

SURVEY

Big Data Privacy and Security Using Abundant Data Recovery Techniques and Data Obliviousness Methodologies

SNEHALATA FUNDE  **AND GANDHARBA SWAIN** 

Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Guntur, Vaddeswaram, Andhra Pradesh 522302, India

Corresponding author: Snehalata Funde (snehalatafunde@gmail.com)

ABSTRACT The concept of big data security is introduced in this article along with many features. It illustrates the need for security in healthcare systems as the volume of data increases continuously over the period of time. The necessity of big data security as well as several big data analytics phases highlighted. It covers many big data privacy-preserving strategies. Many digital storage solutions being used in today's world are designed to work only with fixed format of the data. This paper introduces some methods for maintaining metadata obliviousness. The oblivious RAM technology mentioned in the research article address security concerns and it can be handled with the daily increase in data in several industries. Security needs are introduced at many phases of big data creation, such as information extraction, storage systems, and analytics of the information. Additionally, it presents several data recovery methods for recovering original data in the event of a data crash. This paper covers several data categorization methods for sorting data into normal and sensitive categories as well as methods for anomaly detection. It discusses the advantages and disadvantages of various security measures.

INDEX TERMS Security, privacy, obliviousness, data recovery.


I. INTRODUCTION

The core concepts and phases of big data are introduced in this section. At several stages of big data production, such as data generation, data storage, and data analytics, security requirements are introduced. The importance of security in big data, cloud, and Internet of Things (IoT) infrastructure is discussed in this section. It discusses about several security measures that must be defended against various kinds of attacks. It discusses on the security of healthcare systems. It gives a general overview of the medical industry and discusses how patient data must be safeguarded if it is stored in a dispersed setting.

A. BIG DATA

Big data is data that has large volume, heterogeneity, speed, and volatility all at the same time. The term "high volume"

denotes to the significant quantity of data that is produced on a daily basis by a variety of companies and organizations. The rapid proliferation of new data types, many of which are going to be incorporated in current datasets that have been acquired from a variety of companies and devices, is referred to as heterogeneity. The term "high speed" refers to the rapid rate at which data is collected or captured from a variety of social networking sites into a database [1]. Huge amounts of data present additional difficulties to the security systems that are already in place due to the variety, volume, and unpredictability of the data. Most data storage models used today are designed to work with data that has been properly arranged. The currently available encryption methods are inefficient for use with massive data as once data has been encrypted, producing keys, encrypting it, and then decrypting it takes a significant amount of time. For instance, the data collected from patients, various sensors, social networking sites, and other communications mediums are all quite large. The acquisition of new information invariably results in an increase

The associate editor coordinating the review of this manuscript and approving it for publication was Mansoor Ahmed .

in the risk of the majority of data being compromised [2]. Information leakage is a very significant obstacle in the protection of large data. When various firms employ large amounts of information to improve their company quality, hackers exploit big data platforms to launch attacks against those organizations. Hackers attempt to collect those majority of the data, such as social network, emails, e-commerce, phone numbers, and location, in order to get ready for this attack, and this can make the attack more accurate and result in a significant loss [2], [3]. Hackers have the ability to readily access cumulative data from big data databases in the digital world. When hackers obtain enormous amounts of data all at once, it leads to a rise in the number of efforts made to access data and a decrease in the cost of hacking. The various aspects of big data can be broken down into two categories (i) big data operations, and (ii) big data analytics. The processing of large amounts of data is broken up into four distinct steps. First data generation stage is all about the process by which a significant quantity of data is produced from several resources like social networking sites, corporate enterprises, health care centers, banking sectors, media, educational organizations, military, government organizations, and financial sectors. It is getting increasingly challenging to handle data and address concerns regarding security as the size of this data endures to grow on a daily basis while appearing at a rapid rate [4].

In the second stage, raw data are compiled from a variety of sources, and these data might be in various arrangements. So, to get data into proper storage supporting structure, data transfer is done on it. In the data records that have been acquired, there is a possibility of data duplication as well as the inclusion of worthless data; hence, data pre-processing is carried out in order to obtain correct data and minimize size to a significant degree [4]. The amount of data created from the many different sources is enormous. In order to accommodate such a massive amount of information, numerous data storage facilities are available. If data cannot be contained within a single system, it may be saved on remote workstations using the Hadoop Distributed File System (HDFS). The infrastructure used to collect data should have the ability to recreate data in a dynamic manner for a variety of various sorts of values [4], [5]. The analysis of the data is the final and most important stage of the big data process. In today's world, it is absolutely necessary for businesses to perform data analysis because they must compete with other enterprises. An extensive amount of data is subjected to market data analysis in the context of trade in order to extract information that is of value. However, while performing analysis on data that is not in an organized manner, it is important to utilize these databases.

B. BIG DATA SECURITY AND PROTECTION: INFORMATION DRIVEN VIEW

The following is a definition and characterization of security as it applies to the field of cyber security [27]:

- Reliability: the system's accurate service can be preserved within a specific stage even in the presence of

disturbances such as malfunctions and cyber-attacks; this ability is referred to as reliability.

- Confidentiality is the property that ensures data or information is kept secret from individuals or processes that are not authorised to access it. In the dependability framework that has been suggested [27], this refers to the property that it will be impossible for unauthorised individuals or processes to notice the values or the subtle variables that are associated with the relevant systems.
- Availability denotes readiness to provide the appropriate service. The behaviour of the system as it is given, provided that it falls within the error tolerance boundary, is considered to be the proper service.
- Integrity refers to the absence of any malicious external disruption that could cause the system to produce an output that is not consistent with the targeted service.
- Maintainability refers to the capacity to undergo alterations as well as repairs.
- Authenticity is the capacity to give services with origins that can be demonstrated.
- Non-repudiation ensures that the services rendered cannot be contested at a later date.

Despite the fact that several safety solutions are developed, the vast majority of them do not focus on large datasets. There have been recent advancements made in the areas of privacy and data security pertaining to large amounts of energy-related data [28]. These recent accomplishments cover three essential facets of the security of large amounts of energy-related data: i) big data concerned cryptographic systems [28], [29], (ii) big data secrecy preserving monetary users [30], and (iii) abnormality discovery with big data [30].

C. THE SIGNIFICANCE OF DATA SECURITY FOR BIG DATA

Increasing data in vast amounts across a variety of applications is a significant challenge for data security at each of the stages listed above. The proliferation of internet use is leading to an increase of data. The two most significant concerns in information technology are the safety and confidentiality of data transfers. Regardless of this, the safety and protection become more of a mystery as the amount of data grows. When we look at this from a security perspective, the current cryptographic safeguards are not up to the task of handling massive amounts of data. Therefore, it is necessary to investigate and develop effective systems that can deal with data that is either organised, semi-organized, or unstructured. Due to the open nature of large amounts of information, there is a risk that data will be damaged or altered while being transmitted. Scientist Ron collected 2.8 gigabytes worth of information belonging to Facebook users and made it available for download on the internet. Therefore, the question of privacy is brought back into focus. The most recent storage technology can only hold a maximum of 4 terabytes of data on each disc. It would take 250,000 discs to store 1 Exabyte, which would be insurmountable given the state of the art in terms of transmission [6]. This suggests that innovation is required

not only in storage technology but also in communication systems that guarantee data safety.

The requirements for maintaining the integrity of information and protecting users' privacy have been taken into account by cryptography. Encryption, on the other hand, is not capable of solving the majority of the security problems [7]. When it comes to marketing, where companies try to put their products and services before the safety of their customers, this can become a significant problem. This is due to the fact that when customers are sorted into categories based on their behaviours, there is a greater chance that customers will act inappropriately. Advertisers continue to use massive amounts of information to target individuals through web-based networking media platforms such as web crawlers and email, despite the risk of doing harm to their businesses.

Internet of Things (IoT) drastic evolution has amplified the amount of information detecting devices that are connected to the internet. These devices help people, devices, and things recognise their connection with one another. In addition to this, people are still dedicated to increasing the effectiveness of the data collecting done by devices connected to the internet of things, as seen in [8] and [9]. Unimaginable quantity of data is being produced and stored on the platform offered by cloud service providers [10]. Many of the applications and services that make up smart cities will be housed in the cloud because of the cloud's high performance, scalability, and dependability of its data centres. As a result, residents of smart cities and cloud services may rely on the service providers that serve smart cities so that the smart city services and applications they generate can be hosted, built, or deployed [11]. Aside from that, the fact that pay as per requirement is an option compels the majority of traditional businesses to actively shift their data to the cloud. Not only is the cloud the destination of workloads, but it also enables effective operation practises, which in turn allows businesses to have greater agility and flexibility. Both the digital transformation of businesses and the modernisation of networks have been boosted as a result of this [12].

The United Nations delivered its report on the computerized economy in 2019, and it features that the advanced economy is turning into a vital main thrust for monetary advancement. The global gross domestic product that can be attributed to the digital economy ranges from 4.5 percent to 15.5 percent, according to certain estimates [13]. Processing in the cloud, which is helpful for advancing the profound coordination of the Internet, large information, computerized reasoning, and genuine economy, and which is at the centre of speeding up the development of a cutting-edge financial framework. The worldwide market for public cloud services is expected to increase by 17 percent by the year 2020, getting \$266 billion, up from \$227 billion in 2019. This forecast comes from Gartner, Inc.

When applied to real-world scenarios, the concept of cloud may be broken down into five distinct classifications: public, personal, private, hybrid and communal cloud storage. When businesses use the public cloud, they may outsource their

data storage needs to third-party cloud storage providers like Amazon Web Services and Alibaba Cloud storage with no need to construct and manage their own server infrastructure. Only users who have been given permission can access the data. Personal cloud, which also goes by the name mobile cloud, is a subset of public cloud. However, in contrast to public cloud, personal cloud caters to individual customers and offers public cloud storage services to them. When utilising a private cloud, businesses are required to set up cloud infrastructures and recruit qualified personnel to operate and sustain their servers [14]. This guarantees that the private cloud possesses a greater level of safety than the public cloud storage and that the organisation retains full authority over the data stored within the private cloud. However, the price will escalate significantly. This storage approach is better suited for major corporations that store a substantial volume of data that is both sensitive and expensive.

A private and public cloud are the two components that make up a hybrid cloud, which combines the benefits of both types of clouds [15]. Massive data presents businesses with a number of challenges, including expansion of storage space, data sharing, effective data transfer, reduced costs, and increased data protection. When the amount of data stored exceeds the petabyte level, the limitations imposed by NAS and SAN [16] immediately cause a rise in the cost of sustaining equipment throughout the subsequent time. These challenges can be overcome, but they are not without their difficulties. They are unable to fulfil all of the standards in their entirety of the company for the consistency, accessibility, and safety of mass data and other gauges.

As traditional security techniques are not efficient for big data this article discusses some techniques which can deal with the security of big data in an efficient way. This article comprises different sections on security in healthcare, a literature review of different privacy-preserving techniques, oblivious RAM path hiding techniques, data recovery techniques, data classification techniques for sensitive and normal data classification, future directions, and a conclusion.

II. SECURITY IN HEALTHCARE

Significant leaps forward in terms of digital technology have been made since the turn of the 21st century, which are currently causing significant shifts in the structure of healthcare systems all around the world. A revolution in the medical field is about to be ushered in by the gradual and methodical transition that healthcare systems are undergoing from paper-based records to electronic records [17]. These kinds of advances make it possible to deliver healthcare facilities with a great level of efficacy and flexibility by delivering a platform that enables the effective sharing of healthcare data across various parties. As a result of this development, records that were formerly kept on paper are now being digitized and stored as electronic data, such as Medical Record (MR), Health Records (HR), Personal medical Records (PMR), and Health Data (HD). PMRs hold private data that is handled and supervised on a regular basis by the patients themselves or

by the patient's relatives, in contrast to HRs and MRs, which are patient fitness records that are managed by healthcare specialists. Electronic health records, often known as computerized patient records or simply HD, are a patient's smart health records that have been arranged in a systematic manner [18]. These records contain a wide variety of information, including health histories, medication, immunization status, lab examination reports, and private patient data. When compared to traditional paper-based records, HD systems offer a remarkable number of advantages. Electronic health records, in contrast to paper records, need significantly fewer people, physical storage and time [19].

HRs have many benefits, including the ability to access clinical data more quickly and easily, to preserve effectiveness of clinical workflows, the reduction of medical faults, the improvement of patient protection, cheap costs associated with medical care, and improved and more robust support for clinical decision-making. After becoming aware of the benefits that HD systems have to offer, more than ninety percent of the healthcare facilities in Australia have implemented this structure for promoting operative health resource allocation and effective healthcare [19]. It has been demonstrated and validated by a wide range of users that HDs are capable of delivering superior administration of healthcare services. The shift away from traditional healthcare systems and toward online medical care, then again, presents a new set of issues when it comes to protecting patients' privacy, secrecy, and the integrity of their medical records.

Computing in the cloud is a relatively new concept in the realm of digital technology that is currently seeing widespread use in the field of healthcare [20]. Not only does it make it simple to store medical records, but it also makes it straightforward for many stakeholders to easily communicate or transmit medical information. The massive dissemination of fitness data in this big data era has necessitated the expanding role of cloud systems for storing an indefinite amount of data and for the purpose of facilitating its accessibility all over the internet [21]. It makes it easy for all parties involved in the healthcare system, including healthcare professionals, doctors, and patients, to generate, save, and retrieve facts about their individual medical histories. This is accomplished despite the fact that both time and space are hurdles. Cloud services offer a multitude of advantages, including enhanced productivity and efficiency, decreased costs associated with storing, accessing, processing, and updating data, and a plethora of other advantages. Because the data is stored and processed on a widespread grid of isolated servers that are integrated and managed as an individual ecosystem and utilized by multiple users from a variety of places. It is vulnerable to invasion and, as a result, poses a risk to the confidentiality and safety of the information it contains. In addition, the vast majority of medical data is extremely delicate and must be kept confidential; as a result, the storage of this data on servers belonging to third parties inevitably raises the risks [22]. In general, a patient could have many healthcare suppliers, such as primary care

doctors, specialists and therapists in addition to multiple insurer provider's dental and eye care. [23]. Because of the vulnerable form of health information, there is a quick need to foster a strategy that is safer, proficient, and powerful for the sharing and getting information among the numerous partners.

Even while Electronic Health Record (EHR) are subject to a variety of disputes in the healthcare industry with regard to confidentiality and unauthorized access and the most significant one is which pertains to the confidentiality and safety of patient data [22]. The risks range from malware assaults, which compromise the reliability and privacy of clinical information, to Distributed Denial of Service (DDoS) assaults, which are equipped for denying the frameworks of their capacity to give compelling patient consideration. Malware assaults compromise the reliability and privacy of medical data. DDoS attacks deprive the systems of their ability to provide effective patient care. Cyberattacks that brought on by ransomware may have broader repercussions than simply a loss of money or an invasion of privacy [24]. Hackers in the United States gained access to a large amount of individual well-being information, together with the social numbers of over 10 lac patients, after breaking into the database of Civic Well-being Systems, which is part of a prominent clinical organization. The breach occurred in the United States. In a similar instance, the internet vigilante group Anonymous propelled a distributed denial of service attack (DDoS) against the websites of many hospitals, which crippled the provision of medical services [25]. These occurrences brought to light the urgent requirement for EHR to preserve and secure the credibility, accessibility, safety, and protection of safeguarded wellbeing data. In this setting, the function of cyber security is of the utmost importance in terms of preventing, detecting, and responding to instances of unauthenticated access to health data, as well as the influence that this has on monetary, political, and social contentions. It is the responsibility of healthcare practitioners, as outlined in the Health Insurance Portability and Accountability Act, to maintain the privacy of patients' well-being information [26].

III. LITERATURE REVIEW

A. PRIVACY PRESERVING TECHNIQUES

Figure 1 shows different privacy preserving techniques, which includes cryptographic and non-cryptographic techniques.

1) SYMMETRIC KEY CRYPTOGRAPHY (SKE)

The SKE is successful in electronic HR systems because it employs the same shared secret key. Effective HR exchange requires additional access control mechanisms, which increases complexity. Advanced Encryption Standard (AES), Rivest Cipher 4 (RC4), A5/1, Data Encryption Standard (DES), Blow Fish, and other SKE-based algorithms are often used. Lee proposed symmetric cryptosystems for HIPAA-compliant cryptographic key management.

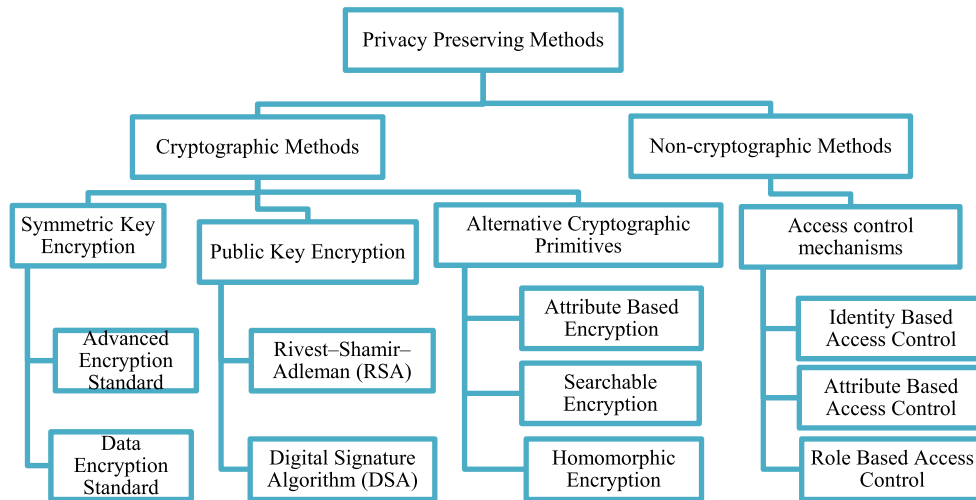


FIGURE 1. Privacy preserving techniques.

The scheme's three steps are record-keeping, encryption, and decryption. The specific patient must get registered with Server of Government (SG) in order to get Server of Health-care provider (SH) medical services. Encrypting patient information requires the user's PIN or biometric authentication. Combining the patient's master key hash with the health-care provider's session key generates a system successfully with cryptographic checksum. Decryption may be done with patient consent or in emergencies. Li et al. [27] developed a safe health record sharing technique to promote patient and record unsinkability. MRs get encrypted by using a one-time secret key. Doctors use digital signatures to deal with the electronic records. It needs a record number, which contains the patient ID, patient's medical card identity seed, and the doctor's random value R to get the medical record. Every key in the method encrypts one medical record to ensure privacy. The smart data card-based identification seed prevents unauthorized access to medical records.

2) ASYMMETRIC KEY CRYPTOGRAPHY

Public Key Encryption (PKE) utilizes public and private keys. Independent PKE strategies are extravagant because of sluggish activities and huge key sizes. PKE strategies are more proficient when combined with SKE strategies to encode the items and secure the symmetric keys utilizing public-private key matches. Design in [28] involves PKE for verification, secrecy, integrity, access control, and non-repudiation, while electronic HRs are scrambled utilizing a typical symmetric key made by medical care suppliers. PKE joins public keys to client personalities through computerized testaments, an enrolment authority, a declaration authority, an endorsement vault data set, and a testament Management System. This engineering lays out a solid electronic HR sharing stage for patients and suppliers. The electronic HR cloud and medical services suppliers validate each other by marking archives with the source's confidential key. Just the designated medical services supplier might affirm the

mark and gain the wellbeing information. Online Referral and Appointment Planer (ORAP) safeguards medical information at the user side in this framework [29]. In the ORAP approach, electronic HRs are housed in a physician's office. Records are encrypted and signed using the receiving entity's public key before being delivered to the cloud. Only authorized entities may decode them. German medical services telematics foundation parts were utilized to offer secure encryption and marks for all archives conveyed to patients' wellbeing histories. Mashima and Ahamad [30] created a patient-centric observing system to protect cloud-stored health data. This research built a system that provides people express or implicit control over their medical information. PKE is used to encrypt health records [31].

The Universal Designated Verifier Signatures (UDVS) system is implemented to limit patient data use to approved organizations. This strategy jeopardises the record's secrecy since the issuer has access to its information, hash values, and signatures. Xun Yi et al. [18] describe a multiparty architecture that preserves patient privacy by encrypting electronic HRs with a single public key and requiring all parties' cooperation to decode. This PKE-based solution uses ElGamal Threshold asymmetric cryptography. This approach uses integrated exponentiation, which avoids re-encryption. This prohibits servers from collaborating with up to $n-1$ others, enabling them to survive external and internal assaults and achieve n server node combined authentication with single database. Narayan et al. [32] presented a cloud-based electronic HR solution using symmetric, public, and attribute-based cryptography. Encryption with symmetric cryptography and a file with metadata including access controls will safeguard medical data. Location data encrypted with the use of broadcast Cipher Text Policy Attribute Based Encryption (CP-ABE) before being stored on cloud server. This method permits immediate reversal without re-encrypting the data, but patients must re-encrypt and update access rules. Trusted authorities can view all encrypted data.

3) ATTRIBUTE BASED ENCRYPTION

Sahai and Waters proposed quality oriented encryption as a method for safeguarding cloud information utilizing public key encryption with encryption and decoding relying upon client credits. The encryption in ABE depends on the entrance structure strategy, which expresses that the code text must be decoded when the client credits equal the encrypted attributes [33]. Key Policy Attribute-Based Encryption (KP-ABE) and Text Policy Attribute-Based Encryption (CP-ABE) are the two significant sorts of ABE. The retrieval strategy in KP-ABE is enciphered in the client's anonymous key, and if the client trait matches the retrieval strategy decryption is possible. However, in CP-ABE every client's confidential key is attached to a bunch of characteristics, and encrypted data is related with a common property, which can be decoded provided that the client credits match the retrieval strategy. By utilizing PKE for versatile approval, this ABE-based arrangement maintains the anonymity of electronic HR [34].

Before the clinical information is shipped off the cloud, the individual's smartcard makes a transaction Code, which is the consent confidential. The patient's card and transaction code are used for confirmation, and PKE is utilized for approval. To encode the clinical information, the wellbeing proficient should enter the transaction code, and the encryption or decryption work creates a public key to encrypt data in light of the hash worth of the patient's personality and transaction code. Authorization and transaction code from a Private Key Generator (PKG) can be utilized to get the original information.

Yu et al. [35] list the difficulties of accomplishing secrecy, adaptability, and fine-grained admittance to re-appropriated statistics in the server of cloud. By consolidating methods like KP-ABE, ABE and Proxy Re-encryption as combined encryption procedure to guarantee fine-grained admittance control. This arrangement tackles challenges like key dispersion. By utilizing key conveyance, a solitary client's encrypted information will be shared across numerous clients. Re-encryption of documents and private key are appointed to cloud machines. The cloud maintains a duplicate of the client's private key for updating secret key parts and once again encoding records. In cloud servers, lazy re-encryption is used to lessen load of computation in the system. When the document items and keys have been changed after client disavowal, it can keep renounced clients from catching the refreshed data. The patient encodes area-based data through broadcast CP-ABE, which permits them to store it on a cloud. By consolidating ABE and PKE with Keyword Search (PEKS), this method furthermore offers a catchphrase scan functionality for directing private examination in encoded information without presenting the match to the cloud. Despite the fact that this technique considers direct reversal without re-encryption, it causes extra computational costs in light of the fact that the patient is answerable for re-encryption and altering retrieval policies. One more hindrance is the interior weakness of the confided in power's

admittance to encoded documents without reference to a permissioned client.

4) SYMMETRIC SEARCHABLE ENCRYPTION (SSE)

Since of the enormous rise of massive information, many things are being moved to cloud servers on a wide scale. Since clinical information and electronic HRs are moved to remote cloud servers that are vulnerable to assaults like DoS and adversary assaults. Cloud information will be encoded for information security and to stay away from data spilling. Typical looking through techniques are unimaginable in light of the fact that the wellbeing information is encrypted and kept on outsider cloud servers. Because of the trouble of looking through scrambled information, SSE has been proposed to permit searching over encoded cloud information [36]. This raises issues, for example, (1) how does the information proprietor allow the information client search consents. (2) How do verified clients look for scrambled information that has been kept on cloud. SSE is one of the choices. SSE is a cryptographic crude that permits clients to look through scrambled material without detection of delicate facts to untrusted locales. Search queries are done on scrambled cipher data with the assistance of a client provided secret entrance strategy. This strategy depends on symmetric cryptography [37], which considers controlled look through in circumstances where the untrusted server can't recover the first plaintext.

5) HOMOMORPHIC ENCRYPTION (HE)

HE is a kind of encoding that performs calculation on cipher texts in which the information is procured in an encoded state and yields the consequence of tasks as though they were directed on the plaintext when unscrambled. To maintain patient security, Barni et al. [38] proposed a multiparty method for handling the encoded Electrocardiogram (ECG) using HE. For electronic wellbeing organizations, privacy maintaining characteristic based verification techniques have been laid out, in which clients' certain characteristics are utilized to verify clients in an E-wellbeing framework. To give information security, the recommended framework utilizes homomorphic encryption, which however has an extremely high calculation cost. Gentry presented totally homomorphic encryption, which permits an erratic number of arithmetic operations over the scrambled information, though Somewhat HE conducts a predetermined number of homomorphic tasks by assessing circuits of given acuity [39].

On account of their failure, completely homomorphic encryption-based procedures are unfeasible. Somewhat HE was introduced by Lauter et al. [40] to perform calculations over encoded information. To further develop e-wellbeing information security in confidential cloud OpenStack stages, this strategy carries out a mixture design that utilizes homomorphic encryption and RSA (Rivest-Shamir-Adleman). Rather than the cloud supplier controlling their cryptographic tasks and key administration, this design permits cloud

clients to make it happen. Sergiu et al. fostered a protection safeguarding determination model in view of homomorphic encryption that processes information without uncovering any private data to the cloud supplier [41]. Prior to being transferred to cloud servers, information will be encoded utilizing the confidential key of the particular client, and information assessment will be finished on scrambled information, with the results staying absent to the cloud. To safeguard client security, this technique mixes state of the art parts, for example, trans encoding, programmed assemblage, parallelization, and message pressing.

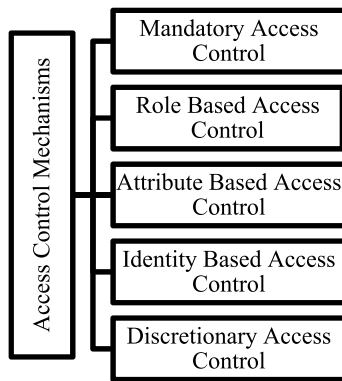


FIGURE 2. Access control mechanisms.

6) NON-CRYPTOGRAPHIC APPROACHES

Figure 2 shows classification of access control mechanisms. To carry out information security control, non-cryptographic methods generally use strategy-based authorization mechanisms, for example, access control approaches. With regards to electronic HR frameworks, information access is exceptionally confidential, and information is put away on outsider servers. As encryption methods, access control strategies are undeniable and fundamental. In a medical care data framework, access control restricts the retrieval and activity of reports in the electronic HR framework, giving key security deterrents to information protection. The proprietor of the thing has total authority over the applications in DAC (Discretionary Access Control). The premise of DAC is that admission to things is conceded in light of the subject's personality.

In Mandatory Access Control (MAC), access strategy choices are chosen by a focal consultant as opposed to by individual item proprietors, and the proprietor can't change access consents [42]. Role Based Access Control (RBAC) figures out which exercises might be performed on which items in light of their work capacities. Jobs are relegated to subjects, and the jobs are related with authorizations that characterize which actions can be executed on which substances. Attribute Based Access Control (ABAC) is a verification-based retrieval control framework in which access choices are made in light of an assortment of client characterized qualities, and requesters are conceded object

access in view of traits that consent to strategy rules. Identity Based Access Control (IBAC) is a strategy for controlling access with the use of a person's confirmed ID.

Khan and Ken [43] supported utilizing RBAC models optional access control to make a setting delicate fine-grained admittance control strategy for individual wellbeing data. This technique utilizes the eTRON engineering, which utilizes PKE for verification and the Diffie-Hellman calculation for protected key sharing. Harsha et al. [44] introduced a patient-driven property-based strategy in which each patient health document is scrambled and kept in an e-wellbeing cloud, alongside a quality-based retrieval strategy that controls entry to the particular asset and furthermore utilizes an intermediary re-encryption method to help validated clients in decoding the proper patient health documents. This framework can endure assaults in light of property plot and can likewise give on-request client disavowal. Suhair and Rajendra [45] proposed a worldview that beats RBAC and ABAC's deficiencies. This paper introduced a BiLayer Access Control (BLAC) framework in which qualities and jobs are joined, and an access demand is contrasted with pseudo-jobs prior to being checked contrary to the strategy's guidelines.

B. OBLIVIOUS RAM PATH HIDING TECHNIQUES

Zhao et al. [46] proposed cycle ORAM for protecting access pattern in retrieving the data in untrusted storage. It provides security definition for metadata as below.

Security definition: ORAM divides data into identically sized blocks, which are defined in terms of security. If the following information is kept from the server, an ORAM method is considered secure: 1) the blocks being accessed 2) the last time the blocks were accessed 3) the number of times a block or sequence has been accessed 4) the access pattern and 5) the operation type. A prerequisite for the client is that it must enable encryption in order to conceal the aforementioned information.

The client performs encryption after retrieving a block from the server to stop the block's plaintext and cipher text from mapping to one another. These are ORAM's security definitions.

$$\text{Let } X = ((ID_L, OPR_L, DT_L), \dots, (ID_L, OPR_L, DT_L))$$

denote the access structure from the user, where index L denotes the most recent retrieval. The j^{th} access type of operation is indicated by the parameter OPR_j . Data is substituted with parameter DT_j if the operation is a write. Block index is represented by parameter ID_j . Let $ORAM(X)$ represent the sequence that results from translating ORAM. The server cannot computationally distinguish between $ORAM(X)$ and $ORAM(\bar{X})$ for other request sequences of the same length, such as \bar{X} . A strong probability exists that $ORAM(X)$ is consistent with X . The concept of Ring ORAM is used in cyclic ORAM. Two features of Ring ORAM were presented by this system: 1) data structure, 2) operations. The associated symbols are shown in Table 1.

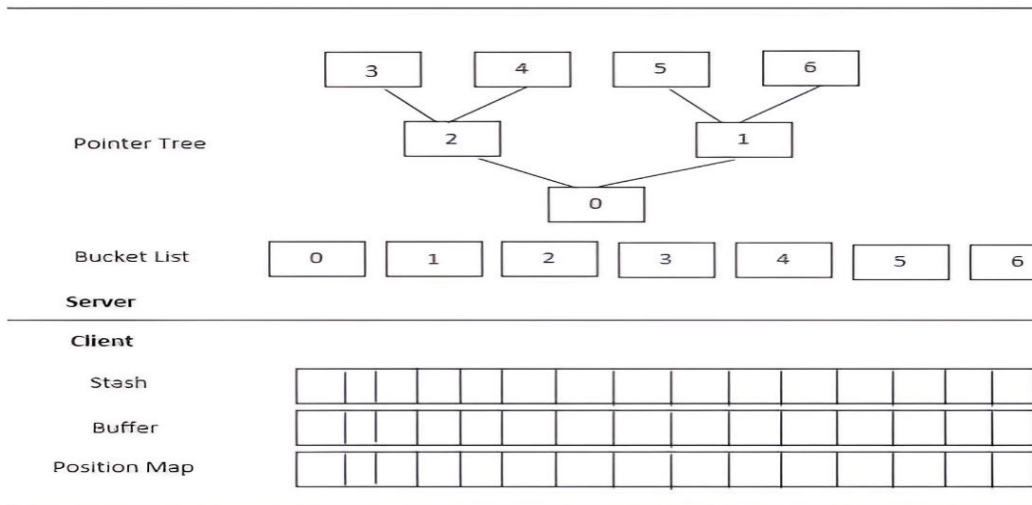


FIGURE 3. Data structure of cycle ORAM.

TABLE 1. Ring ORAM symbols used in algorithm.

Symbol	Description
N	Different no. of real blocks at server node
L	Tree depth
Z	Maximum no. of real blocks in the bucket
A	Evict function rate
B	Block size in bit
$P(l)$	Path l
$P(l, i)$	Level of bucket
$P(l, i, j)$	j^{th} slot in the bucket

1) DATA STRUCTURE

A binary tree serves as the server-side data structure. A bucket is present at each node of the tree, and each bucket is made up of many $Z + S$ slots. These slots serve as storage for blocks, each slot holding a single block. Blocks come in two varieties: genuine and fake. Additionally, a bucket includes certain information, such as the offset of the actual block. In client side, there are different types of data structures: position map, stash, and key-related data. Blocks that are not yet been written to the server are stored in stash. Each entry in the key-value dictionary is formatted as $IDX : DT$, where IDX is the index of the actual block and DT is the block’s payload. The position map is likewise a dictionary, with each entry constructed as $IDX : l$, where l is the block’s mapped route and IDX is the same as above. The information about keys includes encryption keys.

2) OPERATIONS

Three operations are available in a tree-based ORAM: Read and Remove, Evict & Reshuffle, Add Stash.

- Read and Remove: The client searches for the map of position to identify the route that the block has mapped when it needs to access a block. It then moves down the route, reading a block from each bucket. All other blocks

are dummy except from the target block. The metadata determines the offset in each bucket.

- Add Stash: In a read operation, the block obtained from the server is added to stash. Also stored into stash are the blocks in the eviction route or reshuffle bucket when eviction or reshuffle occurs.
- Evict and Reshuffle: In Ring ORAM, PathEvict and BucketReshuffle respectively handle eviction and reshuffle. PathEvict selects a route to evict at each access to A in accordance with the Reverse-lexicographic order. The path’s buckets are securely read out one at a time. Blocks from the cache are then eagerly poured into buckets from leaf to root, ensuring that they are pushed as far into the tree as possible. Reshuffle a bucket using BucketReshuffle if its counter is greater than S . With PathEvict, the read and write processes are identical.

3) CYCLE ORAM

The data structure and access procedure for Cycle ORAM are thoroughly explained in this section.

Data Structure: Figure 3 is an illustration of the Cycle ORAM data structure. It comprises of two major data structures on the server side: a bucket list that stores the payload of the buckets and a pointer tree that stores the pointers to each bucket. The indexes from the bucket list are used to initialize the pointer tree. In accordance with the Breadth First Search (BFS) algorithm, each node in a pointer tree is filled with a bucket list index one at a time, for example, the root hub is loaded up with record 0, its children are loaded with indexes 1 and 2, and so on. According to logic, it regards a bucket as being at the same level as its pointer, therefore in figure 1, it is mentioned that bucket 0 is at level 0 and bucket 1 is at level 1. Buffer, Stash and Position Map are the three basic data structures on the client side. The blocks in the bucket have not been written out. The blocks are kept in removal or rearrangement in the buffer. Position map items

have the format $IDX : (l, ORDER)$, where IDX is the block's index and $(l, ORDER)$ is the location of the node in the pointer tree that links to the bucket block that block IDX is located in. Specifically, l denotes the node's level and $ORDER$ denotes its position inside level l . With the new form, it will be simple to adjust l and $ORDER$ to reflect the location after Cyclical Shift.

Access Protocol: Algorithm 1 is an illustration of the Cycle ORAM access protocol. The pace of Bucket Reshuffle is indicated by the addition of the notation w . Round and pathways are two permanent variables that are used to start PathEvict and BucketReshuffle. Additionally, the collection of pathways used in BucketReshuffle is stored in $paths$. To locate the block, the user initially searches for the map of position. The block is then read using the ReadPath method. If the task is a read, the chunk is immediately returned; otherwise, the block's payload is changed by the argument \overline{DT} . The read block is tucked away in $stash$. In order to write blocks out according to the two aforementioned factors, refreshment operations like eviction and rearrangement are ultimately carried out.

Read Path: Cycle ORAM position map only maintains the location of the node that carries the target bucket's pointer. It chooses at random a route through the node that is the target of ReadPath in the pointer tree. The subsequent steps are almost same when using Ring ORAM. First, the client reads the information in the buckets along the target route to get the offsets of the target block or a valid fake block. Other than the target block, all of the other blocks being read out are dummy. The associated slot is invalidated when a block has been accessed to prevent further access. In the end, each bucket's number along the trail is raised by 1. A case where the goal route is "000", the buckets 0, 1, and 3 are accessible based on pointers in the route. At the conclusion, the three buckets' counts are each raised by 1.

Algorithm 1 Read (IDX, OPR, DT')

Variables: $r \leftarrow 0, paths \leftarrow \emptyset$
 $Pos \leftarrow PM[IDX]$
 $l \leftarrow \text{select a path randomly going through position}$
 $dataset \leftarrow RPath(IDX, l)$
 If $OPR = \text{read}$, then
 Return DT to consumer
 else
 $DT \leftarrow \overline{DT'}$
 $Stash \leftarrow Stash \cup (IDX, DT)$
 $r \leftarrow r + 1 \bmod A, paths \leftarrow paths \cup l$
 if $r = 0$ then
 Evictpath()
 if $paths.size = w$ then
 BucketReshuffle(paths), $paths \leftarrow \emptyset$

Path Evict: Every A access triggers the execution of PathEvict. In the meanwhile, the target route is selected using reverse lexicographic order. Algorithm 2 displays

the PathEvict procedure. To make things simple, it is described that the bucket directed by node P using the form $BucketP(l, i)$. First, the server executes CyclicalShift to the pointer tree. The pointer in each node is cyclically moved from the root to the leaf level. After cyclic shift the position map is revised. The location of a certain block is changed to the node in the next higher level along the route. Additionally, the client runs RootEvict. The bucket that the root node referred to is located first. After that, ReadBucket is used. The buffer is filled with a secure readout of all the Z blocks that have never been accessed previously. Blocks are then written out by using WriteBucket after that. The buffer-stored z blocks are first ready for writing out. In order to fill the bucket, a certain quantity of $Z - z$ blocks must be selected from the cache. Since the root bucket serves as the eviction object in RootEvict, any block in $stash$ may be selected as a candidate for the bucket. The last step involves writing down the genuine blocks that were selected from the contenders after being randomly permuted with fake blocks. These blocks' positions are modified after writing to $(0, 0)$.

Algorithm 2 Evictpath (IDX, OPR, DT')

Global variables: V set to 0
 $l \leftarrow \text{bitReverse}(V \bmod 2^{L+1} - 1)$
 $V \leftarrow V + 1$
 CyclicalShift(l)
 Evictroot(l)
 def CyclicalShift(l):
 for $i = 0$ to L do
 Shift pointer in $P(l, i)$ to $P(l, (i + 1) \bmod L)$
 end
 For IDX in $BPointer(l, i)$ do
 $pos \leftarrow pos[idx], l' \leftarrow (l + 1) \bmod L$
 $pos[idx] \leftarrow (l', 2pos.ord + \text{int}(\text{bit}(l)[i]))$
 End
 def Evictroot(l):
 AccessBucket($BP(0, 0)$)
 PushBucket($BP(0, 0), stash, buffer$)
 Bucket Reshuffle:

Every w access triggers the execution of BucketReshuffle. A bucket gets reshuffled in BucketReshuffle in the w pathways previously accessible whenever its counter hits S . Blocks should be moved as many times as possible to different pathways in order to create unpredictability. Onion ORAM shuffles a bucket among its offspring. Here, this is built upon and developed this notion. The offspring of the reshuffled bucket may not be involved since BucketReshuffle uses dispersed buckets. This technique offers to reorder the buckets so long as they are on the left or right side, whether they belong to a descendant, a sibling, or a descendant of a sibling. This technique sets the buffer's capacity to 3 in order to rearrange to both the left and right sides. Additionally, parameter w affects the randomization of blocks. This section provides definitions key terms used in the article as well as

Algorithm 3 Reshufflebucket(paths)

```

bl ← bucketlist ordered by evict time in paths
count ← 0
For BP (l, i) in bl do
    If BP(l, i).cnt = S then
        Buffer ∪ AccessBucket(BP(l, i))
        count ← count + 1
    If count ≥ 3 then
        Reshuffle(Buffer)
End
def Reshuffle(Buffer):
    leftdest ← discover left terminus of Buffer[0]
    rightdest ← discover right terminus of Buffer[0]
    blocks ←
    Randomize(BP(l, i), leftdest, rightdest)
    Pushbucket(BP(l, i), stash, blocks)
    Remove(Buffer[0])
def Randomize(BP(l, i), leftdest, rightdest):
    l ← level difference between BP(l, i) and leftdest
    lblast ← get blocks in ldest with  $2^{-l}$ 
    Permute lblast with blocks in BP(l, i) randomly
    l' ← level difference between BP(l, i) and
rightdest
    rblast ← get blocks in rightdest with  $2^{-l'}$ 
    Permute rblast with blocks in BP(l, i) randomly
    return lblast ∪ rblast

```

an overview of Federated Oblivious RAM (FedORAM) [47] system architecture and threat model.

4) FedORAM

This development of many federated ORAM models was driven by the idea of a performance-security trade-off. To compare them in terms of security and speed, author came up with two distinct structures.

- Strong federated structure: This federated scheme ensures that both clients and servers adhere to the original ORAM concept in full.
- Weak federated structure: A federate system that, by design, is intended to offer less security contrary to the leakage of metadata than the strong federated structure. But its goal is to go more quickly. This technique is not the first to explore such a federation outside of the setting of ORAM, with the email system being the most well-known example. A federation-based system provides various benefits over a traditional client-server design, despite certain well-known downsides including spam detection problems. As a server may be withdrawn without disrupting the federation as a whole, it offers fault tolerance. By default, it offers safety since only the source and destination servers may gather data. Additionally, the user often selects the source server, making it more reliable than other servers. The federation allows several service providers to simply add and remove servers without

having to establish mutual confidence. FedORAM is a novel federated architecture, as seen in definition 1, which necessitates new server types to perform diverse responsibilities while transmitting messages.

Definition 1 (Servers):

- a) Entrance server S_E : The entrance server is the initial server that user-sent data will encounter. Within the federation, it is the sole interface to which an end operator has access.
- b) Destination server S_D : This is the last server with which a user's data will interface. The destination user will unknowingly get data from this server.
- c) Third party server: Any other federation server that is not the transfer server or not the last server but may function as an intermediate in a data exchange.

5) FEDERATED OBLIVIOUSNESS

Definition 2 (Obliviousness): A technique is deemed oblivious if the data node is unable to differentiate between two accesses that include any sort of user authorization.

Definition 3 (Weak Obliviousness): A technique is weakly federation oblivious in case all of the following requirements are met. The kind of access is known to the entry/destination server. A third-party server at the destination cannot tell the difference between two accesses that include any action that the users are authorized to do on the server. A third-party server and last server are not able to be distinguished by an entrance server. An entry server might be identified by an outside attacker.

Definition 4: A scheme is highly federated oblivious if and only if all of the following requirements are met. The kind of access is known to the entry/destination server. With the exception of push/pull access, servers are unable to discriminate between any two accesses that include any form of action that users are authorized to carry out on the server. A third-party server and a destination server cannot be distinguished by an entrance server. An entrance server and a third-party server are indistinguishable to a destination server. Any server is indistinguishable to an outside attacker.

- 1) **Metadata:** There is need to consider the various sorts of metadata that can be spilled by a server. All information has a particular organization yet it can have various unmistakable arrangements of metadata, each depicting an alternate property of the information. Each unique approach to portraying information is known as a 'sort' of metadata.

WEAK federation structure: Figure 4 shows the general building of the Weak federation model when a user C_1 sends a message to another user in the server S_D . Weak federation can be summed up as follows.

- A set of s autonomous servers S_i , $1 \leq i \leq s$.
- A root machine S_R .
- Every server has its individual set of users C_i , $1 \leq i \leq c$. Each user as a distinctive ID in format $C_i@S_i$.
- Algorithms as mentioned in definition 5.

Definition 5 (Weak FedORAM Scheme): A Weak federation technique is a set of the following algorithms.

$M \leftarrow Push(id, m, user)$ Send message to a user on a server. Create and send the message block to be moved to the final server utilizing OT from the entrance server.

$m \leftarrow Pull(id)$ New message can be pulled from the client's entry server.

$Serve(id)$ Transfer a user message to the federation using OT.

$Evict(leaf, forget)$ Initialization of the eviction algorithm on the entrance server.

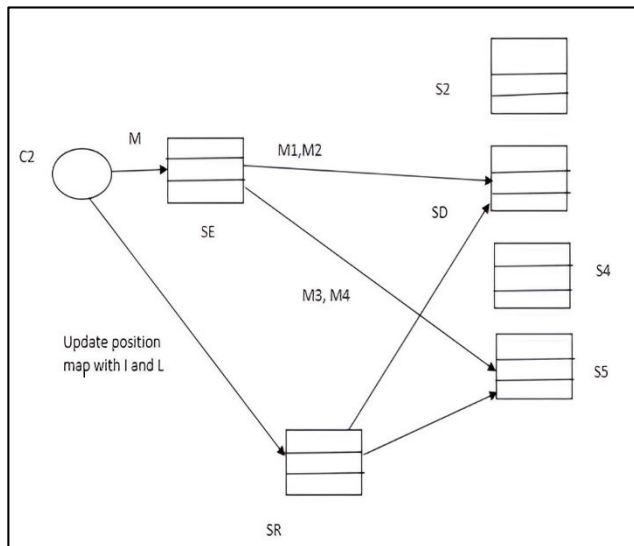


FIGURE 4. Overview of the weak FedORAM scheme.

Figure 4 illustrates how a client $C1$ creates M while communicating with another client in S_D . This message block is then sent by $C1$ to its entry server S_E . S_E is aware of the servers at which it must transmit data (S_D and S_d). Thus, using $M1$ and $M2$, S_E executes two OT transactions to S_D and S_d . Both the genuine and fake destination servers get their $M1$ and $M2$ message block sub-parts. Both servers may match their own server ids to the message once it has been decrypted. The server is a genuine final server and publishes messages to its interior root node if they are equal. The server disregards the transaction in the case if they are equal.

Position Map: All ORAM models must include internal metadata management because metadata enables actions on distant messages. This technique uses Position Map matrix that is kept on the root server similarly to previous ORAM systems. This matrix connects the allocated leaf on the final server to the virtual id ID of a message (actual or fake).

OT Map: Similar to the PositionMap, FedORAM requires root server storage for an unaware transfer matrix. It establishes a connection between a message's virtual id ID and the unaware transfer identifier b , which indicates whether or not the actual message's real id has been transmitted. B is encrypted to prevent anybody other than the target server from reading it. An ID for b is provided by $OTMAP[id]$.

Working of FedORAM: In the primary Strong FedORAM algorithms, on clients, Pull and Push algorithms are both used. Push with the Push algorithm, the client creates the message block M and encodes it with the recipient's public key. Additionally, it creates the OT selection identification for S_D and S_d as well as the leaf of the destination server. The entrance server S_E will proxy M when it is sent there. The client uses the same algorithm as in Path ORAM to read and decode the message using Pull algorithm.

This system differs significantly from a traditional ORAM system in that it relies mostly on through relations to servers within the alliance and does not call for additional expensive computing resources. However, there is a security price for this effectiveness. While it can prevent certain information leaks, it is still vulnerable to access pattern attacks that might reveal the sender and recipient but not the entry or destination server. Making ensuring that all communications are sent via the same server, independent of the server from which they originated, will solve the privacy problem. In order to address this issue, we suggest Strong FedORAM, which extends the famous tree-based route oblivious data structure on a separate server to a federation of servers.

Strong federation: When a client $C1$ transmits a message to another user in the server S_D , Figure 5 depicts the overall architecture of the Strong FedORAM model. Following is a summary of a strong FedORAM.

User transfer a text data M to a server node using the $push(id, m, user)$ command. It creates and transmits the message block using OT from the entrance server to the final server. User fetches new text data from the entrance server using the pull command. The message $Proxy(id, leaf)$ is a proxy sent by a federation server to a leaf. Removal from a situation ($leaf, forget$) start the entry server's eviction mechanism.

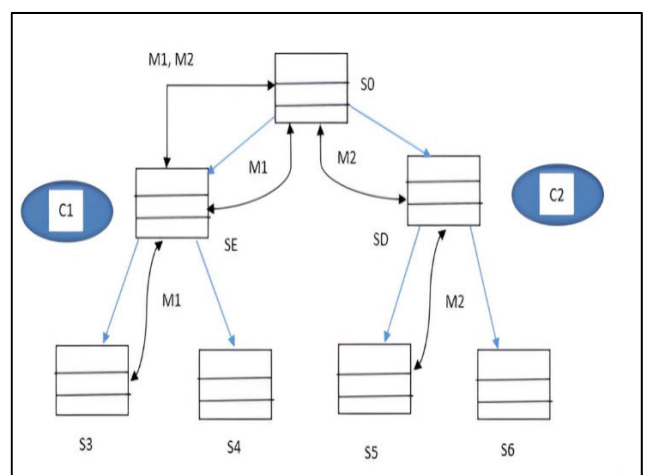


FIGURE 5. Overview of strong FedORAM scheme.

Federation Structure: This technique suggests a federation structure that resembles a binary tree. In the federation, every node serves as a server. Similar to Path ORAM, each

server may store n or more messages in its own oblivious data structure. These data structures are all separate from one another. Each message received from a user to an entry server is connected to a route from the root server to a leaf l , just as in Path ORAM: (l). The user chooses l such that P is the target server (l).

Working of Strong FedORAM:

Figure 5 shows working of strong FedORAM. In strong FedORAM only one algorithm has to be executed on a federation server, and that is the proxy algorithm. Because Pull and Push are client-side algorithms. Similar to Weak federation. Pull uses the same method. Algorithm 1 and Push Algorithm are comparable. The message block M , federation and local leaves, as well as the OT selection identification for each server in the leaves, are all generated by the client. The entrance server S_E will proxy M when it is sent there. Proxy A server may unknowingly transport a message into the federation using the proxy algorithm. S stands for the server id used to perform the algorithm. The approach fetches the interior storage leaf from the metadata matrix first, independent of the server type. Even though it is only required for the final server, if the network were to be queried before the server has determined its status, the federation would learn about it. The message block is then locally decrypted into a tuple (M_l, S_l) , where M_l represents the next message block and S_l represents a server node id. The OT transmit and receive routines are $OTSend$ and $OTReceive()$.

Xu et al. [48] proposed forward searchable encryption scheme. The issue of encrypted data retrieval in a cloud computing environment may be solved very well using Searchable Symmetric Encryption (SSE), which also helps to protect users' privacy. Recent research indicates that certain attacks offer a considerable security risk to SSE, and forward privacy can successfully fend off these assaults, making forward privacy an essential part of the SSE system. Both ORAM-based and Bost-based forward privacy SSE solutions are now available. Despite being simple, the former has a lot of communication overhead and poor dynamic update performance. The latter depends on asymmetric encryption primitives even if it is superior to the former. In this research, they provided forward searchable encryption scheme, a symmetric encryption-based dynamic efficient forward privacy system. It uses deletion lists to determine the final state of a single file that has been repeatedly uploaded and removed in addition to pseudo-random permutation for forward privacy to accomplish dynamic data updating. This research offers a straightforward, adaptable approach that uses little more space and significantly enhances updating performance. Real-time data updating is also conceivable. The correctness of the above strategy is then examined using the Enron email data collection.

Hoang et al. [49] proposed multi-user oblivious storage. Using multi-user oblivious storage, users may access their shared data in the cloud while preserving data confidentiality and access pattern oblivion. The safest and most effective oblivious storage systems give priority to increasing network

bandwidth while supporting concurrent accesses through a reliable proxy. Because the proxy uses a standard ORAM protocol via the network, its performance is limited by the latency and network capacity. Furthermore, there hasn't been enough research done on several crucial aspects of these proxy environments, such access control and security against active attackers. In this work, they present MOSE, a multi-user oblivious storage system that is effective and has several desired security features. Their core concept is to reduce the network bottleneck of proxy-based systems by running proxy logic on an untrusted storage server using a secure enclave, such Intel SGX. This technique addresses a number of technical design obstacles, including memory constraints, side-channel dangers, and scalability problems, while enabling proxy logic in the secure enclave. This technique provides a formal security model and analysis for safe enclave multi-user ORAM with access control. It has been enhanced to handle concurrent requests more quickly. Its performance was evaluated using standard hardware. Their testing confirmed MOSE's effectiveness, showing that it reaches throughput levels nearly two orders of magnitude greater than the current cutting-edge proxy-based architecture and that its performance scales according to system resources.

Zhao et al. [50] proposed Efficient ORAM (ETORAM) for securing metadata. They used Function Secret Sharing (FSS) and optimized it for reducing computation overhead. When scaling a distributed ORAM to a two-party safe computation, the quantity of Pseudo-random Generator (PRG) calls required for the creation and evaluation of a distributed point function in this efficient ORAM are $O(\log n)$ and $O(n)$, respectively. In this distributed ORAM system, where n is number of data blocks, z is the length of shares of outputs in distributed point function and s is the secure parameter, PRG calls are reduced to $O(\log(zn/\lambda))$ for generation where λ is secure parameter. On a technical level, they first apply the FSS optimization to the context of ORAM for safe computing, which causes early termination for functions with tiny output groups. Next, they developed the ETORAM strategy to capitalize on the high efficiency gained by early termination. In ETORAM, each data block has its own access counter. During a write operation, the DPF outputs covertly update the access counter while masking a random string changes the data block in a private manner. In reality, depending on access rates, z varies from 1 to $\log n$ in sequential mode and from 3 to $\log n$ in random mode when the total number of accesses is n .

Li et al. [21] proposed privacy preserving access scheme for securing big data in the cloud storage. They examined a network with 100 buckets and 10 servers to evaluate the performance of the online technique, which is called an online heuristic, for dynamic deployment. They change each bucket's access rate at random to achieve this. This technique also assesses the results of an alternate online strategy called online opt. Depending on the optimization framework, this approach modifies the location of the data. After raising the rebalancing threshold from 100 to 400, they analyze

the maximum access demand for each method. Online opt frequently outperforms our heuristic method, although the difference in performance is not particularly large. By raising the value of from 100 to 400, they can assess the effectiveness of both strategies. When it comes to data replacement, the suggested online heuristic generates much less network traffic than online opt. This is because online opt ignores the location of the current data and does global optimization for load rebalancing.

Al-saleh and Belghith [51] proposed reduced bucket size ORAM. The bucket size of the root (radix) is large, but the fixed length of each subsequent bucket in the tree is more modest. A thorough examination of root bucket occupancy is conducted in order to arrive at a closed-form solution for the necessary root bucket size that has a low failure probability. Using a single testing platform, the effectiveness of the R-Path ORAM is examined and contrasted with that of the standard Path ORAM. Experiments clearly show that compared to Path ORAM, R-Path ORAM offers significantly less server storage and a faster average response time. In addition, this technique offers a background eviction strategy that aims to shorten the root bucket in order to avoid a long-term system failure. The above-mentioned two-way eviction technique was successfully used to reduce the size of the root bucket despite causing a small amount of overhead, as demonstrated by experiments performed on the unified platform.

Hoang et al. [52] proposed efficient oblivious data framework for cloud. Database-as-a-service (DBaaS) enables users to manage and store structured data remotely in the cloud. DBaaS has significant privacy concerns despite its advantages. Existing encryption techniques can aid in reducing privacy issues, but they still let data slip through access patterns that are vulnerable to statistical inference attacks. Despite the fact that the ORAM can stop these leaks, recent investigations have shown that there are significant barriers to integrating ORAM into databases. In particular, direct ORAM use on databases is expensive and only supports a very limited set of query capabilities. This technique suggests two brand-new oblivious data structures, the Oblivious Matrix Structure (OMAT) and the Oblivious Tree Structure (OTREE), which make it possible for tree-based ORAM to be more effectively incorporated into database systems while supporting a wider range of query functions. Because OMAT uses specific ORAM packing strategies for table structures, it is more efficient and supports a wider variety of query types than other frameworks. The use of oblivious conditional queries on tree-indexed databases is now more effective than it was before with the help of OTREE. They carefully put suggested tactics into practice and, using a variety of metrics on a real cloud database, compared their performance to that of their modern equivalents.

Xu et al. [53] suggested attribute-based retrieval control scheme with secure deduplication functionality. Massive amounts of sensitive personal data are now simpler to produce, collect, and handle thanks to recent developments in information technology. Cloud storage services are quickly

becoming the best paradigm for enabling such requests from different domains. These cloud centric solutions provide added safety and confidentiality issues when working with outsourced data, such how to retain granular access control over data that is kept in the cloud. In this research, they proposed a comprehensive, to protect the security and privacy of users' data that is contracted out to and maintained by a cloud service provider, a user-centric attribute-based access control system must be privacy-preserving (CSP). The system's fundamental component is a revolutionary privacy-protecting reversible CP-ABE method. They suggested an enhanced Path-ORAM access protocol that might also shield users' access patterns from prying eyes in order to enable sophisticated access control capabilities including write access to encoded data and a policy that protects privacy adjustments. Additionally, they recommend an integrated secure deduplication strategy for CSPs to improve storage effectiveness while preserving data privacy. At last, they assess the proposed framework's performance and security and contrast it with other available options.

Tian et al. [54] proposed scheme for storing data, which is based on locality. A growing concern is how to prevent data leakage from cloud access patterns as cloud storage becomes more and more widespread. Unaware RAM is made available to do this. It is meant for the memory system, with much earlier research concentrating on improving main memory performance. ORAM is currently known as Oblivious Data Storage since it has lately been extended to the cloud. Modern oblivious data storage system Tao Store greatly reduces mean response time by combining synchronous I/O and ORAM technology. They found that there is a significant locality in user accesses. But this wasn't considered in previous Oblivious Storage studies. In this paper, they introduced Loco-Store, a system for oblivious data storage. In Loco-Store, they provided a special stash controller technique that may dynamically aggregate crucial blocks across unaware I/O operations. Additionally, they provided a locality-based eviction mechanism to uphold the security promise. Theoretical evidence demonstrates that the strategy complies with ORAM's notion of security. Finally, they construct a prototype and conduct extensive testing using data from the actual world. The findings show that Loco-Store may decrease network bandwidth use by 39.19% while reducing total access time by 26.17%.

Liu et al. [55] proposed multi cloud oblivious storage technique. Data privacy cannot be guaranteed by encryption alone since access patterns might reveal certain sensitive information. For large storage and communication/processing overhead, the solution, ORAM, is still far from being practical. It was recommended to employ non-colluding clouds to transfer client processing and client-cloud communication to the clouds in order to decrease them. The multi-cloud ORAM that was proposed lowered client-cloud bandwidth to $O(1)$ and did away with the majority of client processing. In this study, they used "two-layer encryption" and "disconnected ORAM operation" to further decrease these overheads. Compared

to earlier methods, experiments show that this technique significantly decreases the cache size from GB/MB to KB level, with around 2-3 times quicker response times and 20% bandwidth savings for clouds.

Huang et al. [56] proposed thin ORAM for oblivious data retrieval. For applications that need to conceal access patterns, ORAM is essential. However, since the majority of current ORAM implementations are so costly, they cannot be used in lightweight devices like fog nodes. The goal of this research was to examine the usage of expensive ORAM to protect the access patterns of lightweight devices. They also suggested “ThinORAM,” an ORAM system that supports thin clients in non-colluding clouds. This method removes the need for sophisticated calculations on the client side and just needs $O(1)$ communication costs with a reasonable response time. They also showed how to utilize ThinORAM to accomplish oblivious data access at a low client cost in a fog computing environment. Experiments show that this technique can remove the majority of client storage, reduce cloud-cloud bandwidth by $2X$, and provide a $2X$ quicker reaction time when compared to the best technique for reducing client-side overheads.

The application of an encryption technique to user messages might protect the information they contain, but this alone is not enough to solve concerns about metadata. In particular, the information describing the access pattern will cause damage to the full record if it is released. An attacker can gain direct access to private data if they are able to correctly guess the access pattern. There is a possibility that cloud attackers, in addition to other types of attackers, could catch the exposed path of access and exploit it [57]. Internet criminals are always developing new methods to gain unauthorized access to information and use it to their commercial benefit. Therefore, there is a requirement ORAM which enables customers for accessing data from remote servers in a covert manner without revealing the data retrieval path. This strategy is also known as a blind random access machine. The path of the real data access made by the user in ORAM is not the same as the path of the physical location. In essence, a great number of researchers have contributed to ORAM in order to realize its primary goal. Researchers have revised the fundamental model of ORAM in order to enhance ORAM’s overall performance.

Researchers have attempted to simplify ORAM in order to make it more functional so that a dynamic approach may be developed to address the security risks posed to sensitive information [58]. This is because ORAM is limited in relations with complexity. In latest years, the Path ORAM techniques have been utilized for the purpose of enhancing security. These techniques were estimated by Stefanov et al. [59]. Path ORAM is responsible for the process of storing the information blocks within the binary tree construction, which may include numerous leaf nodes like buckets. All buckets that make up a tree has a predetermined and unchanging block numbers, which is represented by the

letter z . After the tree has been initialised, the range of values for the leaf bucket is defined from zero to $n - 1$, while the range of values for the random tag or location for each block is 0 to N . In addition, it has a single, rather modest stash zone that can temporarily store a large number of blocks. If a chunk has the tag a , then it will be stored in the cache or somewhere with the path that leads from the root of the plant to the a^{th} leaf server, as determined by the constant for the tree.

Table 2 discusses different ORAM techniques with their advantages and disadvantages.

C. DATA RECOVERY TECHNIQUES

For data recovery, Zhang et al. [60] introduced the Cauchy Coding (CaCo) technique. The main goal of this approach is to find the lowest schedule feasible for a Cauchy matrix given a redundancy configuration of k , m , and w . They proposed CaCo, a coding method that includes all presently existing matrix and scheduling heuristics and, as a consequence, is capable of finding an appropriate solution for data coding in a cloud storage system within the limits of the state-of-the-art.

1) CHOOSING A MATCHED CODING SCHEME

The main goal of CaCo is to choose a Cauchy matrix and one of its schedules for a redundancy configuration (k, m, w) whose size is to be reduced. CaCo is composed of the following four stages.

- **Cauchy matrix generation:** CaCo creates a series of Cauchy matrices known as $Seies_m = \{m_0, m_1, m_2, \dots, m_{p-1}\}$ by using p distinct heuristics.
- **Building schedules for every matrix:** CaCo generates a collection of schedules using a number of algorithms., designated as $S_{s,i} = \{S_{i,0}, S_{i,1}, \dots, S_{i,q-1}\}$ for each matrix m_i ($0 \leq i < p$) in the set S_m produced in the first phase.
- **Choosing each matrix’s locally optimum schedule:** CaCo chooses the smallest schedule from the set $S_{s,i}$ abbreviated S_i , for each matrix m_i ($0 \leq i < p$) in the set S_m . After this stage, we get a collection of matrices and their shortest schedules, indicated as $S = \{(m_0, s_0), (m_1, s_1), \dots, (m_{p-1}, s_{p-1})\}$.

Choosing the globally optimum solution: The smallest schedule, s_j ($0 \leq j < p$) is then chosen by CaCo from the collection of $\{s_0, s_1, \dots, s_{p-1}\}$ In order to get the best encoding performance, $\{m_j, s_j\}$ is the globally ideal approach.

2) GENERATING CAUCHY MATRICES

It is a combinatorial issue to choose the optimal Cauchy matrix using the enumeration approach. It is impractical to count the size of the matrices that must be generated given a redundancy configuration of 10, 6, and 8 since they may reach 1029. Even now, it is impossible to say which of the matrices will result in the best schedules. They only choose a select few of these for scheduling in the CaCo technique. They often choose the binary Cauchy matrices with fewer ones when taking performance into account. To create

TABLE 2. Comparison of various ORAM techniques.

Sr. No.	Author	Technique	Advantages	Disadvantages
1	Zhao et al. [46]	Cycle ORAM	It improves the transfer speed	Due to a cyclical movement, the bucket that was expected to give the most storage for the blocks in stash has been traded to the root.
2	Pujol et al. [47]	Federated ORAM	Provides strong obliviousness with strong FedORAM	It has a higher overhead cost, is more sensitive to federation size, and suffers from latency as the number of servers in the federation grows.
3	Xu et al. [48]	Forward searchable encryption scheme	Uses a delete list to determine the ultimate state of the same file uploaded and removed repeatedly, allowing for dynamic data updates.	The client's storage space must be reduced.
4	Hoang et al. [49]	Multi-user oblivious storage	MOSE has proved that adopting a secure enclave may solve the bandwidth barrier while also achieving scalability, access control, and robustness against active adversaries.	Not scalable for use at production
5	Zhao et al. [50]	Efficient ORAM	Overhead is decreased, which is dominated by the amount of PRG calls in the production and assessment of a DPF	-
6	Li et al. [21]	Privacy preserving access scheme	Promising in terms of obscuring access privacy in cloud storage	To minimize the ORAM's complexity
7	Al-saleh and Belghith [51]	Reduced bucket size ORAM	It reduces the root bucket size and avoid system failure	To determine the best background eviction approach for reducing the root bucket size while incurring the least amount of overhead in terms of the proportion of new fake requests.
8	Hoang et al. [52]	Efficient oblivious data framework	OMAT and OTREE are new oblivious data structures that allow tree-based ORAM to be incorporated into database systems more efficiently and with different query features provided.	Time complexity increases
9	Xu et al. [53]	Attribute-based retrieval control scheme with secure deduplication functionality	The integrated secure deduplication solution enhances cloud service providers' storage efficiency.	-
10	Tian et al. [54]	Scheme for storing data based on locality	A unique stash controller technique dynamically groups important blocks during oblivious I/O procedures, improving system efficiency.	-

a collection of Cauchy matrices, $S_m = \{m_0, m_1, \dots, m_{p-1}\}$ they used several of the algorithms including Original, Cauchy Good, and Optimizing Cauchy. This scheme creates a Greedy heuristic to broaden the variety of the Cauchy matrices. The following is a description of our greedy heuristic for creating a light Cauchy matrix.

- Building the ONES matrix: In the beginning, CaCo creates a matrix called ONES, whose element j is defined as the quantity of ones in the binary matrix $M(\frac{1}{i+j})$.
- Picking the bare minimum component. Second, CaCo selects the matrix ONES's smallest element. We initialize X to be $\{x_1\}$ and Y to be y_1 if the element is (x_1, y_1) .
- Finding the Y-set. CaCo selects the top $k - 1$ minimums from row x_1 in addition to the element (x_1, y_1) . The result is $Y = \{y_1, y_2, \dots, y_k\}$ after CaCo adds the matching $k - 1$ column numbers to Y .
- Figuring out the set X : CaCo determines $C_r = \sum_{y \in Y} (r, y)$ for each row r ($r \neq x_1$) in the matrix ONES. The set $\{C_r, r \neq x_1\}$ is obtained by selecting the top $m - 1$ minimums from the set $X = \{x_1, x_2, \dots, x_m\}$ and adding the appropriate $m - 1$ row numbers to X .

3) CONSTRUCTING SCHEDULES FOR EACH MATRIX

It calls the function schedule with the arguments schedule (*int k, int m, int w, and int * matrix*) for each matrix m_i ($0 \leq i \leq p$) in the set S_m to execute q heuristics in the function. In this way, a collection of schedules S is obtained. If a useful heuristic for scheduling later emerges, we may include it into the function do schedule.

4) SELECTING LOCALLY OPTIMAL SCHEDULE FOR EACH MATRIX

This scheme chose the shortest schedule from the set $S_{s,i}$, represented as s_i , for each matrix m_i ($0 \leq i \leq p$) in the set S_m . As a result, we receive a set of matrices and their shortest schedules, designated as $S = \{(m_0, s_0), (m_1, s_1), \dots, (m_{p-1}, s_{p-1})\}$. Data may be encoded in an order of XORs specified by s_i for m_i in the set S_m . As a result, there is no longer a direct correlation between XOR operation timings and matrix density. Due to scheduling, the bottom limit of the matrix's number of ones is therefore excluded, which considerably increases speed.

5) CHOOSING THE WORLD'S BEST SOLUTION

They chose the Cauchy matrix and schedule combinations that have the smallest schedule from the group, which includes $(m_0, s_0), (m_1, s_1), \dots, (m_{p-1}, s_{p-1})$. Based on this, they often choose the one that has the fewest ones in the matrix to be (m_{best}, s_{best}) for greater performance. S_{best} may be used to encode data after it has been chosen.

6) USING A SELECTED SCHEME FOR ENCODING AND DECODING

This scheme chooses and save each redundancy configuration once and for all since it is coupled with a mix of m_{best} and S_{best} . Every time it encodes data, it simply grabs the best value from memory. As a result, calculation of scheduling requires less time now, and data encoding is significantly more efficient as a result. It can restore the original data by decoding the remaining working blocks if any of the m data or coding blocks fail. It may first create a matrix md for decoding based on the selected matrix m_{best} for data encoding and the unique circumstance of data corruption in the same group of k m blocks. Then, with the same strategy as above, it determines the best timetable for recovery of the data.

7) PARALLEL COMPUTING ACCELERATES SELECTION

CaCo is performed in parallel by dividing the computing work to specific nodes in a cluster in order to hasten the selection of coding schemes. The computations of Cauchy matrices and schedules are entirely independent for various methods. The distribution of the work of creating matrices and schedules to several processors would thus be possible and efficient.

1) Computational jobs are assigned by the coordinator: The parameters handed in, such as the number of workers and the redundancy configuration “ k ,” m ,” w ” are received by the coordinator. Following that, the coordinator provides the necessary parameters to the workers, including k, m, w, HM (heuristic for creating a Cauchy matrix), and others. It should be concerned with worker load distribution in this method.

2) Employees run heuristics to create schedules and matrices: workers receive the coordinator's message and use the HM heuristic to obtain a Cauchy matrix. Then, workers choose the locally optimum schedule for the matrix using various heuristics, and communicate the chosen combination of “ m_i, s_i ” to the coordinator.

3) The coordinator compiles the outcomes from the workers and chooses the best: The Coordinator gathers the combinations that the workers bring back, and from these chooses the best combination to be “ m_{best}, s_{best} ” by comparing the size of the schedules and the density of the binary Cauchy matrices. The majority of the time, CaCo is used with a cloud storage system that has a cluster of several nodes. Therefore, it can employ easily accessible computers to deploy distributed setups.

Algorithm: CaCo Write Operation

1) The Name Node receives a write request from the Client.

- 2) The Name Node assigns the Client a number of Data Nodes.
- 3) Write the blocks of data into Data Nodes.
- 4) Create a duplicate of the data, and then add it to the Data Queue.
- 5) Use the schedule CaCo has chosen to encode the data.
- 6) Insert the code blocks inside the Data Nodes.
- 7) Coding of data is complete.
- 8) Eliminate the data copies from Data Queue.

Depending on the intended balance of performance and fault tolerance, cloud storage systems always employ various redundancy configurations (i.e., “ k ,” “ m ,” and “ w ”). No matrix and schedule combination work optimally for all redundancy configurations because to the wide variation in the amount of XOR operations. In this article, they introduced CaCo, a novel method that takes into account all known matrix and schedule heuristics and can therefore determine the best coding scheme for a particular redundancy configuration within the realm of what is currently possible. CaCo's selection procedure has a manageable complexity and can be speed up via parallel computing. Additionally, it should be noted that the selecting procedure is final. The experimental findings show that CaCo beats the “Hadoop-EC” strategy by 26.68-40.18 percent in encoding time and by 38.4-52.83 percent in decoding time at the same time. Improving an erasure code's efficiency does not solely include minimizing XORs. Performance may be constrained by other code characteristics, such as the volume of data needed for recovery and degraded reads more so than CPU cost.

Bian et al. [61] proposed optimal weakly secure codes scheme for data recovery. In this technique, they presented an enhanced method for optimizing weak security based on PM-MSR codes, known as optimum weakly secure PM-MSR (OWSPM-MSR) codes.

- **Details of the scheme:** Data pre-processing is encoded using non-systematic RS code based on a Cauchy matrix in the encoding procedure. The suggested method pre-codes the data before using the PM-MSR coding algorithm to encrypt it. For the two encoding systems to have the best weak security, the generating matrix has to have certain characteristics. Data pre-processing and data encoding are the two procedures that make up the scheme.
- **Processing of raw data:** Encoding matrix generation: Set the coding parameters (n, k, α, β, d) in accordance with the node criteria, then produce at random two groups of parameters set x_i and y_i each of which has two elements that are unique. Next, create a Cauchy matrix C element with formula $\left(\frac{1}{x_i+y_i}\right)$. It is the data pre-processing generating matrix.
- **Data encoder:** A PM-MSR encoding strategy is used to encode the data \bar{F} produced by the preprocessing. The first step is the random generation of the parameter sets $\{x_i \in F_q | i = 1, 2, \dots, \alpha\}, \{y_i \in F_q | i = 1, 2, \dots, \alpha\}$ and $\{Z_i \in F_q | i = 1, 2, \dots, \alpha$ where any two members

in each set are distinct. In addition, the following three parameters are satisfied $z_i \cdot x_i = x_i, z_i \cdot y_i = y_i, i \in \{1, 2, \dots, \alpha\}$ then, as seen below, we may produce a diagonal matrix D and Cauchy matrix.

$$D = \begin{matrix} z_i & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & z_\alpha \end{matrix}$$

As a result, which is a Cauchy matrix, produces the PM-MSR encoding matrix. Then G_i is produced by encoding algorithm.

This work shows that it is possible to achieve $(2\alpha - l)$ -block security against an eavesdropper with parameter l ($l < k$) by evaluating the poor security of a Cauchy matrix-based PM-MSR coding scheme under various parameters. They demonstrated the best-case weak security of MSR codes, demonstrating that $(B - l\alpha)$ -block security is feasible. Based on this finding, they provided an enhanced PM-MSR coding scheme that maximizes weak security. The OWSPM-MSR codes system pre-codes the data using Cauchy-matrix-based RS codes before encoding it with PM-MSR codes also based on the Cauchy-matrix. The study shows that the method achieves optimum weak security since the generator matrix of these two coding schemes meets the requirement. With regard to the parameters k and m , which stand for file size, they used various schemes to encode and decode files. To compare the computational overhead, they also timed their encoding and decoding. Various techniques have been utilized to encrypt a 1MB file with k varying from 4 to 10. The findings show that the encoding times for the OWSPM-MSR and A-WSMSRC schemes were substantially greater than those for PM-MSR codes.

In the worst-case scenario, the OWSPM-encoding MSR's time was almost eight times more than that of the PMMSR codes scheme. The A-WSMSRC method took longer to encode than the other two schemes for all values of the parameters k and m . The encoding time gap specifically grew when k and m were increased. Due to the precoding step that the A-WSMSRC and OWSPM-MSR schemes add, which increases computational cost. The precoding procedure uses RS scheme, therefore the variation in encoding time is correlated with the value of B , which is equal to $k(k - 1)$. Pre-processing the data with an ANOT transform based on the Vandermonde matrix is comparable to encoding the data with Vandermonde-matrix based RS codes in the A-WSMSRC scheme. The Vandermonde-based RS coding approach has a large computational cost due to the difficulty of multiplication and division in a limited field. The Cauchy-matrix-based RS coding scheme is used in the OWSPM-MSR system. Addition and subtraction operations over finite fields can be realized by exclusive-OR (XOR) operations, and that multiplication and division can be converted into these operations as well. As a result, compared to the A-WSMSRC method, the OWSPM-MSR scheme has a lower computational complexity and takes less encoding time.

Tang and Cai [62] proposed the novel decoding method for erasure codes. It is a general decoding method that increases

the decoding efficiency for various erasure code types over various finite fields. By using a straightforward matrix transformation of the decoding transformation matrix, it is simple to get the linear combination of the failed symbols by other valid symbols. The suggested approach may avoid matrix inversion during decoding for RS codes over the binary extension field, significantly increasing the decoding efficiency. In order to increase the decoding performance for array codes over $GF(2)$, the suggested approach is still relevant. Additionally, compared to the traditional iteration method, the suggested method can effectively reduce the repair bandwidth for the array codes in the data reconstruction. The valid nodes engaged in the decoding process might be dynamically planned in order to reduce the load on the overworked storage nodes.

Tang and Zhang [63] suggested fast encoding and decoding technique for data recovery. RS codes are widely used in storage systems for failure recovery. In the majority of software implementations, RS codes are defined by a parity check matrix that is either a Vandermonde matrix or a Cauchy matrix padded with an identity. The complexity of the encoding can be reduced by selecting a Cauchy matrix with fewer '1's in its bit matrices or by employing the Reed-Muller (RM) transform in the Vandermonde matrix multiplication. This research offers two fresh tactics that outperform the ones already in use. In this technique in first strategy, they look at a number of finite field constructions to further lower the number of "1's in the Cauchy matrix's bit matrices, and they created a new searching method to identify the matrices with the fewest "1's. By joining an identity matrix and a parity check matrix in the shape of a Vandermonde matrix, their second technique generates RS codes. This method does away with the necessity for inverse erasure columns in the encoding and enables decoding to be done with less complicated formulae. The Vandermonde matrix must be produced using non-consecutive finite field elements in this uncommon RS coding formulation. The RM transform may be utilized in this situation to lessen the complexity of matrix multiplication. For a variety of code word lengths, the two proposed techniques outperform earlier work based on Cauchy matrix and Vandermonde matrix with RM transform by an average of 40% and 15%, respectively, across 4-erasure-correcting RS codes over $GF(28)$. Additionally, there has been a significant boost in decoding throughput.

Yu et al. [64] proposed the scheme which shows a rapid RS encoding using four to seven equality pictures. First, they demonstrated that it is possible to solve the RS code disorder using the RM change. This result is then used to determine the fast-encoding calculation. According to the analysis, the suggested strategy takes 3 XORs for each piece of information that is asymptotically nibbled, which is an improvement over earlier approaches. The reconstruction demonstrates that the suggested technique presents better than elective arrangements and becomes better as the code length increases. When the equality number is 5, the suggested method is twice as rapid as current data recovery methods.

Tai et al. [65] proposed scheme of two-dimensional RS code. Due to a lack of message location data, insert and delete (insdel) errors are problems with the correspondence framework's synchronization. Because of their ease of encoding, excellent structure, and low disentangling limit, RS codes have generated interest in the traditional environment. This interest extends to the insdel metric, where an erasure revising procedure may be provided via the Guruswami-Sudan unwinding approach. Despite this, there haven't been many studies on how well Reed-Solomon codes can fix internal errors. Express definitions of two families of 2-layered Reed-Solomon codes with internal error correction capabilities that asymptotically approach those provided by the Singleton bound are the article's main commitments. An asymptotic group of RS codes with an insdel error adjustment limit are produced by the main concept. The next step creates a collection of RS codes with fantastic indel blunder revision up to the code length. Both of the methods improve the newly discovered development of two-layered RS codes, which only offer logarithmic error correction capabilities.

Oh et al. [66] proposed erasure codes which are often employed in communication and storage frameworks. The employment of major restricted OR jobs in the encoding and disentangling procedures, which reduces computing complexity, is an essential component of display codes. Most extreme distance separable (MDS) display codes that employ Cauchy grids over constrained fields, round change frameworks, and circular Cauchy lattices, independently, include Cauchy RS codes, Rabin-like codes, and circular Cauchy codes. These codes may correct several errors; in any event, the complexity of present codes makes them difficult to untangle. They described another evolution of Rabin-like codes in this study in the context of a cyclic remainder ring. The earlier Cauchy MDS cluster codes have more fixed world sizes, but the modern Rabin-like codes emphasize more maintained bounds (prime p is stretched to an odd integer). In light of the Cauchy grid's LU factorization, the newly created Rabin-like codes may be deciphered effectively. It is shown that the suggested approach has a lower translating complexity than current Cauchy MDS cluster codes. Therefore, in light of the new design, circulating storage frameworks find Rabin-like codes appealing [66]. Chouhan and Peddoju [67] proposed encoding strategy which provides different coded information and equality sections to protect information from tragedy. The majority of frameworks, including distributed storage, now use the elimination coding method to provide more information consistency, steadfastness, and accessibility. Without taking into account the needs of customers, such as high unshakable quality and affordable stockpiling prices, the majority of current research focuses on either the cost of recovery or the above caused by repeating capacity. The optimum features for two encoding boundaries information portions and equality sections should be considered by the capacity specialist co-op when selecting an encoding technique. The benefits of these encoding limits are determined in part by the Quality of Service (QoS) requirements of

the customers, such as storage effectiveness, accessibility, and recoverability. These elements are essential to ensuring greater reliability and lower storage costs. Therefore, in this investigation, they looked for the ideal configurations to provide enhanced reliability and reduced storage costs while taking into account the preferences of the customer. In order to determine the appropriate encoding boundary values, they assessed the RS coding plan from the perspectives of storage discussed above, information accessibility, information recoverability, and storage effectiveness.

Liu et al. [68] proposed scheme GPU accelerated technique. Data replication has lately been mostly replaced with erasure coding in large-scale storage systems. The effectiveness of erasure coding becomes an increasingly important constraint in erasure-coded storage systems as disc I/O speeds and bandwidth usage keep rising. This technique provides GPU-Cauchy Reed-Solomon (G-CRS), a GPU-based implementation of erasure coding that makes use of the (CRS) code, to overcome the aforementioned hurdle. In order to enhance the coding efficiency of G-CRS, they developed and implemented a number of optimization strategies to fully use the GPU resources, including a compact structure to store the bit matrix in GPU constant memory, effective data access via shared memory, and decoding parallelism. To further illustrate the GPU coding speed of G-maximal CRS, they created a simple but precise performance model. G-CRS' performance was compared to that of other state-of-the-art code libraries after being thoroughly tested on modern GPU architectures like Maxwell and Pascal. The analysis revealed that G-CRS outperformed the bulk of other coding libraries in throughput by a factor of 10. Additionally, G-CRS performed up to three times better than PErasure in the same architecture. Wu et al. [69] proposed high speed Cauchy decoding (CODEC) scheme. Erasure codes like RS and Cauchy RS are often used in distributed storage systems. Despite the fact that all erasure codes can successfully recover stored data when errors occur, the computation cost of various Cauchy CODEC implementations has a substantial influence on how well they function in practice, in part because building the decoding matrix is so difficult. In order to build an encoding method based on the hardware decoding mechanism, this work introduced a novel high-speed Cauchy algorithm. This technique needs far less complexity than Gauss LU, GS-Cauchy, and GS-Direct-Cauchy to perform as well as CRS for both decoding matrix generation and CODEC.

Mayo et al. [70] proposed the Convolutional Sparse Coding (CSC) computation that enables single AI of components to occur during depiction learning by simultaneously increasing a 2-standard dedication concept and a sparsity-upholding penalty. In this particular study, a regularization term that is derived from an anticipated Cauchy previously is made to disclose the coefficient of the element guides of a CSC generative model. The sparsity punishment term that was previously constructed is first handled by applying its proximal administrator, and then that arrangement is applied component by component to the coefficients of the separated parts in order

to improve the CSC cost work. The Iterative Log-Threshold (ILT) method and several strategies that rely on the reduction of conventional punishment tasks via delicate and harsh thresholding are correlated with the time spent recreating normal images. The Iterative Cauchy Thresholding (ICT) mechanism has been presented. Zhao et al. [71] proposed RS-polar codes combine Reed-Solomon (RS) codes' unmatched burst error revision capability with the low encoding and translating complexity of polar codes to create a hybrid system. Their code rates are nonetheless constrained by the speed of the external RS codes and the inward polar codes. In order to create a rate-matching plan that combines the benefits of linked and penetrating codes to further increase block execution and adaptability, they provided a normal dispersed penetrating approach and consolidate it with RS-polar codes. In order to increase penetrating execution by conserving more high unchanging quality channels after penetrating, a power level division penetrating calculation is implemented. According to recreation findings, the suggested method outperforms current computations for low coding rates and brief transmission durations. This technique calculation may provide about 1.32 dB execution gain at the Block Error Rate (BLER) of 10⁻³ with code length $M = 2400$ and coding rate $r' = 1/3$, compared to the semi uniform puncturing calculation.

Jiron et al. [72] proposed Reed-Solomon codes with a characteristic 3 Galois field can be used for data transmission through a visible light correspondence (VLC) channel. This is because data can be sent using red, green, and blue tones. It used a VLC channel because it can be used for the arithmetic of a Galois Field of dimension 3 and avoid switching from non-paired to parallel jobs. They also discussed the results of our simulations for Reed-Solomon codes in terms of the 3 and 2 Galois fields, estimating a 1.7 dB difference between these codes. Dau et al. [73] proposed repairing RS codes. Despite having excellent error-correcting features, RS codes have been disregarded in communicated capacity applications because to the widely held incorrect belief that they have a slow fix transfer speed: A naive repair plan would involve duplicating the whole record in order to reinstate a single erased code word indication. Among all direct encoding techniques, Guruswami and Wootters' single-eradication RS code fix approach produces the highest fix transmission capacity. Their main idea is to recover the deleted picture by combining a huge number of its trails, each of which may be constructed from the trails of many other images. To handle deletions two and three, the trail gathering approach is expanded.

Tamo et al. [74] proposed optimal repair problem codes. The data that has been eradicated using a code, such as the RS code, is recovered in the maintenance problem in a distributed manner. They examined the smallest amount of possible inter server correspondence when fixing a single hub or a number of hubs in an RS-coded capacity framework. Effective RS code repair was pioneered by Guruswami and Wootters (2016), who demonstrated how it may be accomplished with less data transfer capacity than the basic methods. In this

investigation, they created sets of RS codes that complete the cut set intended for correcting at least one node. In order for scalar MDS algorithms to obtain the cut-set limit using straight fix operations, super-outstanding scaling is both necessary and sufficient, as shown by the almost identical lower bound on 1 that they also provided. Li et al. [75] proposed RS codes with sub-packetization size. In this investigation, the sub-packetization size of RS codes and maintenance data transfer is focused. The maintenance transmission capacity is the volume of data sent from enduring hubs to a confronted hub. Guruswami and Wootters recently presented a maintenance technique for RS codes at the time when the evaluation focuses cover the whole restricted field. Even though the sub-packetization size may be short for arbitrary reasons, the maintenance transfer speed exceeds the MSR constraint. Tamo, Ye, and Barg were the first to achieve the MSR limit, however the size of the sub-packetization increases faster than the evaluation focus volume can dramatically increase. In this work, they provided code that expands these outcomes to support varying assessment point sizes and repair systems that do so. To put it another way, they devised plots that provide points of intersection. The variable trade-off between sub-packetization size and fixed transmission capacity is taken into account by these approaches.

Chen et al. [76] in their research has shown that RS codes provide a maintenance method that considers the optimal repair transmission capacity for destroyed hubs. In this review, this result is spread among two categories. First, this technique provided a different maintenance method for the RS codes and demonstrated that it is resilient to inaccurate data provided by the helper hubs while retaining the optimal fix transmission capacity. Then, they created a unique set of RS codes that provide optimum access to every single compromised hub. They also demonstrated how the built codes, which take ideal access fix and ideal slip-up adjustment into account, can support the two viewpoints. They also demonstrated how every scalar MDS code that satisfies the cut set bound and has a fixed data transfer capacity provides a maintenance method with an ideal access property.

Table 3 shows comparison of different data recovery techniques.

D. DATA CLASSIFICATION TECHNIQUES

Pham and Prakash devised the Bagging-Based Naive Bayes trees (BAGNBT) method [97]. It is employed in Vietnam to categorize landslide vulnerability. This tactic was validated using statistical indices and tests like the Chi-square test. Numerous analytical models for the association were used in this study. The results show that the BAGNBT with the Area Under the receiver operating the characteristic Curve (AUC) (0.834) gave superior outcomes when compared to Random Forest based Naive Bays Trees (RFNBT) (0.830). This shows that the technique is a superior and practical elective way for determining landslide vulnerability. Utilizing tools like the AUC, statistical pointers, and the Chi-Square test, the models in the present test have been verified. Using

TABLE 3. Comparison of various data recovery techniques.

Sr. No.	Author	Technique	Advantages	Disadvantages
1	Zhang et al. [60]	CaCo technique for data storage in the cloud	Capability to select the best coding scheme	Complexity increased with multiple calculations
2	Bian et al. [61]	Optimal Weakly Secure Minimum Storage Regenerating Code	Provides optimal weak security with improved performance	In the worst-case scenario, the OWSPM-encoding MSR's time was almost eight times more than that of the PMMSR codes scheme
3	Tang and Cai [62]	Binary extension field RS codes	The proposed approach eliminates matrix inversion in decoding, which considerably improves decoding efficiency.	Encoding process has not been considered.
4	Tang and Zhang [63]	Data recovery Scheme with Joining an identity matrix and a parity matrix in the shape of a Vandermonde matrix	Significant boost in decoding throughput.	Complex
5	Tai et al. [65]	two-dimensional RS code	Methods improve the newly discovered development of two-layered Reed-Solomon codes, which only offer logarithmic error correction capabilities	-
6	Wu et al. [69]	hardware decoding mechanism	Far less complexity than Gauss LU, GS-Cauchy, and GS-direct-Cauchy to perform as well as CRS for both decoding matrix generation and decoding	No real time implementation
7	Tamo et al. [74]	RS codes, scalar MDS algorithms	Recovers data effectively	Optimization needed to reduce time overhead
8	Li et al. [75]	RS codes with sub-packetization size	Trade-off between sub-packetization size and fixed transmission capacity is taken into account	-
9	Chen et al. [76]	RS codes, MDS code	optimal repair transmission possible	-

the AUC standard, prototypes are verified. The “sensitivity” and “B100-specificity” parameters are used to display the AUC. For value 1, models are thought to be faultless. When a model’s AUC value is greater, it is categorized as being excellent. The model’s acceptability is determined by a variety of performance indicators, including Root Mean Squared Error (RMSE), Kappa, and Accuracy (ACC). The following landslide-causing elements were chosen for the assessment of landslide vulnerability: inclination, separation to defects, bend, street thickness, profile bend, perspective, plan bend, height, separation to streets, separation to rivers, precipitation, flaw thickness, and area utilization. Maps of these components were created for analysis. The BAGNBT process comprises the following four fundamental steps:

- **Dataset generation:** This step primarily involved the creation of two necessary datasets. The training dataset comes first, followed by the validation dataset. They utilized 70% of the datasets with rockfall and non-rockfall values for the training dataset and 30% of the remaining data for the validation datasets. Rockfall and

non-rockfall were represented by the Boolean values 1 and 0, respectively.

- **Training of novel models:** In this method, the hybrid model was trained using the training dataset. The given dataset was optimized for classification in this phase using the BAG ensemble. To match the high accuracy of the BAGNBT model, 11 iterations were found to be necessary. Along with it, improved training data items were utilized to identify the non-landslide and landslide classes for the geographical prediction of the landslide using a classifier similar to the Naive Bays Tree (NBT). The last stage was the recycling of a BAG combination to create classifiers for naive bays trees in order to create a new model.
- **Model validation:** Using several methods, including the AUC and the Chi-Square test, the new model was accepted.
- **Creating maps of landslide vulnerability:** NBT, Support Vector Machine (SVM), RFNBT, and BAGNBT models were used to develop maps of landslide risk.

In order to assess the risk of rockfalls in the most susceptible cities where landslides occur regularly, such as Vietnam, the recommended technique was combined with the BAG collaborative and Naive Based Trees as a classifier. Testing has been successful using statistical indicators, AUC, and chi-square methodology. Estimation and relationship effects of the proposed system demonstrate that the BAGNBT model has a strong showing for the analysis of rockfall vulnerability, with an AUC value of 0.834 in comparison to the RFNBT value of 0.830. It moderately categorizes data. Its flaw is that it takes a little time since it uses numerous rounds to categorize data.

A random forest (RF) classifier-based method for data classification was introduced by Lakshmanprabu et al. [77]. The analysis of a specific big data set gathered from various sources is done in this paper using the RF classifier. They used patient data, specifically, as well as details about a variety of health issues. Database author used improved Dragonfly algorithm for proper classification of data taken from the healthcare. The healthcare data is characterized by a chosen RF classifier with the aid of ideal characteristics. The output numbers from the precision result are limited to 94.2 as of execution. Therefore, unique metrics are used and contrasted with current tactics to evaluate the feasibility of the procedures. The author utilized online information as training data and real health care data as test data. Information gathering, observation, splitting, digitization, control, and support are some of the steps of this task. Following data collection, the highlighted features are extracted with the aid of the dragonfly algorithm and provided to the RF classifier for further data categorization in accordance with the user's requirements.

As it has little impact on big data diseases and variances in very large volumes of data, the RF classifier approach works well for categorizing the data. It is based on a tree generation method where several trees are formed and the best estimate of the outcome is selected among them. RF creates a random sample of the data and determines the essential arrangement of attributes for constructing a decision tree. Figure 6 shows how the RF structure is made. RF builds a case from the data and verifies the conclusion of the best solutions using several decision trees already produced.

The training dataset is made available in place of bootstrap tests for creating each option tree. At each node division in a decision tree that is being improved, a random subset of a few irregular components is browsed to determine the optimal division based on these few characteristics.

Following are the three primary criteria that the RF classifier used to categorize huge data:

- Node sizes are chosen that are different from those used in decision tree comparison.
- Typically, a tree count of up to 500 trees is appropriate.
- The count of sampled predictors to be tested at each split would seem to be an important factor that would influence how effectively RF functions. The following

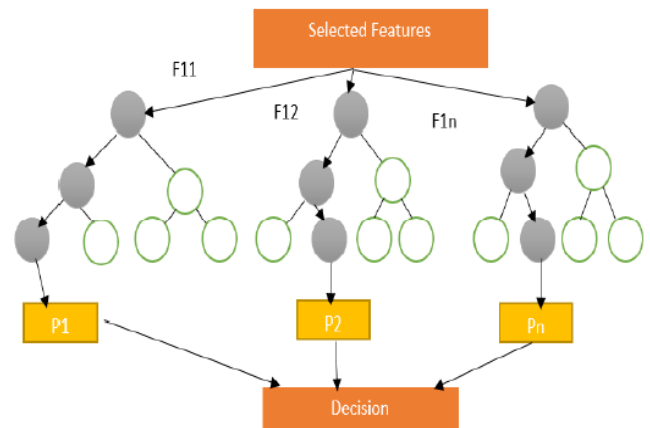


FIGURE 6. Structure of RF.

(1) is applied in mistake circumstances.

$$RandomForestError = RandomForestg_1, g_2(F(v_1, v_2) < 0) \quad (1)$$

where, g_1, g_2 are random values and v_1 and v_2 are vectors.

Classifier accuracy and interdependence are measured by quality and relationship criteria. Different features utilized during the base classifier's phase return the amount of data that is randomly selected from the properties' initial arrangement, notably at each node of the base decision tree. For each tree to provide ideal outcomes and be useful for RF classification, the most remarkable probability method is used. In this study, accuracy rates are between 90% and 95%. The large database is the cause of the suggested calculation's moderate computational deficit.

Abawajy et al. [78] proposed LIME classifiers for classifying big data. In artificial intelligence and data mining, a variety of techniques for building ensemble classifiers are widely known. Each ensemble meta classifier creates a common classification scheme by combining a number of basis classifiers. For instance, in earlier publications explored more versatile multi-classifier systems and effective multi-tier classifiers scheme. Although unusual, four stage LIME classifier structure was influenced by earlier studies in the literature. When provided a hint, an illustration, or a template of a single base classifier as an input parameter, customary collaborative meta classifiers produce their gathering of base classifiers. Following the generation phase, they analyzed data, gathered their outputs, and integrated them to create the final judgment using the whole ensemble of the basic classifiers. As seen in Figure 6, Random Forest, for instance, automatically creates a set of random trees and utilizes them. Other basic classifiers and other methods for producing and merging them are used in many other known ensemble meta classifiers, which operate similarly. The focus of the study is a novel method for producing enormous LIME classifiers designed specifically for managing big data. The creation of a complex multitier system is automated using LIME classifiers. They made it simple to combine several ensemble meta

classifier algorithms at various levels to produce extremely big classifiers. In this study, LIME classifiers with four layers are employed. The third-tier classifier functions as an integral part of the fourth-tier classifier, and each second-tier classifier functions as an integral part of its third-tier classifier parent. They simultaneously incorporated different classifiers into the second, third, and fourth tiers and combined them into a single integrated iterative system. This construction's fourth layer ensemble meta classifier repeatedly calls its third-tier classifier, which calls its second-tier classifier.

A LIME classifier is simple to set up and produce. The fourth-tier classifier uses a single instance of a second-tier ensemble as an input parameter to construct all third-tier classifiers. All third-tier classifier are generated by the fourth tier classifier, who then uses them in the same manner as base classifier. The generation and combining of each third-tier classifier's second tier classifier follow a similar process. The second layer classifier then generates, uses, and combines its base classifier in accordance with established procedures. A designer must choose which classifier will function at the fourth tier in order to initiate a four-layer LIME classifier. The fourth-tier classifier is then given a parameter by the designer specifying which third tier classifier will be utilized as part of the fourth-tier classifier's standard generation process. The designer then specifies the base classifier handled by the second-tier classifier and the second-tier classifier technique to be utilized by the third-tier classifier. The Waikato Environment for Knowledge Analysis's (WEKA) diverse ensemble meta classifier and base classifier were employed in this study. The WEKA Simple CLI command line, allows for the designer to provide all settings for a LIME classifier. The Java programming language's embedded iterative and recursive capabilities are then used by the Simple CLI to automatically construct the whole system. Each classifier selected by the designer uses its own approach to generate the classifier at the lower tier after initialization.

The fourth level classifier first creates a collection of the classifiers at the third layer. Second, each of the third-tier classifiers developed employs its own method for producing second tier classifiers. Finally, each second-tier classifier created at the previous stage employs its own method of producing a collection of base classifiers in accordance with the type of base classifier. The LIME classifier's work on the generation step is now complete. The LIME classifier then analyses the data, where the direction of the arrows represents the flow of the data. The second layer ensemble meta classifier receives the output from the base classifier after it has analyzed the characteristics of the original cases. In order to transmit their own output to their parent third tier classifier, the second-tier classifier gathers all of the outputs of the base classifier, combines them, and sends their own output. Similar to the second layer classifier, the third-tier classifier gathers, analyses, and combines the outputs of the second-tier classifier before sending its own output to the fourth-tier classifier. The ultimate determination of the whole

LIME classifier is produced by the fourth-tier classifier after analysis of the third-tier classifier's output.

1) ENSEMBLE META CLASSIFIERS

They examined the effectiveness of the following ensemble meta classifiers: Bagging, AdaBoost, Dagging, Grading, MultiBoost, Decorate and Stacking using WEKA's Simple CLI command line. AdaBoost employs a number of classifiers sequentially. Each classifier is trained using examples that presented more of a challenge to the one before it. To do this, each occurrence is given a weight, and if it proves challenging to categorize, its weight is increased. We made advantage of the very effective AdaBoost classifier from. By randomly and with replacement resampling the provided training set, bagging (bootstrap aggregating) creates a collection of new sets. Bootstrap samples refer to these collections. For each of these new training sets, a new classifier is subsequently trained. When the base classifier is sluggish, dragging is helpful. The training set is split up into a number of disjoint stratified illustrations. Duplicates of the identical basic classifier are then trained, and their productions are averaged using ballot. Decorate is the process of creating unique fake training samples to create various classifier ensembles. Numerous studies have shown that decorate regularly generates ensembles that are more accurate than the basic classifier, Bagging, Random Forests, and that are equivalent to Boosting on bigger training sets. These ensembles are also more accurate than boosting on short training sets. Grading certifies the basic classifier's output as accurate or incorrect, and the graded results are then merged. With the wagging technique, MultiBoost expands on AdaBoost's methodology. A variation of bagging called wagging chooses the bootstrap samples based on the weights of training examples produced during boosting. According to [79], tests on a broad and varied array of UCI data sets have shown that MultiBoost produces superior correctness noticeably more often than wagging or AdaBoost. Stacking, in which a meta-learner accumulates the outputs of several base classifiers, might be seen as an extension of voting [80].

2) FEATURE EXTRACTION

The purpose of the study is to create a brand-new LIME classifier for big data examination. They didn't utilize any novel or complex feature selection approaches; instead, they employ straightforward static characteristics that are well-known in the field of malware detection to make the dataset building and pre-processing easier. Obfuscation methods may often be used to get around a set of static characteristics. Although they may speed up processing and be utilized when malware has only tried to enter the system and hasn't yet been executed, they are still often employed for malware detection and categorization. For the purposes of the research, it is advantageous to view additional changes to static features produced by obfuscation as a benefit to consider new obfuscated versions of the same malware in

a dataset as separate different instances of malware rather than as variations of the same malware. This enriches the data and makes it more suitable for testing methods designed for big data. They used byte sequences, sometimes known as n-grams, in our investigations. They are n-byte sequences taken from an executable file to be categorized. N-grams generate effective static characteristics for malware detection, which is well-known in the literature. We used the feature extraction technique suggested in [81]. It ranks n-grams using TF-IDF scores to cut down on the number of sequences used as features.

They did not try to extract more complex groupings of features; instead, the current paper analyzed a unique strategy for enhancing classifier performance. For the purposes of this work, they only extracted a basic collection of the features because the focus of this paper is on the contribution of the four-tier LIME classifier. Its most recent tests made use of straightforward features in a malware data set gathered from the honey net and VH Heavens by laboratory's industry partners. To select n-grams in order to cut down on the number of features, they used term frequency inverse document frequency sequence weights, or TF-IDF weights.

The number of malware and cleanware instances in the provided data set where the sequence w appears at least once is known as the document frequency of the word w , indicated by the symbol $DF(w)$. The importance of each term is calculated using the inverse document frequency. The formula defines it, and it is denoted by $IDF(w)$. TF-IDF defines the phrase frequency inverse document frequency of a word w in instance of malware or cleanware m , or w 's weight in m . This research demonstrated that if several ensemble meta classifiers are joined at various tiers, huge four-tier LIME classifier are fairly simple to use and can be employed to enhance classifications. Investigating LIME classifier for other sizable datasets is an intriguing question for future research. For the malware data set, Random Forest outperformed other base classifiers, and decorate enhanced its results more effectively than other ensemble meta classifiers. The four-tier LIME classifier, which used MultiBoost at the fourth tier, decorate at the third, and Bagging at the second, produced the best results with an AUC of 0.998. Several numerical input parameters affect how well the ensemble Meta classifier performs, which is what this paper is concerned with. To compare the results across all of these ensemble Meta classifiers uniformly across all experiments, we applied the same default settings for these parameters when using them.

In light of the two problems the SVM [82] method has with processing enormous amounts of data, the study proposed a weighted Euclidean distance, radial integral kernel function SVM, and dimensionality reduction technique. Multiple classifications cannot be handled by the SVM, and the modelling procedure is time-consuming. The algorithm addressed these problems. The improved method reconstructs the data feature space, makes it clear where different data samples begin and end, shortens the modelling process, and increases classification precision. The method's viability and efficiency were

confirmed via experiments. The experimental results indicate that the upgraded method may provide better results when employing multi-duplicated samples and a huge data capacity for multi classification.

Fu et al. [83] proposed fine grained technique for data classification. As a consequence of the quick adoption of automated procedures, the variety of malware has swiftly increased, presenting a serious threat to Internet security. Recently, certain approaches for quick malware analysis have been put forward, however they often have a significant processing cost and are unable to accurately identify samples for complex and large-scale malware datasets. In order to quickly and effectively perform fine-grained classification in this work, this paper provided a novel visualization technique for characterizing malware globally and locally. By seeing malware as RGB-colored pictures and extracting global features, they used a unique approach. The Grey Level Co-occurrence Matrix (GLCM) and color moments are selected to define global texture characteristics and color features, resulting in low-dimensional feature data that lessens the complexity of the training model. Simhash also extracts a collection of odd byte sequences from the code and data sections of malware and transforms them into feature vectors as local features. Finally, they combine global and local characteristics to achieve malware classification using RF, KNN (K-Nearest Neighbor), and SVM. This technique has the highest accuracy of 97.47 percent and the highest F-measure of 96.85 percent based on the results of 7087 samples from 15 families. When combined with texture characteristics, color features and local features aid to increase the F-measure by 3.4 and 1 percent, respectively. Overall, fine-grained malware classification may be accomplished at a low processing cost by integrating global and local information.

Some of the most pressing data security issues that come up when examining substantial volumes of data kept in the cloud. Big data has a variety of problems, including problems with big data reduction, big data diversity, big data integration etc. Some of these clients could be in direct rivalry with one another in the market, or they might possess information that is private and private about their own clients. The data kept in the data center may be accessed by the staff of cloud services, which implies that it might be compromised or used to steal identities. The research offers a practical method for preserving data that prevents staff members from acquiring information that may be utilized improperly in the manner previously mentioned. They also advocated a method of securing data access that requires the engagement of many staff members at once. This ensures that, in the case of a dispute over whether or not a staff member accessed the data, there will always be a witness to the fact that she did so. Ten different Perceptron-derived algorithms were utilized in the study described in the publication [84] to explore the proactive nature of malware detection. Most current conventional ways to analyze malware are focused on two main pillars: static analysis and dynamic analysis. The disassembled file's opcode was recovered by Kang et al. using N-gram, and it was

then organized into feature vectors. Their conclusions may be found in [85]. Additionally, they assessed the categorization's correctness depending on the opcode's length. The results show that the opcode characteristic is helpful for classifying malware and that a shorter opcode is preferred over a larger one. Iwamoto et al. [86] similarly obtained API sequences, but instead of calculating the similarity directly, they converted the API sequences to Function Call Graphs (FCG), then reduced those graphs. The method takes into account their calling connection in order to better distinguish the APIs from one another.

The methods in [87] provide efficient classification but are vulnerable to code obfuscation due to their structure. As a consequence, many other methods for dynamic analysis were presented, such as analysis of system calls and network activities. Xu et al. representations of the system call that were extracted from the computer included system call histograms, N-grams, and Markov Chains. [88] used a graph to isolate and summarize the application layer protocols, with the implementation details acting as the nodes and the commonalities across the protocols acting as the edges. The degree of similarity between various network activity graphs was then determined using a method that measures closeness using graph distance. In the research by Nari et al. [89], network activity was also represented as a graph, but the graph characteristics were determined by statistical properties of graphs. These statistical characteristics include the graph's size and degree.

Numerous visualization-based approaches have been proposed for the analysis of malware as a result of the development of image processing technology. Analysis of static [90] and dynamic [91] features was first often done using graphical techniques. Despite employing a small data set for training, the results showed that the approach had a high degree of accuracy. Their methods for extracting opcodes included disassembly and dynamic execution, which makes them handy for malware encoding and packers. While Trinius et al. used thread graphs and tree maps to represent individual threads and generic behaviors, they were approaching the problem from a different angle. In their study [92], Saxe et al. chose to display the system call log and used the system calls to construct Markov chains that were then used to compute a similarity matrix. Shaid et al. [93] method of color-coding risky APIs according to their level of malice was used to convert behavioral data into graphics for categorization. This method was used to categorize harmful APIs. The semantic strategies for context association were first published by Antunes et al. [94] in 2017, and they expanded our unsupervised model to learn word categories in a natural manner. When the solution, the "Miller-Charles dataset," and an Internet of Things semantic dataset obtained from a mainstream Internet of Things stage were compared, a correlation of 0.63 was discovered. Furthermore, by revealing latent semantic information concealed within distributional profiles, non-negative lattice factorization may be employed to increase accuracy.

Big data mining was used by Shadroo and Rahmani [95] to investigate the most recent studies on IoT. In order to set the scene for researchers in the years to come, an overview of the approaches that have been utilized in the areas of Internet of Things-big data and Internet of Things-data mining is provided here in the context of three categories. According to research by Amroun et al. [96], using a convolutional neural network is the best way to describe human movement. Walking, sitting, standing, and laying down are the four different types of actions for which we opted to assign directions. The results show that the discrete cosine transform, when combined with CNN as a classifier, can achieve an average accuracy of more than 98 percent in the classification of behaviors including walking, standing, sitting, and laying down.

Table 4 compares different classification techniques with accuracy.

TABLE 4. Comparison of various data classification techniques.

Sr. No.	Author	Technique	Accuracy
1	Pham and Prakash [97]	Bagging-Based Naive Bayes trees (BAGNBT) approach	83.4%
2	Lakshmanaprabu et al. [77]	Random forest (RF) classifier-based method for data classification	90% to 94%
3	Pham and Prakash [97]	Support Vector Machine	77.30%
4	Funde and Swain [98]	Weight Based Similarity	92.10%

IV. FUTURE DIRECTIONS

In the future, various points can be covered in regards of privacy of big data and recovery of data. There is scope to improve the already in place security system by including a massive data processing environment. In next development, a primary focus can be placed on finding ways to simplify the process of encrypting and decrypting large amounts of data. The system for hybrid cloud environments may be planned to trade-off resource virtualization with energy conservation using punctured Cauchy RS codes that can further optimize network resources like power and bandwidth.

It's possible that other aspects of the code, such as the amount of data needed for recovery or reads that aren't as good, are a bigger performance bottleneck than the CPU overhead. In the future, it will be exciting to take on these challenges, and we look forward to doing so. It is necessary to conduct additional research in order to identify the background eviction method that is most effective in terms of providing the greatest reduction in root bucket size for the least amount of additional overhead in terms of the proportion

of dummy requests that are added. Important property of “disconnected ORAM operations” in non-colliding clouds can be further researched and exploited to achieve a better ORAM design. This can be done to achieve the goal of having a better ORAM design. It is possible to implement yet another improved strategy by calculating the valid nodes involved in the decoding dynamically in order to relieve some of the strain placed on the overworked storage nodes. It would be interesting to consider ORAM technique to pave the way for scaling multi-user oblivious storage to production levels in the future direction so that people all over the world can take advantage of the high security assurances it provides. An anticipated and impending implementation of the achievement of this objective will be sped up by the creation of secure enclaves in public clouds.

V. CONCLUSION

This article introduces concept of big data security with different security aspects of big data, significance of big data, security requirement in healthcare systems with increasing data day by day. This paper presents the different techniques for preserving privacy of data in healthcare systems with cryptographic and non-cryptographic approaches. This research article demonstrated path hiding approach using different ORAM techniques which covers and provides meta-data obliviousness. It describes the different data recovery techniques and compares various existing techniques like CaCo, Optimal Weakly Secure Minimum Storage Regenerating Codes Scheme etc. with their advantages and disadvantages. This article describes classification techniques for anomaly detection and classifying data into diverse categories like sensitive and normal so that security techniques can be applied on sensitive data to preserve privacy of the that data.

REFERENCES

- [1] D. E. O’Leary, “Big data and privacy: Emerging issues,” *IEEE Intell. Syst.*, vol. 30, no. 6, pp. 92–96, Nov. 2015.
- [2] B. Blobel, D. M. Lopez, and C. Gonzalez, “Patient privacy and security concerns on big data for personalized medicine,” *Health Technol.*, vol. 6, no. 1, pp. 75–81, Jun. 2016.
- [3] J. Hu and A. V. Vasilakos, “Energy big data analytics and security: Challenges and opportunities,” *IEEE Trans. Smart Grid*, vol. 7, no. 5, pp. 2423–2436, Sep. 2016.
- [4] C. Perera, R. Ranjan, and L. Wang, “End-to-end privacy for open big data markets,” *IEEE Cloud Comput.*, vol. 2, no. 4, pp. 44–53, Jul./Aug. 2015.
- [5] P. Jain, M. Gyanchandani, and N. Khare, “Big data privacy: A technological perspective and review,” *J. Big Data*, vol. 3, no. 1, pp. 1–25, Dec. 2016.
- [6] S. Kaisler, F. Armour, J. A. Espinosa, and W. Money, “Big data: Issues and challenges moving forward,” in *Proc. 46th Hawaii Int. Conf. Syst. Sci.*, Jan. 2013, pp. 995–1004.
- [7] R. X. Lu, H. Zhu, J. K. Liu, J. Shao, and X. Liu, “Toward efficient and privacy-preserving computing in big data era,” *IEEE Netw.*, vol. 28, no. 4, pp. 46–50, Jul./Aug. 2014.
- [8] H. Teng, Y. Liu, A. Liu, N. Xiong, Z. Cai, T. Wang, and X. Liu, “A novel code data dissemination scheme for Internet of Things through mobile vehicle of smart cities,” *Future Gener. Comput. Syst.*, vol. 94, pp. 351–367, May 2019.
- [9] Y.-Y. Teing, A. Dehghantaha, K.-K.-R. Choo, and L. T. Yang, “Forensic investigation of P2P cloud storage services and backbone for IoT networks: BitTorrent sync as a case study,” *Comput. Electr. Eng.*, vol. 58, pp. 350–363, Feb. 2017.
- [10] W. Shen, J. Qin, J. Yu, R. Hao, J. Hu, and J. Ma, “Data integrity auditing without private key storage for secure cloud storage,” *IEEE Trans. Cloud Comput.*, vol. 9, no. 4, pp. 1408–1421, Oct. 2021.
- [11] A. Gharaibeh, M. A. Salahuddin, S. J. Hussini, A. Khreishah, I. Khalil, M. Guizani, and A. Al-Fuqaha, “Smart cities: A survey on data management, security, and enabling technologies,” *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2456–2501, 4th Quart., 2017.
- [12] M. S. Farooq, S. Riaz, A. Abid, K. Abid, and M. A. Naeem, “A survey on the role of IoT in agriculture for the implementation of smart farming,” *IEEE Access*, vol. 7, pp. 156237–156271, 2019.
- [13] *Digital Economy Report*, United Nations, New York, NY, USA, 2019.
- [14] *Gartner: Gartner Forecasts Worldwide Public Cloud Revenue to Grow 17% in 2020*. Accessed: Feb. 2020. [Online]. Available: <https://www.gartner.com/en/newsroom/press-releases/2019-11-13-gartner-forecasts-worldwide-public-cloud-revenue-to-grow-17-percent-in-2020>
- [15] *IBM. Block*. Accessed: Feb. 2020. [Online]. Available: <https://www.ibm.com/cloud/learn/block-storage>
- [16] Spectralogic. *Comparing File (NAS) and Block (SAN) Storage*. Accessed: Mar. 1, 2020. [Online]. Available: <https://edge.spectralogic.com/?fuseaction=home.displayFile&DocID=4630>
- [17] N. Dong, H. Jonker, and J. Pang, “Challenges in eHealth: From enabling to enforcing privacy,” in *Proc. Int. Symp. Found. Health Informat. Eng. Syst.*, 2011, pp. 195–206.
- [18] X. Yi, Y. Miao, E. Bertino, and J. Willemson, “Multiparty privacy protection for electronic health records,” in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2013, pp. 2730–2735.
- [19] C. S. Kruse, M. Mileski, A. G. Vijaykumar, S. V. Viswanathan, U. Suskandla, and Y. Chidambaram, “Impact of electronic health records on long-term care facilities: Systematic review,” *JMIR Med. Informat.*, vol. 5, no. 3, p. e35, Sep. 2017.
- [20] L. Griebel, H.-U. Prokosch, F. Köpcke, D. Toddenroth, J. Christoph, I. Leb, I. Engel, and M. Sedlmayr, “A scoping review of cloud computing in healthcare,” *BMC Med. Informat. Decis. Making*, vol. 15, no. 1, pp. 1–16, Dec. 2015.
- [21] P. Li, S. Guo, T. Miyazaki, M. Xie, J. Hu, and W. Zhuang, “Privacy-preserving access to big data in the cloud,” *IEEE Cloud Comput.*, vol. 3, no. 5, pp. 34–42, Sep./Oct. 2016.
- [22] A. Abbas and S. U. Khan, “A review on the state-of-the-art privacy-preserving approaches in the e-Health clouds,” *IEEE J. Biomed. Health Informat.*, vol. 18, no. 4, pp. 1431–1441, Jul. 2014.
- [23] R. Zhang and L. Liu, “Security models and requirements for healthcare application clouds,” in *Proc. IEEE 3rd Int. Conf. Cloud Comput.*, Jul. 2010, pp. 268–275.
- [24] M. Ahmed and A. S. B. Ullah, “False data injection attacks in healthcare,” in *Proc. Australas. Conf. Data Mining, in Communications in Computer and Information Science*, vol. 845, 2018, pp. 192–202.
- [25] E. AbuKhoua, N. Mohamed, and J. Al-Jaroodi, “e-Health cloud: Opportunities and challenges,” *Future Internet*, vol. 4, no. 3, pp. 621–645, 2012.
- [26] D. McGraw, “Building public trust in uses of health insurance portability and accountability act de-identified data,” *J. Amer. Med. Inform. Assoc.*, vol. 20, no. 1, pp. 29–34, Jan. 2013.
- [27] Z.-R. Li, E.-C. Chang, K.-H. Huang, and F. Lai, “A secure electronic medical record sharing mechanism in the cloud computing platform,” in *Proc. IEEE 15th Int. Symp. Consum. Electron. (ISCE)*, Jun. 2011, pp. 98–103.
- [28] A. Ibrahim, B. Mahmood, and M. Singhal, “A secure framework for sharing electronic health records over clouds,” in *Proc. IEEE Int. Conf. Serious Games Appl. Health (SeGAH)*, May 2016, pp. 1–8.
- [29] A. Kaletsch and A. Sunyae, “Privacy engineering: Personal health records in cloud computing environments,” in *Proc. Int. Conf. Inf. Syst. (ICIS)*, Shanghai, China, 2011, pp. 1–11.
- [30] D. Mashima and M. Ahamad, “Enhancing accountability of electronic health record usage via patient-centric monitoring,” in *Proc. 2nd ACM SIGHT Symp. Int. Health Informat.*, 2012, pp. 409–418.
- [31] X. Sun, M. Li, H. Wang, and A. Plank, “An efficient hash-based algorithm for minimal K -anonymity,” in *Proc. 31st Australas. Conf. Comput. Sci.*, vol. 74, 2008, pp. 101–107.
- [32] S. Narayan, M. Gagné, and R. Safavi-Naini, “Privacy preserving EHR system using attribute-based infrastructure,” in *Proc. ACM Workshop Cloud Comput. Secur. Workshop*, 2010, pp. 47–52.
- [33] A. Sahai and B. Waters, “Fuzzy identity-based encryption,” in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2005, pp. 457–473.

- [34] T. Hupperich, H. Löhr, A.-R. Sadeghi, and M. Winandy, "Flexible patient-controlled security for electronic health records," in *Proc. 2nd ACM SIGHIT Symp. Int. Health Informat.*, 2012, pp. 727–732.
- [35] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.
- [36] N. Pramanick and S. T. Ali, "A comparative survey of searchable encryption schemes," in *Proc. 8th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2017, pp. 1–5.
- [37] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. IEEE Symp. Secur. Privacy (SP)*, 2000, pp. 44–55.
- [38] M. Barni, P. Failla, R. Lazzeretti, A. Sadeghi, and T. Schneider, "Privacy-preserving ECG classification with branching programs and neural networks," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 2, pp. 452–468, Jun. 2011.
- [39] C. Gentry and S. Halevi, "Implementing gentry's fully-homomorphic encryption scheme," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2011, pp. 129–148.
- [40] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?" in *Proc. 3rd ACM Workshop Cloud Comput. Secur. Workshop*, 2011, pp. 113–124.
- [41] S. Carpov, T. H. Nguyen, R. Sirdey, G. Constantino, and F. Martinelli, "Practical privacy-preserving medical diagnosis using homomorphic encryption," in *Proc. IEEE 9th Int. Conf. Cloud Comput. (CLOUD)*, Jun. 2016, pp. 593–599.
- [42] V. C. Hu, D. Ferraiolo, and D. R. Kuhn, *Assessment of Access Control Systems*. Gaithersburg, MD, USA: National Institute of Standards and Technology, 2006.
- [43] M. F. F. Khan and K. Sakamura, "Fine-grained access control to medical records in digital healthcare enterprises," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, May 2015, pp. 1–6.
- [44] H. S. G. Pussewala and V. Oleshchuk, "A patient-centric attribute based access control scheme for secure sharing of personal health records using cloud computing," in *Proc. IEEE 2nd Int. Conf. Collaboration Internet Comput. (CIC)*, Nov. 2016, pp. 46–53.
- [45] S. Alshehri and R. K. Raj, "Secure access control for health information sharing systems," in *Proc. IEEE Int. Conf. Healthcare Informat.*, Sep. 2013, pp. 277–286.
- [46] B. Zhao, Z. Chen, and H. Lin, "Cycle ORAM: A practical protection for access pattern in untrusted storage," *IEEE Access*, vol. 7, pp. 26684–26695, 2019, doi: [10.1109/ACCESS.2019.2900304](https://doi.org/10.1109/ACCESS.2019.2900304).
- [47] A. Pujol, L. Murphy, and C. Thorpe, "FedORAM: A federated oblivious RAM scheme," *IEEE Access*, vol. 8, pp. 187687–187699, 2020.
- [48] X. Wanshan, Z. Jianbiao, and Y. Yuan, "DESSE: A dynamic efficient forward searchable encryption scheme," *IEEE Access*, vol. 8, pp. 144480–144488, 2020.
- [49] T. Hoang, R. Behnia, Y. Jang, and A. A. Yavuz, "MOSE: Practical multi-user oblivious storage via secure enclaves," in *Proc. 10th ACM Conf. Data Appl. Secur. Privacy*, New York, NY, USA, Mar. 2020, pp. 17–28.
- [50] Z. Chen, B. Zhao, H. Lin, and L. Chen, "Etoram: A more efficient ORAM for secure computation," *IEEE Open J. Comput. Soc.*, vol. 1, pp. 285–294, 2020.
- [51] K. S. Al-Saleh and A. Belghith, "Radix path: A reduced bucket size ORAM for secure cloud storage," *IEEE Access*, vol. 7, pp. 84907–84917, 2019.
- [52] T. Hoang, C. D. Ozkaptan, G. Hackebeitl, and A. A. Yavuz, "Efficient oblivious data structures for database services on the cloud," *IEEE Trans. Cloud Comput.*, vol. 9, no. 2, pp. 598–609, Apr. 2021.
- [53] R. Xu, J. Joshi, and P. Krishnamurthy, "An integrated privacy preserving attribute-based access control framework supporting secure deduplication," *IEEE Trans. Depend. Sec. Comput.*, vol. 18, no. 2, pp. 706–721, Mar. 2021.
- [54] W. Tian, R. Li, Z. Xu, and W. Xiao, "Loco-store: Locality-based oblivious data storage," *IEEE Trans. Depend. Sec. Comput.*, vol. 19, no. 2, pp. 1395–1406, Mar./Apr. 2022.
- [55] Z. Liu, B. Li, Y. Huang, J. Li, Y. Xiang, and W. Pedrycz, "NewMCOS: Towards a practical multi-cloud oblivious storage scheme," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 4, pp. 714–727, Apr. 2020.
- [56] Y. Huang, B. Li, Z. Liu, J. Li, S.-M. Yiu, T. Baker, and B. B. Gupta, "ThinORAM: Towards practical oblivious data access in fog computing environment," *IEEE Trans. Services Comput.*, vol. 13, no. 4, pp. 602–612, Jul. 2020.
- [57] A. Tarannum, Z. U. Rahman, L. K. Rao, T. Srinivasulu, and A. Lay-Ekuakille, "An efficient multi-modal biometric sensing and authentication framework for distributed applications," *IEEE Sensors J.*, vol. 20, no. 24, pp. 15014–15025, Dec. 2020.
- [58] R. Nellutla and M. Mohammed, "Survey: A comparative study of different security issues in big data," in *Emerging Research in Data Engineering Systems and Computer Communications*, vol. 1054. Singapore: Springer, 2020.
- [59] E. Stefanov, M. V. Dijk, E. Shi, T.-H. H. Chan, C. W. Fletcher, L. Ren, X. Yu, and S. Devadas, "Path ORAM: An extremely simple oblivious RAM protocol," *J. ACM*, vol. 65, no. 4, pp. 1–26, 2018.
- [60] G. Zhang, G. Wu, S. Wang, J. Shu, W. Zheng, and K. Li, "CaCo: An efficient Cauchy coding approach for cloud storage systems," *IEEE Trans. Comput.*, vol. 65, no. 2, pp. 435–447, Feb. 2016.
- [61] J. Bian, S. Luo, Z. Li, and Y. Yang, "Optimal weakly secure minimum storage regenerating codes scheme," *IEEE Access*, vol. 7, pp. 151120–151130, 2019.
- [62] D. Tang and H. Cai, "A novel decoding method for the erasure codes," *Secur. Commun. Netw.*, vol. 2021, pp. 1–12, Jul. 2021.
- [63] Y. J. Tang and X. Zhang, "Fast en/decoding of Reed–Solomon codes for failure recovery," *IEEE Trans. Comput.*, vol. 71, no. 3, pp. 724–735, Mar. 2022.
- [64] L. Yu, Z. Lin, S.-J. Lin, Y. S. Han, and N. Yu, "Fast encoding algorithms for Reed–Solomon codes with between four and seven parity symbols," *IEEE Trans. Comput.*, vol. 69, no. 5, pp. 699–705, May 2020.
- [65] T. D. Duc, S. Liu, I. Tjuawinata, and C. Xing, "Explicit constructions of two-dimensional Reed–Solomon codes in high insertion and deletion noise regime," *IEEE Trans. Inf. Theory*, vol. 67, no. 5, pp. 2808–2820, May 2021.
- [66] M. Oh, K. Kim, D. Choi, H.-J. Lee, and E.-Y. Chung, "Per-operation reusability based allocation and migration policy for hybrid cache," *IEEE Trans. Comput.*, vol. 69, no. 2, pp. 158–171, Feb. 2020.
- [67] V. Chouhan and S. K. Peddoju, "Investigation of optimal data encoding parameters based on user preference for cloud storage," *IEEE Access*, vol. 8, pp. 75105–75118, 2020.
- [68] C. Liu, Q. Wang, X. Chu, and Y.-W. Leung, "G-CRS: GPU accelerated Cauchy Reed–Solomon coding," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 7, pp. 1484–1498, Jul. 2018.
- [69] R. Wu, L. Wang, and Y. Wu, "A high-speed Cauchy CODEC algorithm for distributed storage system," in *Proc. Int. Conf. Internet Comput. Sci. Eng.*, Jan. 2020, pp. 20–24.
- [70] P. Mayo, O. Karakus, R. Holmes, and A. Achim, "Representation learning via Cauchy convolutional sparse coding," *IEEE Access*, vol. 9, pp. 100447–100459, 2021.
- [71] J. Zhao, W. Zhang, Y. Liu, J. Gao, and R. Zhang, "A rate-matching concatenation scheme of polar codes with outer Reed–Solomon codes," *IEEE Wireless Commun. Lett.*, vol. 10, no. 3, pp. 459–463, Mar. 2021.
- [72] I. Jirón, I. Soto, S. Gutiérrez, and R. Carrasco, "Reed–Solomon codes over Galois fields of characteristic 3 for a VLC channel," in *Proc. South Amer. Colloq. Visible Light Commun. (SACVC)*, Jun. 2020, pp. 1–5.
- [73] H. Dau, I. Duursma, H. M. Kiah, and O. Milenkovic, "Repairing Reed–Solomon codes with multiple erasures," *IEEE Trans. Inf. Theory*, vol. 64, no. 10, pp. 6567–6582, Oct. 2018.
- [74] I. Tamo, M. Ye, and A. Barg, "The repair problem for Reed–Solomon codes: Optimal repair of single and multiple erasures with almost optimal node size," *IEEE Trans. Inf. Theory*, vol. 65, no. 5, pp. 2673–2695, May 2019.
- [75] W. Li, Z. Wang, and H. Jafarkhani, "On the sub-packetization size and the repair bandwidth of Reed–Solomon codes," *IEEE Trans. Inf. Theory*, vol. 65, no. 9, pp. 5484–5502, Sep. 2019.
- [76] Z. Chen, M. Ye, and A. Barg, "Enabling optimal access and error correction for the repair of Reed–Solomon codes," *IEEE Trans. Inf. Theory*, vol. 66, no. 12, pp. 7439–7456, Dec. 2020.
- [77] S. Lakshmanaprabu, K. Shankar, M. Ilayaraja, A. W. Nasir, V. Vijayakumar, and N. Chilamkurti, "Random forest for big data classification in the Internet of Things using optimal features," *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 10, pp. 2609–2618, 2019.
- [78] J. H. Abawajy, A. Kelarev, and M. Chowdhury, "Large iterative multitier ensemble classifiers for security of big data," *IEEE Trans. Emerg. Topics Comput.*, vol. 2, no. 3, pp. 352–363, Sep. 2014.
- [79] G. I. Webb, "MultiBoosting: A technique for combining boosting and waggling," *Mach. Learn.*, vol. 40, no. 2, pp. 159–196, 2000.
- [80] D. H. Wolpert, "Stacked generalization," *Neural Netw.*, vol. 5, no. 2, pp. 241–259, Jan. 1992.

- [81] A. Shabtai, R. Moskovitch, Y. Elovici, and C. Glezer, "Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey," *Inf. Secur. Tech. Rep.*, vol. 14, no. 1, pp. 16–29, 2009.
- [82] H. Dai, "Research on SVM improved algorithm for large data classification," in *Proc. IEEE 3rd Int. Conf. Big Data Anal. (ICBDA)*, Mar. 2018, pp. 181–185.
- [83] J. Fu, J. Xue, Y. Wang, Z. Liu, and C. Shan, "Malware visualization for fine-grained classification," *IEEE Access*, vol. 6, pp. 14510–14523, 2018.
- [84] S. Rajan, A. Bardhan, Y. Chen, A. Fuchs, A. Kapre, A. Lane, R. Lu, P. Manadhata, J. Molina, A. C. Mora, P. Murthy, A. Roy, S. Sathyadevan, and N. Shah, "Expanded top ten big data security and privacy challenges," Cloud Secur. Alliance, Los Angeles, CA, USA, Oct. 2013. [Online]. Available: <http://cloudsecurityalliance.org/research/big-data/>
- [85] M. Cimpoesu, D. Gavriluț, and A. Popescu, "The proactivity of perceptron derived algorithms in malware detection," *J. Comput. Virol.*, vol. 8, no. 4, pp. 133–140, Nov. 2012.
- [86] B. Kang, S. Y. Yerima, K. McLaughlin, and S. Sezer, "N-opcode analysis for Android malware classification and categorization," in *Proc. Int. Conf. Cyber Secur. Protection Digit. Services (Cyber Security)*, Jun. 2016, pp. 1–7.
- [87] K. Iwamoto and K. Wasaki, "Malware classification based on extracted API sequences using static analysis," in *Proc. Asian Internet Eng. Conf. (AINTEC)*, 2012, pp. 31–38.
- [88] L. Xu, D. Zhang, M. A. Alvarez, J. A. Morales, X. Ma, and J. Cavazos, "Dynamic Android malware classification using graph-based representations," in *Proc. IEEE 3rd Int. Conf. Cyber Secur. Cloud Comput. (CSCloud)*, Jun. 2016, pp. 220–231.
- [89] S. Nari and A. A. Ghorbani, "Automated malware classification based on network behavior," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Jan. 2013, pp. 642–647.
- [90] J. Donahue, A. Paturi, and S. Mukkamala, "Visualization techniques for efficient malware detection," in *Proc. IEEE Int. Conf. Intell. Secur. Inform.*, Jun. 2013, pp. 289–291.
- [91] J. Saxe, D. Mentis, and C. Greamo, "Visualization of shared system call sequence relationships in large malware corpora," in *Proc. 9th Int. Symp. Vis. Cyber Secur. (VizSec)*, 2012, pp. 33–40.
- [92] P. Trinius, T. Holz, J. Göbel, and F. C. Freiling, "Visual analysis of malware behavior using treemaps and thread graphs," in *Proc. 6th Int. Workshop Vis. Cyber Secur. (VizSec)*, Oct. 2009, pp. 33–38.
- [93] S. Z. M. Shaid and M. A. Maarof, "Malware behavior image for malware variant identification," in *Proc. Int. Symp. Biometrics Secur. Tech. (ISBAST)*, Aug. 2014, pp. 238–243.
- [94] M. Antunes, D. Gomes, and R. L. Aguiar, "Towards IoT data classification through semantic features," *Future Gener. Comput. Syst.*, vol. 86, pp. 792–798, Sep. 2018.
- [95] S. Shadroo and A. M. Rahmani, "Systematic survey of big data and data mining in Internet of Things," *Comput. Netw.*, vol. 139, pp. 19–47, Jul. 2018.
- [96] H. Amroun, M. H. Temkit, and M. Ammi, "Best feature for CNN classification of human activity using IoT network," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber; Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Jun. 2017, pp. 943–950.
- [97] B. T. Pham and I. Prakash, "A novel hybrid model of bagging-based Naïve Bayes trees for landslide susceptibility assessment," *Bull. Eng. Geol. Environ.*, vol. 78, no. 3, pp. 1911–1925, 2017.
- [98] S. Funde and G. Swain, "Security aware information classification in health care big data," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 5, pp. 4439–4448, 2021.



SNEHALATA FUNDE received the Bachelor of Engineering degree in computer engineering from Pune University, Pune, Maharashtra, India, in 2012, and the Master of Engineering degree in computer engineering from Savitribai Phule Pune University, Pune, India, in 2014. She is currently a Research Scholar of computer science and engineering with Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India. She has authored more than ten articles in various domains like data mining, data security, and networking.



GANDHARBA SWAIN received the M.C.A. degree from the University College of Engineering, Burla, in 1999, the M.Tech. degree in CSE from the National Institute of Technology, Rourkela, India, in 2004, and the Ph.D. degree in CSE from Siksha 'O' Anusandhan University, Bhubaneswar, India, in 2014. He is currently working as a Professor with the Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India. He has more than 20 years of teaching experience, and authored two books and more than 90 research articles. Many of his articles are published in journals of reputed publishers like Elsevier, Springer, Hindawi, Wiley, and Inderscience. He has done extensive research work on digital image steganography, particularly addressed the various problems like fall off boundary problem, range mismatch problem, fall in error problem, detection by RS analysis, detection by PDH analysis, and tradeoff between PSNR and capacity. His research interests include security, image tamper detection, and block chain technology.

• • •