \hspace{1cm} (1)

If the curve point (x, y) is a, b represents the value of the curve. On the present curve the 'P' foundation position is preferred and the random Pr(SK) number surrounded by the individual variety is chosen. The below is the public key created:

$$ Pu(SK) = Pr(SK) \times P $$  \hspace{1cm} (2)

We be capable of make out here the random generation of the private key (Pr(SK)) that attackers can crack quickly. The Cuckoo algorithm is used to enhance the key generation process.

4.3 Cuckoo search explanations:

Each egg is a solution in a nest and a new solution is the cuckoo egg. The aim is to substitute not so nice nesting solutions with new and hopefully better solutions (cuckoos). The simplest shape of each nest is an egg [21]. The algorithm can be applied to more complex cases in which several eggs are present in each nest representing a set of solutions.

Three idealized rules are based on CS:
1. Per cuckoo lays one egg on a single basis and dumps the egg into a randomly chosen nest.
2. The better nests with good egg content will hold the next generation.
3. In this scenario the host bird will throw away the egg/give up the nest and make a whole new nest.

For continuous non-linear optimization, the cuckoo optimization algorithm is used. The lifestyle of the cuckoo family of birds is influenced by COA. This development optimization algorithm is based on the life-style of these birds, their egg laying and their breeding features. As other emerging approaches, cuckoo optimization algorithm is introduced by an initial population. Here are two categories of cucumbers in various societies: mature cucumbers and larvae. The algorithm is based on the attempted survival. Any are discarded as they fight for life. The remaining cuckoos migrate to well again seats and begin raising and laying their eggs. Finally, the surviving cuckoos converge in such a way that there is a society of cuckoos with the same rate of profit.

In order to address an optimization issue, the variable values of the problem should take shape from an array. The "habitat" is called this array.

In an optimization problem, the next N_{var} of a habitat will be a 1xN_{var} array that shows the current living location of cuckoos. This array describes as:

$$ Habitat = [X_1 \ldots X_{n}] \hspace{1cm} (3) $$

The suitability (profit) in the current habitat obtains by computing the function profit ( p f ) in the habitat. So:

$$ Profit = f b (habit) = f b [X_1 \ldots X_{n}] \hspace{1cm} (4) $$

In order to establish an optimization search algorithm, a habitat matrix of N_{pop} x N_{var} size will be prepared and a random number of eggs will be allocated for each habitat. allowing for the number of eggs that every one cuckoo lays and as well the space between the cuckoos and the current optimized zone, the laying radius will be calculated. After that, in that zone, the cuckoos start to lie. The laying radius calculates as:

$$ ELR = a \times \frac{Number\ of\ current\ cuckoos\ eggs}{Total\ number\ of\ eggs} \times (Var(high) - Var(low)) \hspace{1cm} (5) $$

Then each cuckoo begins to lay her eggs in the nests within her ELR. So after each laying round, the less profitable percent of eggs (p%) (usually 10 percent) (their profit function is at the lower level) is destroyed. In the host nests, other chicks power up and develop.

The cuckoo's migration

While growing up and getting older, the cuckoos live in their environments, but when the laying time comes, they migrate to superior habitats where the eggs have more chances to survive. The group with the best location will be targeted after composing the groups in different living locations in general (justified area or problem search space), and other cuckoos will migrate there.

When the cuckoos that are grown live all around the environment, it is not easy to find out which group belongs to each cuckoo. The cuckoos will be grouped by the' K means' to solve this issue. Method, which is a traditional method of grouping (finding a K between 3 and 5 is usually acceptable). They don't travel the direct way when the cuckoos migrate to the target. With the deflection of (\phi), they just travel (\lambda % almost a percent) of the way.
These two parameters ($\lambda, \phi$) help cuckoos to explore a larger area. $\lambda$ is a random number between 0 and 1 and $\phi$ is a number between: $-\frac{a}{6} + \frac{b}{6}$. 

(6)

The Pseudo-code of Cuckoo Optimization Algorithm:

```
Algorithm

Start Function objective $f(x)$, $x=(x_1, x_2, ..., XD)$; 
Initial host nest population $x_i I = 1, 2, ..., n$) 
Duration or stop criteria ($t< Max$ generation) 
Get a cuckoo on Levy Flights by random means; 
Analyze $Fi$ fitness 
Select a nest randomly between $n$ (says $j$) 
If ($Fi > Fj$) 
Substitute $j$ for the current result 
Finish If 
A fraction of the worst nests are deserted and new nests are created 
Maintain the right options (or quality solutions nests) 
Grading the solution and finding the right solution 
End for 
Posting and visualizing outcomes of processes 
End Start
```

In the method, the Cuckoo algorithm selects an enhanced vector $f(x)$ and is allotted to $Pr$ (SK). To determine the secret key generated is difficult for cyber criminals because the Cuckoo algorithm selects the random number more optimally.

2) Authentication using Secure Block Verification Mechanism (SBVM):

For those logged-in users, CAS produces secret keys and beginning points. For each operator of a particular circle the root points are $(Ox, Oy)$ co-ordinates. For each user at CAS, the respective credentials (ID, PW, SC) are saved. In all stages of verification, all passwords are checked. The CAS key is a random code that makes it impossible for an attacker to invent the code for each user. By following the equation, a circle is defined by:

$$
(Ax - Ox)^2 + (By - Oy)^2 = R^2 
$$

(7)

Each user builds a SC consisting of origin points by using origin points $(Ax, By)$. The user chooses a SC that follows the circle equation to effectively complete the authentication. While a client has to use the cloud, the client shall have every single one ID and passwords along with the time stamp (TS).

Algorithms illustrate the method of SBVM based authentication. A user who has legitimate passwords will complete validation effectively. By making an allowance for SC next to with TS, the protection level of the SBVM mechanism is increased. Although the SC differs over time, the attacker cannot split the SC. The attacker cannot use SC for the next authentication without being aware of the source points, despite the SC being cracked at a time by the attacker.

4.4 Confidential Data Encryption

Users that have successfully completed the authentication process will enter the cloud computing environment in the planned forensic system. Within the cloud network users store up their information in the form of a ciphertext with a digital signature. Users at this point settle on the extent of confidentiality of results. For eg, data such as banking data, Identity data and other information such as humorous videos data are sometimes referred to as sensitive information, whereas movies are unconfidential data. When mentioned in the prior paragraph, secret keys are produced by means of the cuckoo search algorithm. Data are translated into ciphertext by using the created powerful secret key in the confidentiality encryption (CB-EL GAMAL) algorithm.

```
Algorithm 1 SBVM authorization mechanism (Pseudocode)

Input: passwords for users 
Output: Status of authentication 
1. Begin 
2. For VCU // Registration of Cloud User 
3. Register $ID, Password$ $\rightarrow$ CAS
```
4. CAS uses cuckoo algorithms to produce Secret Key (SK)
5. CAS provides SKs; Origin Points → CU
6. End for // Registration completed of Cloud User
7. If Ui requires to right to use cloud // Require validation
8. Calculate secret code (SC) via Equation (7)
9. CUi submits IDi, Passwordi, SC, TS → CAS
10. CAS verifies User credentials
   If (User Credentials are correct match)
   Ui = Authorized user
   Else
   Ui = Unauthorized user
11. End if
12. Else
13. End process
14. End if
15. End
16. End
17. End
18. End
19. End

The EL GAMAL algorithm is paired through the CB-EL GAMAL algorithm probability and algebra. Algebraically homogenous crypto-systems is a quick-release solution that is embedded in the decryption and encryption process across many unseen layers. The input layer of the Homomorphic Cryptosystems algorithm is used to encrypt and Pu(SK) is initialized and encryption is done on secret layers. CB-EL GAMAL, however, is confidential and carries out the following data encryption procedures.

Algorithm 2 demonstrates the overall technique with an efficient hidden key for CB-EL GAMAL algorithm. The CB-EL GAMAL algorithm being proposed is shown in Fig. 3.

**Algorithm 2** CB-EL GAMAL (Pseudocode)

---

**Input:** Public key and input data  
**Outputs:** Cipher-text

1. Initialize public key \((Pu(SK))\) and Input data \((d)\)
2. If \((d = \text{Confidential})\)
3. Split data \(d \rightarrow d_1\) and \(d_2\)
4. For data \(d_1\)
5. Calculate ciphertext 1\((c_1)\) as,
6. \(c_1 = d_1 \varphi d_2\)
7. End for
8. For data \(d_2\)
9. Initialize \(Pu(SK), d_2\) at input layer
10. Calculate cipher-text 2 \((c_2)\) in hidden layers as,
11. \(c_2 = k \times P // k \text{ is random number}\)
12. \(c_b = d_2 + k \times Pu(SK)\)
13. \(c_2 = (c_a, c_b)\)
14. End for
15. Get cipher-text \((c)\) as,
16. \(c = (c_1, c_2)\)
17. Else
18. For \(d\)
19. Repeat steps number (8 to 13)
20. End for
21. End if
21. End

Similarly, the input layer begins ciphertext and the output layer gets the original text when the data are decrypted. The participation in encryption of the Homomorphic Cryptosystems Algebraically algorithm [24] strengthens data security. In order to retain documentation of possession, data will be signed by the customer sooner than outsourcing to the cloud computing surroundings. Digitally signed using the EL GAMAL algorithm is generate the same as motioned. The hash value is first created to sign data as

\[ HV = \text{Hash}(d) \]  

(8)

The digital signature is then created,

\[ \text{Signature} = \frac{HV + \text{Pr(SK)}K2}{K1} \]  

(9)

Where random numbers are k1 and k2. The data have got to be registered by the same data proprietor if data are updated or ownership transferred.

4.5 Efficient Collection of Evidence using Blockchain Technology

In cybercrimes, digital data is an important source for analysis. The offenders will conceal their details and erase the evidence in a variety of parts of the infrastructure as a service cloud system. The key issue with the infrastructure as a service cloud infrastructure be with the intention of data collection is spread on a wide scale. In comparison, cloud consumers monitor more than scholars, making it a difficult challenge to gather and preserve data. SDN and blockchain technologies are utilized in the proposed digital forensic infrastructure to gather and maintain cloud forensic data to combat all this issue. The evidence be store within the blockchain ledger within the be in charge of the software defined network control plane. In cloud forensics, some relevant meanings are:

**Evidence Integrity:** Integrity of evidence guarantees that the certificate reflects correctly the information contained in the PC. Several areas of the cloud impacts knowledge respectfully, but preserving integrity is a core component of the cloud crime scene investigation. The recognized technique to encrypt trustworthy information uses validated hash techniques, for example, MD5, SHA1, SHA256.

**Data Origin:** It is a form of authentication that corroborates a party as the (original) source of specified data generated in the past at some (typically unspecified) time.

**Data volatility:** Unpredictability, after the power is switched off, leads to memory or power failure of the material. This is an important problem from a measurable standpoint since both memory and CPU procedures would vanish if the server crashes. If virtual computers are involved, these difficulties increase (VM). For eg, IaaS VM does not have permanent storage in this way, if the VM crashes, all the volatile data may be lost.

**Custody in Chain:** The method of retaining and recording the chronological past of treating data as digital information can be represented. Data may be moved from the first responder, prosecutors (one or more) and judges to various layers of hierarchy of the automated forensics system. The proof is treated during this lifetime by these provisional owners. Because any evidence-based measure is held in the blockchain, our proposed work holds the custody chain.

**Digital Evidence Ownership proof:** Digital evidence of ownership is defined here as proof of existing digital proof of ownership. The data can be managed by multiple owners during its lifespan. If the status of the data has shifted, the original owner must sign the data, to retain proof of cloud-based ownership. The patented evidence is retained in the framework as the transition in ownership is still preserved in the blockchain data history.

**Graph Neural Network (GNN) Based Smart contracts:** It is a computer program which tracks data history automatically. When the necessary conditions are met, the smart contract is activated and executed. To optimize smart contracts, graph theory algorithm rules are deployed in this work.

**Data lineage:** It documents the history of possession and the paper process throughout its entire life cycle. In other words, the record sequence showing the behaviour taken by the data is known as a lineage or origin. With the aid of blockchain, we retain the data root, that is, in our work any alteration made to the data is saved and traced by GNNNSC in the blockchain.

The evidence has the hash value of the public ledger in the blockchain. We give a SHA-3-512 algorithm, better in terms of security, for hash value generation. The hash value in SHA-3 is determined accordingly for each block:

\[ \text{Hash} = \text{sponge} [g, \text{pad}, q] (T; L) \]  

(10)
The hash unique value be calculated here designed for input that is transaction (T) padding, rate q, permutation g function, and output length L. The hash value is often created by the "sponge building" mechanism in SHA-3-512 as in EQATERY (10) rather than by the "sponge building" procedure. Accepting SHA-3-512 to hash calculation bring various benefits over the current system with respect to time consumption and protection. Let us look at the U1 user's data d1 on time t1 in the cloud. After that the block is formed for d1 and the hash value is created by SHA-3-512. Each transaction, i.e., the shift kept on d1, is based on the time the GNNSC block was installed in the system. Every update is processed and circulated as evidence in the blockchain network between the peers. The log contains user Name, IP address, time and all other hardware information of the proof. The proof log the information history is kept as proof for each change found in details in the blockchain. Past of data can involve lines that describe change, ownership transition and other behaviour on cloud-specific data. Algorithm 3 explains the method of collecting evidence. In favor of each single data reside within the cloud, the evidence be gathered and preserved within the blockchain here. Furthermore, in the cloud environment, GNNSC track and wheel the ease of access of data stored by users.

Algorithm 3: Efficient Evidence Collection Method (Pseudocode)

Input: cloud user data
Output: collected digital evidence
1. Start
2. For every C\(U_i \in C\)
3. Creates Cloud users with GNNSC
4. End for
5. For every data
6. \(U_i\) stores \(d_i\) in Infrastructure as a Cloud Service
7. Create block for \(d_i\)
8. Calculate Hash value (\(d_i\)) with the help of Equation (10)
9. Track \(d_i\) and modernize evidence
10. End for
11. For every transactions on \(d_i\)
12. Store Log timestamp, source or origin IP, Virtual machine disk file
transactions hash, details, Virtual Machine server, action made etc.
13. If (Graphtheoryrulesarenotrue) // GNNSC
14. Report Generation
15. Else
16. Dont produce the report
17. End of If
18. End of for
19. End

During our initiative, smart contracts be used for alert a cloud server when a graph theory law, which is often integrated as a proof record within the blockchain, is met. A lot of registered users be able to right of entry the information contained in the cloud atmosphere. This thesis draws intelligent contracts from the graphology that functions on a secret stage of data. The smart contract is execute by means of the graph theory principles used in the framework. Fig. 4 demonstrates GNNSC's pictorial representation. FSC presence tracks all big activities conducted under the data contained in the cloud server machine. Thus, any accurate evidence of the cloud server machine evidence is gathered and the correctness of evidence is conserved using blockchain technologies in our proposed forensic architecture.

Table 2 displays the laws of the graph theory in GNNSC. On the basis of these set of laws, the statement is generated and saved like evidence log. An modification of the data made after previous access is the previous danger. If earlier hazard is restricted and information non confidential, the log right of entry evidence will be overlooked and the report will not be produced. The produced statement is well thought-out otherwise noteworthy and stored in the blockchain.
Figure 4: pictorial representation of GNNSC.

| Data type       | Past Risk | Action performed | Report Generation by GNNSC |
|-----------------|-----------|------------------|----------------------------|
| Non Confidential| Low       | Read             | No                         |
| Confidential    | Low       | Read             | No                         |
| Non Confidential| Low       | Edit             | Yes                        |
| Confidential    | Low       | Edit             | Yes                        |
| Non Confidential| Low       | Delete           | Yes                        |
| Confidential    | Low       | Delete           | Yes                        |
| Non Confidential| High      | Read             | No                         |
| Confidential    | High      | Read             | Yes                        |
| Non Confidential| High      | Edit             | No                         |
| Confidential    | High      | Edit             | Yes                        |
| Non Confidential| High      | Delete           | Yes                        |
| Confidential    | High      | Delete           | Yes                        |

Table 2: Attributes rule for GNNSC

5. Cloud Forensic Investigation

If a cyber-crime has been detected, the designated investigator (police, lawyers) must examine the digital evidence. The investigator is also authenticated by CAS prior to the inquiry. If a criminal enters an election voting room, his basic details, such as his Aadhar number and voter id, is kept in the election commission's database. If he tries to update or erase the evidence history by hacking the database or deleting or modifying his entry into the voting space, he is attempting to upgrade or remove the evidence history.

Given that every one of evidence record logs stored within blockchain, we know that is a distributed ledger, our suggested forensic architecture will be useful in this situation. She must also pass the powerful authentication before gaining access to the device. According to the investigator, the following steps should be taken when analyzing data.

Evidence Identification

The first step in a digital forensic investigation is to locate a possible evidence source of reliable evidence. As a result, the investigator must obtain legal consent from the relevant authority.

Evidence Acquisition

The investigator possibly will gather round all evidence logs records as of blockchain by way of the consent of officially authorized authorities. The evidence log record inside the study contains mutually credentials of user and evidence based on hardware. During this time, the investigator have got to adhere to court restrictions while also abiding by SLA agreements.

Evidence Analysis

The investigator then goes through all of the data logs and compiles a report on digital evidence. Logical graph with evidence for better research This paper proposes a graph of proof. The evidence is used to build a graph of data with matching log attributes. If the perpetrator checks in at a polling site, the history of persons visited in voting center,
i.e., original details, be submitted just before the cloud with the administrations of election commission, i.e., registered person. The evidence is currently being developed on blockchain for every single one log record attributes (source_IP, time_stamp, actions_made, transaction_hash, server of virtual machine, DFS_ID, and the like).

Think about the case where the suspect's check-in history was changed at \( t_2 \). Then, in a subsequent block of log attributes, the next log is modified. Similarly, as soon as the hacker tries to access the information or erase it from the cloud, this be treated as evidence and recorded in the subsequent block. The investigator must complete the following steps in order to create a graph of evidence:

• Sequentially arrange evidence according to the timestamp
• Established each and every one evidence through its attributes of log record
• Build an evidence graph according the evidence order and log record attributes.

Table II shows properties of the survey evidence collection. A graph of evidence can be constructed using this data, as seen in Figure 5. The investigator can see from the graph of evidence that the suspect has edited (modified) the evidence (User X). However, the authorised user's location and IP addresses are different.

**Reporting of Evidence**

Into the evidence review level, every one of of the evidence within the Graph of Evidence is authenticated using a cryptographic digital signature that is kept together in the midst of the value of hash and data. Data should be sign earlier than being sent toward the cloud, according to our proposal. As a result, at what time an intruder modify evidence data, he or she should generate a digital signed signature.

For all evidence, the current transaction's hash value at blockchain storage repository. The hash significance of data stored in the cloud must match the Merkle tree root value of the block. The investigator compiles a report based on these findings and submits it to the court as digital testimony. From acquisition to submission to jurisdiction, algorithm number 4 illustrates the collection process of evidence.

As a result, our designed architecture of cloud forensic, which incorporates blockchain and SDN technologies, allows for secure collecting evidence from the cloud. A powerful authentication protocol stops unauthorised users from gaining access to the cloud environment, while a sensitivity aware encryption process improves data protection. Evidence storage using blockchain and SDN is an intellectual approach for distributed data protection. From evidence analysis to evidence reporting to the court, our designed architecture of cloud forensic facilitates the whole investigation.

**Investigational Result Evaluation**
Within this investigation result evolution, we compare the efficiency measurements of the our designed architecture of cloud forensic to earlier research contribution. We present our simulation environment in this section and at that time judge against our designed architecture of cloud forensic to the prior centralized log record process collection.

**A. Configuration and Simulation:**

In a combined simulation platform, we configure our designed architecture of cloud forensic. Using CloudSim, we introduced an IaaS cloud environment in Python. Blockchain is built data storage mechanism of IaaS cloud in Python Programming, as described in algorithm.

---

**Pseudocode Algorithm 1:** for Forensic Investigation

---

**Evidence as:** input  
**Graph of evidence as an:** output

1. Begin
2. Using the SBVM system, verify the investigator’s identity.
3. Determine the case’s evidence.
4. Collect evidence in the form of {Evidence_Identity, Time_stamp, IP_Source, uploaded_User, accessed_User, Performed_Actions, Hash_Tn, BlockHash, Location_Attribute, virtual machine server, and OFS ID} from blockchain.
5. Create an evidence graph by means attributes of evidence.
6. For every one of evidence
7. Ensure that {Block_Hash & IP_Source} are right and correct.
8. If this is the case (Verification D True)
9. Verify the signature //Validation of evidence
10. If this is the case (Signature is valid)
11. Collect reliable evidence
12. Else
13. Prepare illegitimate evidence
14. End if
15. End if
16. End for
17. Prepare and share the copy of evidence to the juridictory court.
18. End

---

Both of the tests were run on Ubuntu OS by means of an Core-i7 Intel CPU running next to 2.80 ghs, 16 GB of RAM, and a 1000 GB SSD. The simulator version network 3.30 simulators, that is committed to network simulation for software defined network, is also compatible with the cloud and blockchain environment. The Python platform’s performance is merged by ns-3.30, in the direction of create a simulation environment.

The Ubuntu operating system underpins the entire work; we use NetBeans-8.2 for PYTHON blockchain setup, Network Simulator-3 for software based network simulator, and CloudSim for IaaS cloud deployment. In Table 4 of our experiments explains important parameters of simulation used in the direction of apply our designed architecture of cloud forensic. Prior to we get interested in the study, well go through a real-world use of the proposed forensic scheme.

| Parameters                        | Value   |
|-----------------------------------|---------|
| Number of Users                   | 120     |
| Number of OFS’s                   | 8       |
| Number of Controllers             | 1       |
| Number of Cloud Authentication Servers(CAS) | 1       |
| Number of Keys Generated          | 120     |
| CUCKOO                            | HMS     |
|                                   | 120     |
|                  | HMCR,PAR   | 0.996, 0.9 |
|------------------|------------|------------|
|                  | Maximum Iteration | 120        |
| DECC             | Number of Hidden Layers | 4          |
|                  | Key Size       | 256        |
| SHA-3            | Block Size     | 576        |
|                  | Word Size      | 64 bits    |
|                  | Number of Rounds | 24        |
|                  | Customized Contract | GNNSC    |
|                  | Maximum Handles | 2048       |
| Cloud            | Number of Virtual Machines | 35 |
|                  | Average RAM    | 512 MB     |
|                  | Average Bandwidth | 1000000 MB |
| Simulation Time  | 100 ms        |            |

**Table 4: Simulation configuration setting**

The Proof-of-Work principle is used by the Miner to validate the blockchain. A corresponding block is generated for each piece of data that the user stores in the cloud environment, and stored the hash values.

1) **Use case Diagram of our Designed architecture of cloud forensic using Blockchain (FAuB)**

IaaS be a cloud environment so as to is extremely versatile and can be used by any rising business. Many real-world implementations will benefit from our designed architecture of cloud forensic IaaS platform. In this paper, we look at one application of the proposed work in crime detection. Consider several voting centres that store their data such as voter records, financial information, maintenance information, personnel information, and surveillance information into IaaS cloud. Each and every one data be encrypted depending on top of the extent of data protection earlier than being out sourced toward the cloud, as per our job. Furthermore, each voting center's administrators must be CAS-registered. The SDN controller collects evidence designed for every one of data store within the cloud atmosphere and stores it on blockchain. Additionally, each admin may use GNNSC to monitor their data.

Figure 6 depicts an example of the proposed use case. Consider the case of a suspect who voted for a few hours at polling centre A. The suspect’s information will then be found in voting centre A’s election record file. Furthermore, video of the perpetrator in the polling centre will be used in the data obtained from security cameras. This could aid detectives in locating the suspect as soon as possible. Any change made to the voter registration database and surveillance data is recorded within the blockchain as evidence. The perpetrator will erase or change the register of voter registry and data of surveillance contained inside the cloud if we don't have a good forensics mechanism architecture. Every evidence is preserved into the blockchain, that is a distributed block ledger, in our proposed forensic architecture. We also store the VM logs in the blockchain as evidence. The investigator will obtain information from the blockchain even though the hacker changes and modifies the details into the cloud. Plotting an evidence graph with the collected data log will reveal whether there are any differences in the evidence. The investigator will pass digital evidence with a CoC to court based on the evidence obtained from blockchain.

**Figure 6: Digital Forensic Crime Investigation use Case Diagram**

**B. Comparative Analysis**
This section compares our designed architecture of cloud forensic to the current CFLOG [5] framework for safely collecting digital evidence. In CFLOG collect and store the evidence in centralized fashion, which is a major contrast between current forensic infrastructure and CFLOG. As mentioned in section III, this causes several problems. We designed a architecture of cloud forensic that collects in addition to stores digital information safely using SDN and blockchain technologies to overcome these challenges.

1) **Response Time Comparison**

The time in use for user on the way to get a response to a data request is known as response time. The number of users interested with the forensic method validates this metric. In supplementary terminology, response time refers to time it takes the forensic method to provide the necessary information or documentation to the users.

In graph 2, the designed architecture of cloud forensic SDN-blockchain based forensic framework is compared to current CFLOG framework, which has a centralised framework. The numeral of requests as of users increase by means of the increase inside the number of users in both works, so response time steadily increases with the increasing the user numbers. Still for a more user numbers, our designed architecture of cloud forensic responds to requested users easily. That use of software defined network technologies improves ability of scalling, or ability to accommodate a large number of users at the same time. As a result, any cloud user can link to server of cloud instantly as well as download data requested by user. Similarly, prosecutor be able to obtain information from the blockchain without having to wait for the SDN controller to respond.

As a result, the proposed forensic architecture reduces time of response. CSP performs together data managing as well as evidence collection in a consolidated fashion in CFLOG, which increases response time when there are a large number of users. The CFLOG system takes 100ms to answer in the presence of 120 users, while the designed architecture of cloud forensic system takes 72ms intended for the identical amount of user numbers. As a result, the designed architecture of cloud forensic outperforms the CFLOG system by 27%.

![Graph 2: Response Time Comparison Analysis](image)

2) **Evidence Insertion Time Comparison**

The point in time it take to (or create) insert digital data of evidence collected on a server of cloud is known as evidence insertion time. It know how to exist describe at the same time as the time it takes SDN plane controller to generate proof meant for the CSP’s stored data inside our analysis.

As shows in graph 3 insertion of evidence period as a function of the different user numbers. When the user number grows, so does the volume of data that must be alive store and the number of piece digital evidence that must be generated. As a result, the amount of time it takes to insert evidence increases as the number of users increases in all works. Every one of evidence be collected and stored in a consolidated way beneath the supervision of CSP in the CFLOG process.
Graph 3: Evidence Insertion Time Comparison Analysis

As a result, the centralised evidence collection procedure lengthens the time it takes to insert evidence. In addition, we protect the history of data in our work, which means that each change to data is treated as evidence and incorporated into the blockchain. The SDN controller, on the other hand, is in charge of creating and preserving documentation without the intervention of CSP. As a result, relative to previous work, evidence insertion in blockchain takes less time.

3) Evidence Verification Time Comparison

The times it takes an investigator in the direction of collection and validate evidences commencing a blockchain is known as evidence verification time.

The time taken for verification of evidence within the CFLOG process and proposed forensic system is compared in Graph 4. The proposed automated forensic technology achieves the shortest possible time for evidence verification. The investigator would use CSP to collect evidence in the CFLOG process, and verification is done in the conventional method. Instead of CSP, the investigator in the suggested work aggregates all evidence from the controller. In addition, for improved study, evidence testing is carried out by creating a Graph of Evidence. Furthermore, we suggested SHA-3-based hash computation to maintain evidence consistency while reducing time consumption. As a result, we gain evidence integrity with the least amount of time spent on evidence verification.

Graph 4: Evidence Verification Comparison Analysis

In the presence of ten users, CFLOG takes 62 milliseconds to collect and validate digital evidence, while planned digital forensics FAuB takes just 37 milliseconds, reducing verification time by nearly half.

4) Computational Overhead Comparison

The bandwidth quantity used in the direction of execute a particular activity (transfer data, reading, update, generation of evidence, and verification of evidence) within the system of forensic is known as computational overhead.

Graph 5 depicts a comparison of computational overhead based on the different user numbers. Because the amount of data on the way to be interpreted grows in tandem with the number of users, the computational overhead increases. The computational overhead is raised in the absence of blockchain technologies owing to centralised device administration. Both data and evidence collection in CFLOG takes place in CSP, which raises the overhead.
The suggested forensic method, on the other hand, keeps evidence processing like collection, hash computation, and preservation on the SDN controller, reducing total computational overhead. Furthermore, incorporating SDN technology increases scalability without adding overhead. Thus, the proposed digital forensic infrastructure adds 7KB of overhead for ten cloud customers, while the CFLOG framework adds 10KB of overhead.

5) Total Change Rate Comparison
The rate of total change is calculated by dividing amount of evidence modification by total evidences exist within the forensic framework faced problems with the old CFLOG system. When a hacker person changes data into organize on the way to destroy evidence, the net modification rate rises. The collected data must be accurate, and the evidence’s accuracy must be maintained, for an effective forensic method. Since only registered users are included in the proposed forensics scheme, any information along with data as of unauthorised users is refused. Furthermore, we use blockchain technologies based on top of the SHA3 algorithm to maintain the credibility of evidence.

According to our findings, the proposed forensic method modifies 11.1% of the evidence. However, since we guarantee credibility, CoC, and PoO for evidence, this alteration is also registered as evidence in blockchain. Because of (i) centralised infrastructure ever since CSP be able to be malicious (ii) node single vulnerability (attacker just wants to break CSP’s), (iii) no credibility is protected, as well as (iv) interference to unauthorised user’s accessing, approximately 60% of evidence is changed using the CFLOG process. We overcome all issues by means of the help of blockchain and SDN technologies, that reduces system's overall change total rate. Table 5 compares the cumulative outcomes of the CFLOG process and the proposed forensic system in terms of performance measurements. We will see that each metric has improved with the proposed digital forensic FAuB architecture.

6) Efficiency of CB-EL GAMAL with Cuckoo Algorithm
The elliptic curve cryptography (ECC) algorithm is regularly used designed for digital signature concept in blockchain technology. On the other hand, there are several issues with key generation, encryption, and decryption. We suggested the CB-EL GAMAL algorithm with the Cuckoo Search Optimization algorithm for key generation to
improve the conventional ECC algorithm. As a result, we compare our proposed CB-EL GAMAL algorithm to the Paillier encryption algorithm proposed for blockchain technology using the Cuckoo Search Optimization algorithm. The suggested CB-EL GAMAL algorithm is examined in detail from Graph 7 to Graph 9. For a stable blockchain architecture, the Paillier encryption algorithm is proposed in. The Paillier encryption scheme, on the other hand, quickly improves key generation, encryption, and decryption times. The Paillier scheme consumes more time because it involves massive homomorphic computations.

**Graph 7: Key Generation Comparison Analysis**

**Graph 8: Encryption Time Comparison Analysis**

**Graph 9: Decryption Time Comparison Analysis**
On the other hand, data encryption is necessary in an environment of cloud, and here determination be several users. The algorithm of Paillier takes an average of 500 milliseconds to generate a key. Encryption and decryption, on the other hand, necessitate a significant amount of time, which is incompatible with the cloud environment.

The proposed CB-EL GAMAL algorithm, on the other hand, reduces key generation time by using the CUCKOO algorithm, which has a quick convergence time. Similarly, the CB-EL GAMAL algorithm's deep architecture reduces the time taken for encryption and decryption. As a result, the suggested SA-ECC algorithm outperforms the traditional algorithm in terms of increasing protection without increasing time consumption.

7) SHA-3 Algorithm Efficiency

The most widely used algorithm hashing used in blockchain technology is (SHA-256 bit)v2 is the. Hash computation in our proposed forensic scheme to increase hash computation time and security standard is calculated by SHA-3 algorithm.

Graph 10 compares the hash computation time of the proposed SHA-3 algorithm to that of the previous (SHA-256)2 algorithm. In review SHA-3 reduces calculation time of hash for 100 users to 16 milliseconds lacking sacrificing security. Inside general, SHA-3 outperforms SHA-256 against a variety of security threats, including length extension attacks. As a result, Merkle tree SHA-3 algorithm construct a tree and increases protection without adding time to the process.

Overall, the proposed digital forensic FAuB architecture outperforms the current CFLOG scheme, according to the report. The use of blockchain and SDN technologies increases the efficiency and scalability of the system.

6. Conclusion:

In this research work, with blockchain technology, a valuable architecture of digital forensic is proposed to gather and safeguard unfailing evidence from the infrastructure as a service cloud environment. Cloud authentication server CAS, with a secure verification mechanism known as the SBVM mechanism, authenticates all cloud users. The CB-EL GAMAL algorithm was proposed for data security. The cuckoo algorithm to generate optimal keys prior to this. A block in the controller is created for every information stored into the cloud. The integrity of evidence is ensured in every block by SHA-3-512 based hash tree building. All evidence is collected and blockchain technology maintains evidence integrity, data origin, data lineage, digital evidence ownership evidence and custody chain. GNNSC is deployed in the system in order to trace data activities. The CB-EL GAMAL algorithm is proposed for data protection. The Cuckoo algorithm generates optimum keys before that. At the controller, a block is generated for each piece of cloud data. Merkle tree structure based on SHA-3 guarantees the consistency of evidence in each block. All documentation is collected, and the chain of custody and proof of ownership (CoC, PoO) are maintained using blockchain technologies. GNNSC is installed in the system to monitor data events. Finally, the use in a graph of evidence analysis simplifies evidence analysis. Overall, the forensic device is investigated using a Python and ns-3.30 simulation environment. Experimental findings suggest that the proposed forensic architecture outperforms the current unified forensic system. In order to improve the digital forensic infrastructure, we want to integrate network forensics in software based networks as well as cloud forensics in the future.

Declarations

Ethics approval and consent to participate
Not applicable.
Consent to publish
Not applicable.

Availability of data and materials
The datasets generated during and/or analyzed during the current study are not publicly available, but are available from the corresponding author who was an organizer of the study.

Competing interests
The authors declare that they have no competing interests.

Funding
No Funding till date but we applied for in a government organization.

Authors' contributions
All of the authors contributed to the conception of the study, drafting and critical revision of the manuscript, and provided final approval of the manuscript.

7. References:
[1] Syed A.A., Shahzad M. & Farhan S. (2020). Analysis of Cloud Forensics Techniques for Emerging Technologies International Conference on Computing, Networking, Telecommunications & Engineering Sciences Applications (CoNTESA) Tirana, Albania, Albania.
[2] Ahmed A., Hany F.A. & Gary B.W.(2019). Experts reviews of a cloud forensic readiness framework for organizations Journal of Cloud Computing volume 8, Article number: 11.
[3] Abdullahi A.K., Aman J., Mohd N.Y., Aminu M., Mohamad K.I. & Nor R.M.N. (2019) Evidence Collection and Forensic Challenges in Cloud Environment. MACE Technical Journal (MTJ) MTJ Vol.1(01) , pp. 8-18 eISSN: 2710-6632.
[4] Omi A., Arnisha A., Asraf U. & Manowarul I.(2020). Cloud Forensics: Challenges and Blockchain Based Solutions”, IJ. Wireless and Microwave Technologies, 2020, 5, 1-12 Published in MECS (http://www.mecs-press.org) DOI: 10.5815/ijwmt.2020.05.01.
[5] Samuel A.K., Suhardi & Tutun J.(2020) Modeling Cloud Forensics Readiness using MetaAnalysis Approach. IEEE, International Conference on Information Technology Systems and Innovation (ICITSI), Bandung - Padang, Indonesia, Indonesia.
[6] Haider A.K., Gregory E. & Herbert D (2019). Blockchain for Modern Digital Forensics: The Chain-of-Custody as a Distributed Ledger. Part of the Advanced Sciences and Technologies for Security Applications book series (ASTSA), Springer.
[7] Akbarzadeh A. & Shadkam E. (2015). The Study of Cuckoo Optimization Algorithm For Production Planning Problem. International Journal of Computer-Aided Technologies (IJCAx) Vol.2, No.3.
[8] Fran C., Thomas K.D. & Constantinou P. (2019). A systematic literature review of blockchain-based applications: Current status, classification and open issues. Telematics and Informatics Volume 36, Pages 55-81, Elsevier.
[9] Ezz E.H. & Manjaiah D.H An efficient digital forensic model for cybercrimes investigation in cloud computing. Multimedia Tools and Applications volume 80, pages14255–14282, Springer.
[10] Mndeep K., Navreet K. & Suman K. (2016). A Literature Review On Cyber Forensic And Its Analysis Tools International Journal of Advanced Research In Computer Engineering And Communication Technology Volume 5, Issue 1 ISSN Online 2278-1021 ISSN Print 2319 5940.
[11] Shweta N. J. & Geetha R. C.(2020), Secure Log Scheme for Cloud Forensics. Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), IEEE, Palladam, India, India.
[12] Srivastava P. & Choudhary A. (2021). Evolving Evidence Gathering Process: Cloud Forensics. Proceedings of International Conference on Big Data, Machine Learning and their Applications, Lecture Notes in Networks and Systems 150, https://doi.org/10.1007/978-981-15-8377-3_20, © Springer Nature Singapore Pte Ltd.
[13] Liwen P., Jing L. & Jin Li (2020) Information Fusion-Based Digital Forensics Framework in Cloud Environment ”, 3rd International Conference on Artificial Intelligence and Big Data (ICAIBD), IEEE, Chengdu, China, China.
[14] Pedro R.B. (2019), Computer Forensics in Cloud Computing Systems. Budapest International Research in Exact Sciences (BirEx) Journal 1(1):71-86. DOI: 10.33258/birez.v1i1.141.
[15] Siva R.K.T. & Chakravarthy A. S. N.(2020). Intrusion detection system for cloud forensics using Bayesian fuzzy clustering and optimization based SVNN. Springer-Verlag GmbH Germany, part of Springer Nature.
[16] Sebastian S., Kim K.R.C. & Nhien A. L. (2020). Digital Forensic Approaches for Cloud Service Models: A Survey Cyber and Digital Forensic Investigations, pp 175-199, Springer. [6] Emi M. & Mehrdad S. S.(2016). Digital forensics research on cloud computing: An investigation of cloud forensics solutions. IEEE Symposium on Technologies for Homeland Security (HST) Waltham, MA, USA.
[17] Gayatri S., Saurabh S. & Wandra K.H. (2020), Exploration of Vulnerabilities, Threats and Forensic Issues and its impact on the Distributed Environment of Cloud and its mitigation. Elsevier, Proceedia Computer Science, Volume 167, Pages 163-173.
[18] Harihara S.G., Akila S.S., Gayathri A. & Jebin A.V.C.A. (2019). Digital Forensics Using Blockchain, International Journal of Recent Technology and Engineering (IRTE) ISSN: 2278-3787, Volume-8, Issue-2511.
[19] Fei Y., Yunzhi Z., Xiao F., Bin L., Xiaojiang D. & Mohsen G.(2020). TamForen: A tamper-proof cloud forensic framework. Trans Emerging Tel Tech, 2020:e4178. wileyonlinelibrary.com/journal/ett ©. John Wiley & Sons, Ltd., DOI: 10.1002/ett.4178.
[20] Mervat M., Yasmine A. & Galal E.(2014) Extension and Application of El-Gamal Encryption Scheme”, Researchgate Conference Paper, DOI: 10.1109/WCCAIS.2014.6916627.
[21] Jie Z., Ganqu C., Zhengyan Z., Cheng Y., Zhiyuan L., Lifeng W.C.L. & Maosong S. (2019), “Graph Neural Networks: A Review of Methods and Applications”, arXiv:1812.08434v4 [cs.LG] 10.