ORIGINAL ARTICLE

A machine learning framework for providing data integrity and confidentiality for sensitive data cloud applications

Eswara Narayanan[1] · **B. Muthukumar**[2]

Received: 5 January 2022 / Revised: 5 April 2022 / Accepted: 1 July 2022

Abstract Most cloud computing applications require an innovative, efficient model for balancing their data security. People from anywhere at any time can access cloud applications with sensitive data. Sensitive data needs a legal method for protecting with more security aspects like constrained-based access or cryptography. Cloud computing cannot assure security against various real-time threats continuously affecting and trying to access the data. Data security is a severe problem in cloud computing because data storage is placed in multiple locations all over the globe. All cloud users worry about two major security aspects: data integrity and Data Confidentiality in cloud computing. Several earlier research works have proposed many methods for cloud computing security and were tested with academics and industry-based applications regarding data security and privacy preservation. Recent Government, industrial, business, and other applications are also concern about data security and privacy following the hardware and software in cloud architecture. This problem is taken as the serious problem of cloud computing, and this paper is motivated to design and implement a Machine Learning model for security provision. Thus, it is aimed to develop and implement a Machine Learning Used Confidential (MLC) protocol for analyzing the data, user, and storage information when the data-owner is accessing and sharing their data to detect and eliminate malicious threats to provide security. The proposed MLC uses one of the popular machine learning algorithms, such as, Convolution Neural Network algorithm, which ensures confidentiality by addressing the susceptibilities during the training process. The experimental results show that the proposed MLC protocol outperformed the others and proved its efficiency regarding security provision.

Keywords Data integrity · Data confidentiality · Cloud security · Data-level security · Privacy preservation

1 Introduction

Cloud computing provides a poor level of security regarding the user, data, and data storage due to its pay-n-use architecture. Any user can access the cloud anytime from anywhere. It makes non-restricted cloud usage and increases fraud. Most financial industries suffer from fraudulent people and their activities since online malicious people steal many financial losses. In the economic sectors, various sectors like medical, healthcare, corporate information are losing their data because of the insurance people. So, cloud computing applications require a highly secured model for their business transactions regarding message, data, and finance. By considering all the cloud application aspects (see Fig. 1). A complete security framework used for any cloud application is illustrated in Fig. 1. The framework comprises three levels of security, such as user-level, data-level, and data storage-level. This paper initiates to provide data storage-level security using data integrity, data confidentiality, and data availability. Data Integrity (DI) and Data Confidentiality (DC) helps to provide software-based data security, whereas Data Availability (DA) and Data Privacy (DP) provide hardware-based protection.

✉ Eswara Narayanan
eswaranarayanan0112@outlook.com

B. Muthukumar
anbmuthusba@gmail.com

[1] Sathyabama Institute of Science and Technology, Chennai 6001191, India

[2] Department of Information Technology, DMI College of Engineering, Chennai, India

Published online: 26 July 2022

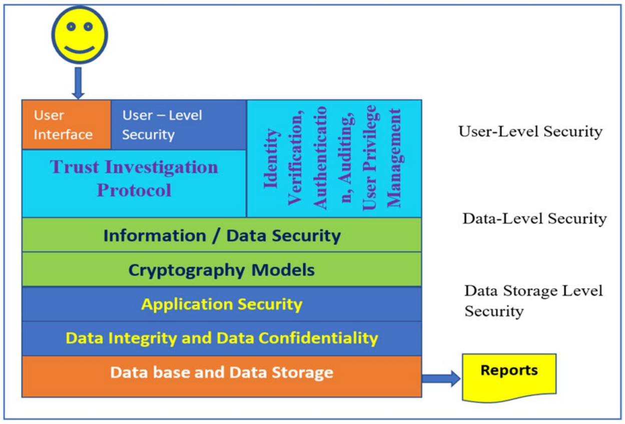

User Interface
User – Level Security
Identity Verification, Authenticatio n, Auditing, User Privilege Management
Trust Investigation Protocol
User-Level Security
Information / Data Security
Cryptography Models
Data-Level Security
Application Security
Data Integrity and Data Confidentiality
Data Storage Level Security
Data base and Data Storage
Reports

Fig. 1 Complete security framework for cloud computing

1.1 Contribution of the paper

This paper profoundly explains Data Integrity and Confidentiality theoretically. A machine learning model is implemented to provide the same for a real-time cloud application. It demonstrates the application of a security model that integrates the DI and DC. The proposed model is trained with a benchmark dataset and evaluates the performance. The DI provides correctness, completeness, freshness, and DC provides the data-level encryption. From the comparison, it is concluded that the proposed model outperforms the existing one. To understand the research problem, a detailed study of the DI, DC, and DA is carried out. Based on the limitations, the proposed model is constructed and trained for testing the actual industrial data.

1.2 Data integrity

One of the authors (Alsheikh-Ali et al. 2011), explained that a system's data integrity is one of the most important aspects. Overall, data integrity refers to preventing unlawful deletion, alteration, or creation of information. As well as the management company's rights to such resources Protecting important information and services is a priority for enterprise resources. Stolen, exploited, or abused in any manner in a stand-alone system, data integrity is very straightforward to implement. A centralized database management system usually handles database constraints and transactions, ensuring data integrity in a stand-alone system (DBMS).

Transactions must follow the ACID principles (atomicity, consistency, and isolation). Isolation and endurance are two qualities that guarantee data security. Most databases can manage many transactions at the same time and support ACID transactions. The integrity of data is critical. Authorization is a method of restricting data access. The process by which a system determines the degree of security required for authorized users to get access to a secure environment under the system's control. The preservation of data in the cloud is referred to as "cloud data integrity." Information integrity It is essential that data is not lost or changed. Users are not permitted to keep the integrity of their data (Bajaj and Sion 2013). The integrity of data is the bedrock upon which services are built. Cloud computing services include SaaS, PaaS, and IaaS. The cloud is also used to store massive quantities of data. Environmental data processing services are often offered. For maintaining DI, some data techniques such as RAID and like-arrays were employed, and both methods and a digital signature were used (Balmford et al. 2005).

Only those who have been given access to the data may interact with it. Refraining from Organizations that restrict unauthorized access may have a greater chance of success. Assurance of data integrity Monitoring methods provides you an additional knowledge about who or what may be attacking you, as well as changed data or system information that might jeopardize your honesty and sincerity. Cloud computing services need data integrity and accuracy. In addition to consumers and cloud computing providers, a third-party monitoring (TPM) system is needed (Federer et al. 2018). The practice of remotely validating cloud data integrity is now generally recognized as critical for application deployment. Bowers and his colleagues proposed a Theoretically, the usage of remote data integrity checks and mistake repair is predicated on proof of retrievability. Spots are used to implement controls and scripts. The HAIL system uses POR.

For example, a method is used to assess the storage of data in several clouds. Redundancy is achieved by having several copies of the same file. Verify the system's availability and integrity by running a few tests. In their research, Schiffman et al. suggested a Remote Trusted Platform Module (TPM). Make sure you can monitor and control your data integrity through the internet. The phrase "data integrity" relates to the accuracy and completeness of the data in your database. We don't want repeated, erroneous, or broken links between tables while saving data in a database. Therefore, we don't want them. Let's look at an example to understand how broken connections may lead to inconsistencies in data (Gherghina and Katsanidou 2013).

1.3 Data confidentiality

Consumers must ensure that their personal or sensitive data is kept private while keeping it on the cloud. Authentication and access control methods guarantee data confidentiality. Cloud computing may address data confidentiality, authentication, and access control problems by improving the cloud's confidence and trustworthiness (Kumar and Saxena 2011).

Customers don't trust cloud providers, and cloud storage companies struggle to eliminate insider threats. Thus, putting sensitive data on their PCs in cloud storage is very hazardous. In contrast to complicated needs such as inquiries, simultaneous modifications, and thorough approval, simple encryption is hindered by a key management issue and cannot meet those expectations. Algebraic text cipher operation's outputs are consistent with the explicitly specified after-encryption outcomes, guaranteeing. This approach is used to address data confidentiality and cloud data activities. First, Gentry provides a completely homomorphic encryption method allowing it to do every action that can be performed in plain text without decoding it. It represents a significant advancement in the field of homomorphic encryption technology (Lee et al. 2004).

On the other hand, the encryption technique requires a complex mathematical computation, which increases processing and storage expenses. Fully homomorphic encryption is still a long way off in real-world applications. Digital signature authentication may be used to create a secure communication link, and 3DES is beneficial for encrypting large amounts of data in blocks. Furthermore, in cloud computing, various encryption methods are being developed to guarantee the security of user data. Database search and encryption as a consequence, scientists are concentrating their efforts on cloud-based homomorphic encryption algorithms, which have shown to be more effective than homomorphic encryption techniques (Puttaswamy et al. 2011).

Cryptography is often used to look for information. Cloud-based systems, where client privacy is a key concern, may benefit from using these techniques. For encrypted cloud data, this article explains how to use a privacy-preserving multi-keyword search method. Searching for keywords in encrypted cloud data and categorizing search results while preserving user privacy is possible with this approach. The proposed approach may offer the greatest degree of safety by separating the user's input into pieces (Ranganathan et al. 2002).

Cloud design requires that these data bits be encrypted and stored in a variety of databases. It is because each segment of data in the cloud is encrypted and kept in a separate database. It is possible to adjust resources over time, based on changes in network architecture and particular transit routes, by using bespoke measurement technologies. Depending on the computing and storage capabilities available, customized measurements may be performed. Because of the dynamic nature of networks, resource allocation may not always result in the most efficient allocation of resources based on a customized and active approach. Because resources may vary, the system must constantly optimize changes in user needs, whether offline or online and changes in resource connectivity. In terms of privacy, data protection is accomplished by ensuring that the data is only available to authorize or by presenting the data in such a manner that it is only accessible to those who need it (e.g., a key for decrypting the enciphered data) (Sandhu 1993).

1.4 Data availability

When a catastrophe occurs, such as a hard disc failure, an IDC fire, or a network failure, data availability refers to how much data the user can access or retrieve and how it is recovered. Users utilize ways to verify their data rather than depending on others. Relying only on the Cloud Service Provider's credit guarantee Data storage through cross-border servers is no longer an issue. Consumers have valid worries regarding the practices of cloud providers (Sivathanu et al. 2005).

The laws of the nation in which they live regulate the cloud. Customers must be informed of the laws that govern them. In addition, the cloud. The service provider, in particular, must take precautions to ensure data security. Security and integrity of data are critical concerns. If you want to establish trust with your customer, the cloud computing service provider should share all of your worries with them. This link has a relationship. Providers of cloud services should offer data security assurances and clarify who is in charge of data security. As a customer in a foreign country, it is important to be informed of the regulations that govern you. Numerous data problems and challenges are connected with location-based and mobile data storage and data cost, availability, and security that are discussed in the article. Consumers may increase their system confidence by knowing where information is stored. Cloud storage is used to offer a transparent storage service (Wu and Kuo 2001).

Users will benefit since the complexity of the cloud will be reduced, but there are also drawbacks. Storage businesses and storage service providers use the phrase "data availability" to describe goods and services that guarantee data is accessible in a variety of circumstances, from "normal" to "disastrous." Data redundancy is used to ensure data availability (Pedregosa et al. 2011). A data center and a storage-centric mentality are more important to certain providers than servers. Often, big corporate computer systems machines access data through a high-speed optical fiber link to storage devices. For example, ESCON and Fibre Channel are two of the most often used access methods. A redundant array of separate discs is often used to store data (RAID). Storage systems may be installed and reconfigured using unprogrammable or manually controlled switches, such as a director, which can automatically switch to a backup or failover scenario as required (Abadi et al. 2015). In the past 5 years, the authors in Laptev et al. (2015) proposed an Openpath model with software-defined strategies for providing automatic security in IoT-based cloud computing applications. The SDN-based MUD comparison increases the user-level security in terms of authentication and authorization. Lightweight protocol security is applied for 5G-enabled mobile communication. But the data integrity and data confidentiality are less, need to improve.

1.5 Limitations and motivation

Various earlier approaches have been proposed to enhance cloud security in terms of user-level security. Some of them have focused on data security using encryption-decryption methodologies. Whereas, only very few researches have discussed three ways of security but not implemented. The theoretical information understands that any cloud applications need complete security in terms of user level, data level, and data storage level. Thus, this paper motivated to provide Data Integrity and Data Confidentiality (DIDC) as part of the research. The security is ensured by implementing a CNN algorithm.

1.6 Proposed machine learning model

Generally, the DI model verifies the input, validates, eliminates the duplicate, persists, and backup the data. Also, it checks for access control permission and audits the trails. Each time of data breach, the DI tracks the source of the critical. Figure 2 shows the list of threats highlighted as an aspect of data security that preserves the DI and reduces the organizations' risk where the data is used. While data is supplied by an unknown data owner, application, any normal user, or abnormal user, it is essential to validate the input. It ensures the input data is more accurate. Next, it is confirmed that the processed data is not corrupted. It is obtained by verifying the attributes and specifications of the data and organization. While data sharing among various departments of an organization, it is careful to erase duplicate or stray data.

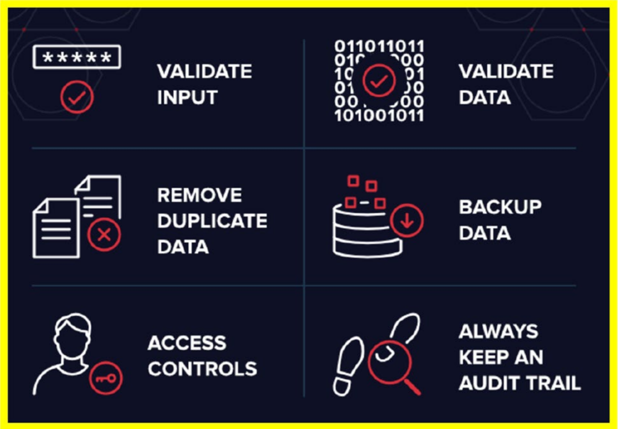


Fig. 2 DI processes

While thinking about DI, maintaining the DI is a complex problem. But the dynamic learning nature of Machine learning is applied for model decay. The ML models are increasing rapidly and drive-by complex feature pipelines and automatic in-built workflows which involve data modifications. The data goes under multiple transformations associated with the application usage to shape it into the correct form fit for the ML model. The above-said transformation is required to apply consistently for all the application data along with the ML model. Also, the ML model utilizes the data in different pipelines used by various organization teams and exposes completely distinct interfaces. The DI concerning a database is maintained with certain constraints, defined by a set of rules based on data manipulation operations like insert, detect, and update is performed to provide the DI. The domain, entity, referential and user-defined integrities are the four types of DI.

The DI is defined differently based on the field it is accomplished to, whereas the common meaning is, the data is highly accurate and consistent throughout its lifecycle. The data without accuracy affect the performance of the ML models. Thus, the data analysts mainly focused on improving the data quality and spending more time in data engineering processes. The feature engineering processes correct the bad data and replace it with good data by applying rules and policies. They are,

1. Eliminates the data-rows
2. Remove the missing values
3. Call data by index values
4. Replace missing values using a statistical representation

Various ML models are available for DI processes to solve the above-said challenges. Thus, the ML models are trained to predict the bad data instead of improving the data quality. In this paper, there are three ways the DI is given to improve the data quality. They detect and eliminate missing values, check the available data within the range and check if any data input belongs to different data types.

The bad data is handled by using non-critical decisions and critical decisions because the bad data generates errors and inaccuracy during prediction. It is impossible to use general decision-making procedures in all the systems, making irrelevant operations. So, for example, non-critical decisions are used in recommendation systems, and critical decisions are used in healthcare/medical systems. Also, the outputs obtained using the decision-making should be back up, resulting in complications in the solution (Fig. 3).

The missing value identifies the entity's integrity. That is if any input is unavailable or null during the inference time. It needs to be eliminated since the CDB cannot respond to

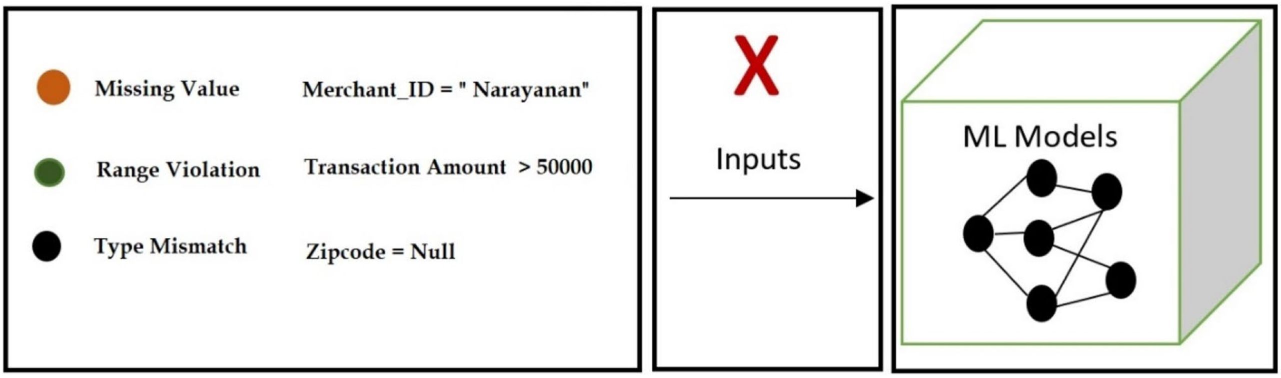


Fig. 3 Data integrity issues

the user queries. Some of the problems are considered and solved (see Fig. 4). They are:

1. Replacing the missing values by statistical metric is a challenging problem since it hides the issues behind the data issues. Regularly replacing the missing values make data drift and cannot provide accurate features for distribution. The data drift degrades the performance of the classifier and increases the delay.
2. Similarly, replacing with the out-of-range values using unique or approximate values creates a data drift and impacts model performance.
3. In certain cases of the missing values, the ML models provide an option to acquire the missing elements in the data to fill it, but it is not suitable for most of the cases.

For solving the above-said issues, an early warning system needs to be designed for immediate detection and solution. Figure 3 shows the DI issues like missing values, range violation, and data-type mismatch investigated in the input values feed before ML analysis.

1.7 Fetching the DI issues

The DI issues are caught by setting data checkpoints in the program. An invocation code is included in the ML implementation code to check the entire training or testing dataset to catch the DI issues. Adding the missing values, range violation, and type mismatch as features is a critical job. An efficient method for catching the DI issues includes a rule-set or policies that regularly check the input data before feeding into ML. These rules or policies are activated whenever a data input comes from CDB into the ML model.

The data checkpoints are included with the ML model as model inference. The proposed ML model is evaluated by analyzing a sample dataset as input. It also helps to assess the ML model based on the data and the rules or policies. The ML has an alert system that alerts the admin or the developer whenever it catches any DI issues. The data checkpoints might fix the error and make the developer understand the type of the DI issue. Undetected issues affect the performance of the ML model. Thus, the ML model should include the following:

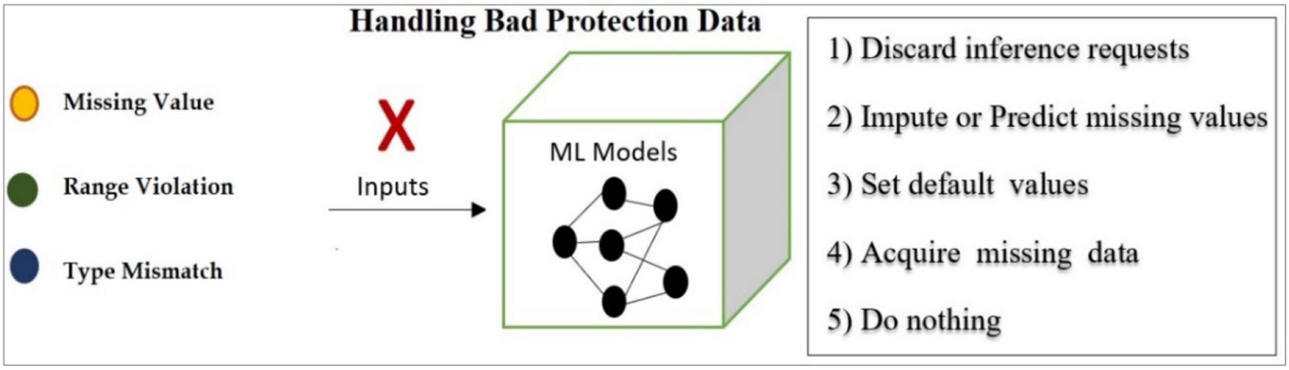


Fig. 4 Protecting data by handling bad data

Table 1 Domain integrity

Emp_ID	Emp_Name	Salary	Age
1	Sumathi	3,456,271	45
2	Loganathan	694,753	46
3	Parimala	3,945,462	38
4	Easwaran	583aqwe	36

Table 2 Entity integrity

Emp_ID	Emp_Name	Salary	Age
1	Sumathi	3,456,271	45
2	Loganathan	694,753	46
3	Parimala	3,945,462	38
	Easwaran	583aqwe	36

1.8 Local analyzation

A fine-grained ML model starts the best practices of the critical use cases for prediction analysis by including the ML inferences with the DI issues and verifying the ML model's impact. The ML model is created to understand the contextual nature of the data to identify the problems quickly. It means the ML model should behave like a black box model sometimes. Thus, the ML model becomes a time-consuming model. The model recreates all the factors based on the context of the data to catch the DI issues. The ML model finds it challenging to reproduce the output of the ML model that is not correctly versioned.

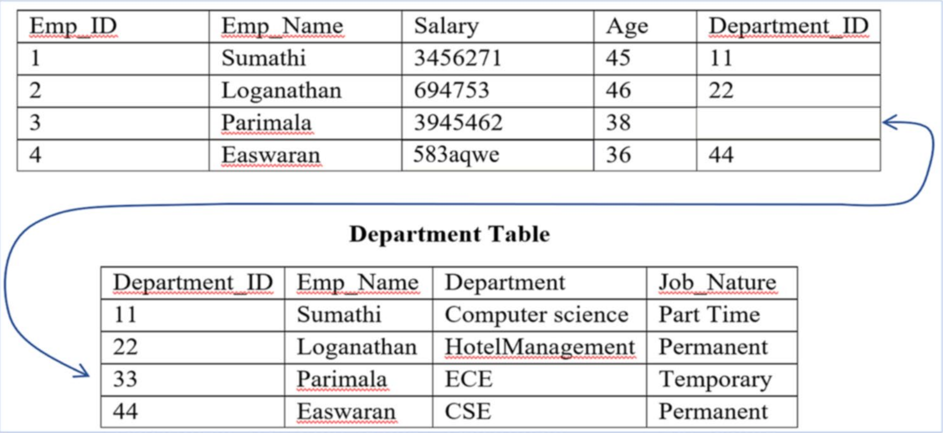
1.9 Global analyzation

The global analysis of the ML model is used to solve the severity of the DI issues. It includes analyzing the data features in a broader range to identify the issues at the creation time. The alteration of the data coincides with the product releases. Thus, queries triggered for data alteration can incorporate the DI issues to a particular code, and data release supports reverting it speedily. The DI issues are feed into the ML model as data drifts and, based on their impacts, a related drift in the output. Thus, a drift analysis model is incorporated for identifying the causes of DI issues. The above-said ways are only suitable for DI issues; else, it will create DI violations.

The local and global data analysis helps to assess and pinpoint the DI issues in the ML pipeline. Thus, the proposed ML model incorporated with the data-check-points to check the data, anomaly detection, and solve DI issues. For understanding easily and efficiently, the DI issues are illustrated in DB format in Tables 1, 2, and 3. For example, DI is provided in a CDB is by assigning identity for the customer and purchase tables of a business are (cust_table) and (purc_table). The cust_table has cust_ID, cust_Name, purch_ID, and the purc_table has purch_ID, purch_item. Both tables are related since the customer cannot purchase any items from the shop without a customer. It means the customer details are stored in cust_table, and purchase information is stored in purc_table. If a record is removed from the purc_table, the related data must be removed from the cust_table. Thus, the modifications of one table seriously affect the other related table. Thus, the DI issues need to be rectified in the following table data. For example, Table 1 shows the domain integrity, Table 2 shows the entity integrity, and Table 3 shows the referential integrity.

Initially, a new person should connect to the cloud and get a membership for participating in the cloud functionalities and perform well. Each user is connected with the cloud by creating a virtual IP address (VIP), using the following set of procedures.

Table 3 Referential integrity



Emp_ID	Emp_Name	Salary	Age	Department_ID
1	Sumathi	3456271	45	11
2	Loganathan	694753	46	22
3	Parimala	3945462	38	
4	Easwaran	583aqwe	36	44

Department Table

Department_ID	Emp_Name	Department	Job_Nature
11	Sumathi	Computer science	Part Time
22	Loganathan	HotelManagement	Permanent
33	Parimala	ECE	Temporary
44	Easwaran	CSE	Permanent

```
Connect($customer, VIP$)
$Get(\ u - id, pwd)$
$Put\ (permission\ given\ for\ connection)$
{
$Create\ aggreement\ with\ the\ security\ gate\ way$
$T = 0$
$\boldsymbol{While}\ (T\ < 25)\ and\ connect - status = unknown\ do$
        $Msg$ “$wait\ for\ connection$”
  $T = T + 1$
$\boldsymbol{End\ while}$
$If\ (T >= 25)\ then$
    $Msg$ “$connection\ failed$”
    $Msg$ “$repeat\ the\ connection\ process$”
$\boldsymbol{Else}$
    $Msg$ “$The\ connection\ established$”
$\boldsymbol{End}$
```

All the data are received from the user CDB, obtain the RAID. From the user profile and other information, any one of the files is downloaded and verified. If the file segments are correct and available in the user-profile-based DB, then the user and the data are valid. It says that the data is correct, complete and no data lost. For enhancing the DI, the CNN algorithm is used.

1.10 Data confidentiality

Data confidentiality is used here by encrypting and decrypting the user data using the RSA algorithm. One of the most authentic and reliable algorithms available in this highly technological world is RSA Algorithm (Rivest–Shamir–Adleman). It is a cryptographic and hybrid encryption scheme, widely used for specific security purposes and secure data transmission. It is also known as asymmetric cryptosystem, which uses two different mathematical linked keys, in which the receiver and sender use different keys for encryption and decryption purposes. One is a public key used for encryption, and the other is a private key used for decryption. The public key can be well known and shared openly with anyone as it is used to encrypt the message without the need to exchange the private key. This private key is secret only to the particular user who owns that key. After successful encryption, only the user who has the private can decrypt.

The RSA algorithm generates the above-said key generation method and makes it highly useful for secured data communication purposes, where it helps to maintain data confidentiality. Unlike symmetric algorithms, it does not use the same private key in both the encryption and decryption process. RSA algorithm is so special when compared to others because it generates keys using mathematical computations, as the key generation part is quite essential in this. The idea of the RSA algorithm is based on the complexity of factorizing the large integers, which are the product of two large prime numbers. Determining its original prime number from the factoring total is infeasible even in supercomputers.

The encryption strength depends on the key size usually represented in bits. These public key generations require mathematical computations of the pair, which is the crucial and complex part of the RSA algorithm.

1.11 Generation of public and private keys

- The large two prime numbers ***p and q*** are selected
- Then ***p and q*** have to be computed to find the multiplication of these prime numbers***, n=p. q*** (where n is the modulus). This number is used by both the public and private keys to provide the connection between them.
- So this public key has the modulus, *n,* and the public exponent, e, and choose a number e less than n so that we can calculate **φ(n)** where n is relatively prime **to (p − 1).(q − 1).** This means **e** and **φ(n)** has no common factor except one. This public exponent e need not be secretly shared prime integer, as the public key is given to anyone.
- So integer e has to be chosen such that ***1 < e < φ(n),*** and ***gcd(e,d(n)) = 1 and*** make sure that **e is coprime to φ(n).** The two integers' greatest common divisor(GCD) can be calculated using the Euclidean algorithm, as calculating prime numbers is tedious and time-consuming.
- The Private key consists of n(modulus), and the **d,** which is a private exponent, can be calculated using Euclidean algorithm, and integer d is computed using

$$\mathrm{d} = \mathrm{e}^{-1} \mod \varphi(\mathrm{n})$$

Thus, the asymmetric keys are generated for the encryption and decryption process. The public key consists of **e and n** if n is *p.q* and the plaintext m is encrypted using this public key ***e and n,*** and the private keys consist of d, and the ciphertext message c can be decrypted using this private key, ***d, and n***.

To encrypt and decrypt the information using the RSA algorithm, after computing the variables mentioned above, we use the below stated formula to get the ciphertext from the plaintext, and it is clear that the public key consists of ***n and e***, such that

For encryption: $\mathbf{c \equiv m^{e}}$ ***(mod n)***, m is the plain text, and m must be smaller than n here

And the private key has ***d and n***, and the below stated formula is used to get the plaintext from the ciphertext**, c,** which can only be decrypted by using the private key.

For decryption: $\mathbf{m \equiv c^{d}}$ ***(mod n)***, where c is the ciphertext

RSA serves great use in today's modern era in terms of data confidentiality and high security. Encryption strength increases gradually if we double the key length, and RSA keys are mostly 1024 or 2048 bits. So, this asymmetric algorithm is of high use where no one else can decrypt the data except the client, even if third-party individuals have access to it.

1.12 Asymmetric cryptography

1.12.1 Encryption

The Sender X knows the following:

- The receiver receives Sender X's public key.
- The plaintext message as an integer, m
- Compute the ciphertext by using the formula $\mathrm{C}=\mathrm{m}^{\mathrm{e}}$ (mod n)
- This computed C is then sent back this encrypted data to the receiver client.

1.12.2 Decryption

The Receiver Client knows the following:

- The receiver receives the encrypted data and decrypts it using the private key, computed by m = cd (mod n), and receives the original plain text, m.

The RSA-based encryption and decryption method provides data confidentiality in DIDC, where it fulfills the data-storage level and data-level securities in the cloud.

1.12.3 Proposed ML model

This section explains the internal architecture and the functionality of the machine learning algorithm accomplished for implementing and evaluating the DIDC. The proposed model comprises two major processing stages. In the first phase, the Convolution Neural Network (CNN) algorithm is used as the supervised learning algorithm and trained with multiple layers. The CNN uses multiple hidden layers, feed-forward ANN for enhancing the feature extraction capabilities. First, obtain all the data from the various cloud user, user data, and other IoT data and clone as traffic data (X). The user data and the data's meta-data are fed as the input (X) to CNN. The DI cleans and enhances the data quality and verifies the data while uploading and downloading from the CDB by the user. Based on the verification, the owner and malicious users are classified. If the user is malicious, they are completely rejected and deleted from the user table (Figs. 5, 6, 7).

The entire model of the CNN is illustrated in Fig. 8, which analyses three different data sets obtained from benchmark datasets publicly available and experimented with by earlier approaches. It has multiple layers at multiple levels.

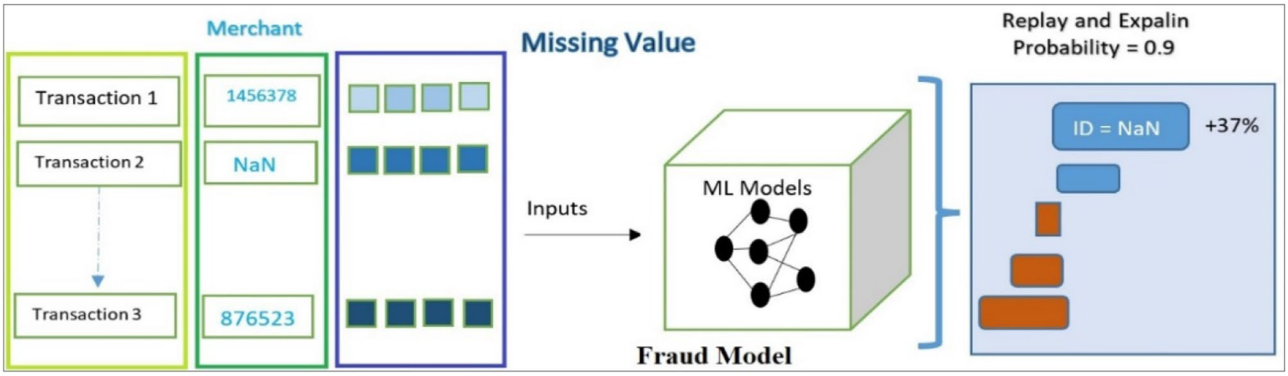


Fig. 5 Data check points for ML model

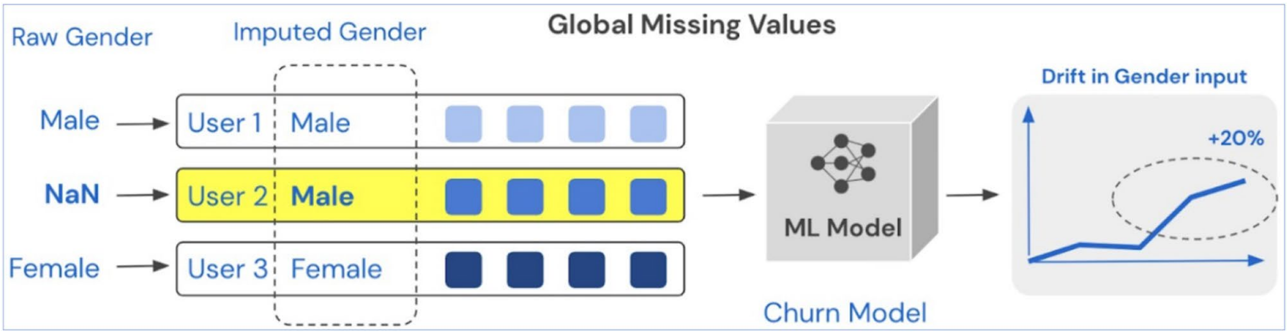


Fig. 6 Local and global analysis

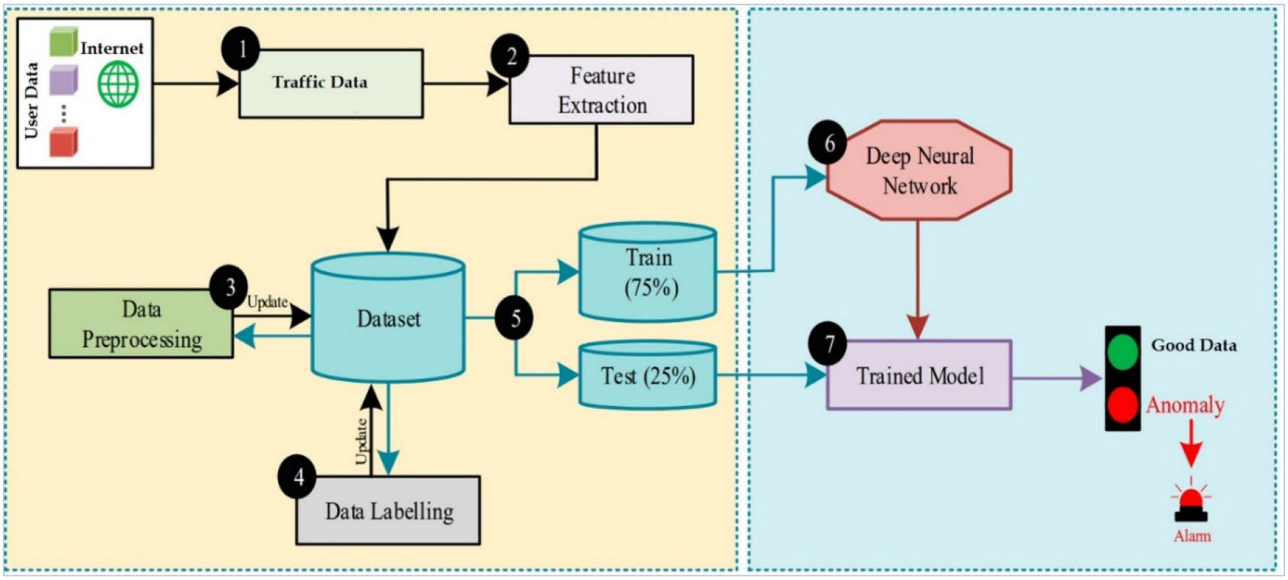
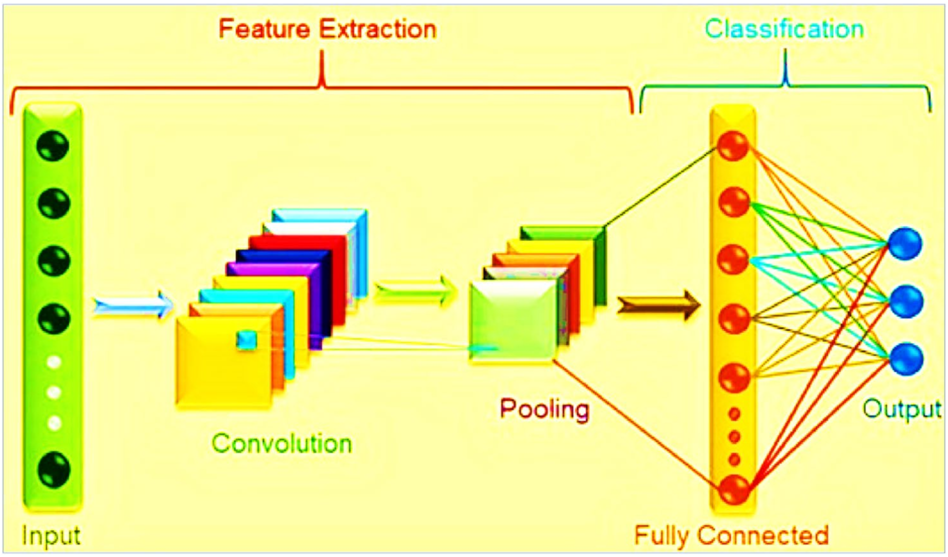


Fig. 7 Proposed architecture

Fig. 8 Convolution neural network model



Initially, the input data feed into the CNN, where the convolution layers (4 convolution layers) learn and extract the data (X) features and feed them to fee-forward hidden layers. The kernel size of the 4-CNN is 32nos, 64nos, 128nos, and 128nos, respectively. These layers learn the similarity of the features according to the context and syntax and reduce the dimensionality. The data and the feature values are normalized using weight values assigned to each x. The pooling layers act as the hidden layers. The fully connected layers summarize and obtain the data's final classes and deliver the output classes such as malicious user, bad data, and malicious storage.

In this paper, it is ensured that the model is trained well to increase the testing accuracy. The data investigation incorporated checkpoints, during the training process, for improving the data quality. It also eliminates the data validation loss. The testing process is applied to the unknown data to evaluate the results.

1.13 Data set

The proposed model has experimented on two different time-series (anomaly detection) datasets, such as Yahoo (Lavin and Ahmad 2015), NAB (http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html), NASA (Chollet 2015), and evaluating the same. In all the datasets, webserver logs, application information, and other additional cloud service provider information. Based on the data, the performance of the proposed method is evaluated. Both datasets are benchmark datasets used to evaluate the anomaly detection process to bring data owner verification for the DC. The data comprises user information, data information, meta-information of the data, data storage information, and cloud information. The existing SVM classifier (Krishnan et al. 2021a) used the Python Scikit Sklearn library (Krishnan et al. 2021b), whereas, in this paper, Keras Library (Tiwari et al. 2017) and TensorFlow are used.

2 Experimental results and discussion

In this paper, the JAVA programming language is used for implementing the proposed model in NetBeans IDE, where it has an in-built Cloud-Sim, which helps to stimulate the cloud functionalities. Java-NetBeans is installed in an intel-i7 Pentium system with 3.0 GHz processor speed, 8 GB RAM, and a 1 TB HDD. The dataset is experimented with the proposed programming module, computes few evaluation metrics to obtain the performance of the ML-DIDC. The execution is carried out by changing the dataset to examine the results. The evaluation metrics are referred from are:

True-positive correctly identifies and classifies the abnormal user/data as the abnormal user/data.

True-negative incorrectly identifies and classifies the abnormal user/data as the normal user/data.

False-positive correctly identifies and classifies the normal user/data as the normal user/data.

False-negative incorrectly identifies and classifies the normal user/data as the abnormal user/data.

Based on the above-said evaluation metrics, the performance factors of the classifier is calculated. They are:

Accuracy denotes the correctly classified data concerning the total number of data used in the calculation.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision, which represents how much data is predicted as positive from the existing positive classes, and written as

$$Precision = \frac{TP}{TP + FP}$$

The recall is a performance factor, which determines the completeness of the proposed model. The high recall says the lower FP, and lower recall says the high FP and is written as:

$$Recall = \frac{TP}{TP + FN}$$

From the experiment, the performance of the proposed ML-DIDC is evaluated by calculating various factors, such as time, F1-Score, CPU utilization, loss estimation, and accuracy comparison. First, the number of requests can be handled by the webserver based on the time is calculated. If the time increases, then the number of requests handled by the webserver is decreased. The result obtained from the experiment is shown in Fig. 9.

Then the CPU utilization is estimated according to the time. When time increases, the number of REQ-RES and application execution increases. Thus, CPU utilization is also increased concerning the time-based REQ, RES, and application functionalities. Sometimes like off-timings, the

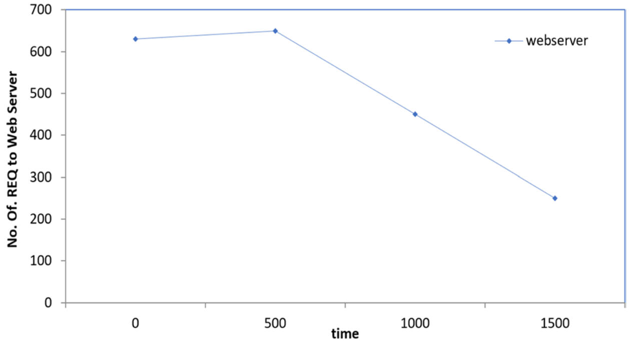


Fig. 9 Time efficiency w.r.t. number of requests

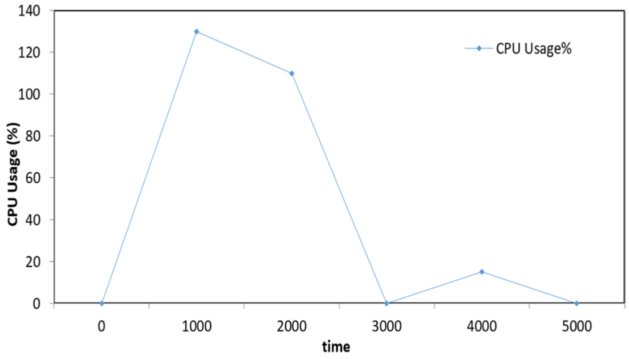


Fig. 10 Time vs. CPU utilization

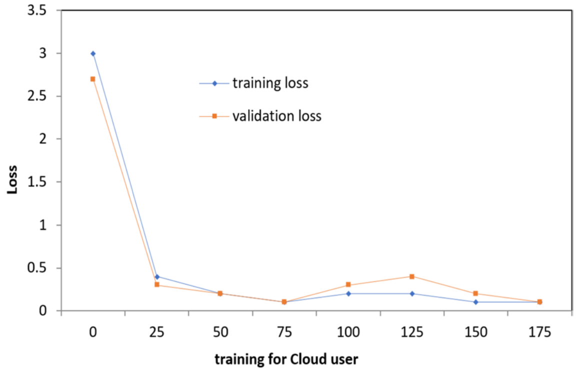


Fig. 11 Training loss vs. validation loss

applications will not be executed in the server or virtual machines. So, the CPU utilization is not constant, and it varies depending on the time. The obtained result is shown in Fig. 10.

The performance of the machine learning classifier is obtained by calculating the loss. The loss occurs due to the lower quality of data, delay in the REQ-RES, and the number of applications executed simultaneously in the webserver. It also depends on the virtual machine, resources availability, and resource allocation processes in the cloud. Loss may occur due to the route is busy and unavailability of the resources. The loss comparison is shown in Fig. 11. The loss decreased suddenly and was maintained constantly for an increasing number of users in the cloud. Finally, the accuracy obtained from the experiment is compared between the actual and predicted. The accuracy increases when the data or the number of users increases in cloud computing (Fig. 12).

The performance of the proposed ML-DIDC is compared with the existing algorithms regarding analyzing the log information. Table 4 shows the comparison, from that, it is identified that the proposed ML-DIDC obtained highest F1-Score than SVM, and LSTM. From the comparison, the predicted accuracy is merely equal to the actual accuracy, and thus, it is decided that ML-based DIDC can provide better security in cloud computing applications.

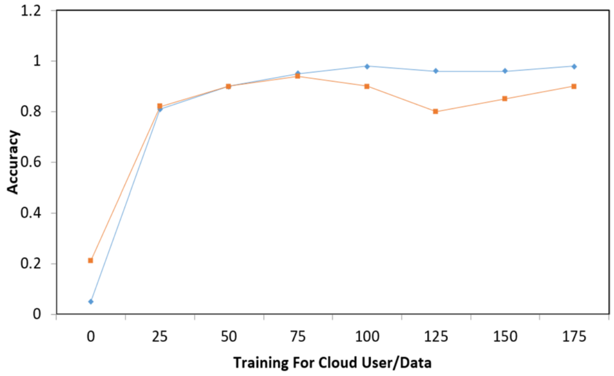


Fig. 12 Actual versus prediction accuracy

Table 4 F1-score comparison

Methods	F1-score (%)
One-class SVM	93
LSTM	89
Proposed ML-DIDC	98

3 Conclusion

The main objective of this paper is to provide data level and data storage level security for cloud applications. It can be done by providing data integrity and data confidentiality. Both the data integrity and confidentiality are not given by the earlier systems together, reducing the security level. This problem is taken as the serious problem of cloud computing, and this paper is motivated to design and implement a Machine Learning model for security provision. This paper incorporates DI and DC together to provide complete security in terms of data and storage. Since the number of users and the data size are enormous, the CNN model is used to analyze the user& data information to predict the abnormality. 80% of the data was taken for the training phase and 20% for the testing process from the whole dataset. The JAVA software is used for the experiment, and the experimental results show that the proposed MLC protocol outperformed the others and proved its efficiency regarding security provision.

In future work, some of the real-time cloud applications are taken for experimenting and verifying the performance of the ML-DIDC model.

Funding The authors received no specific funding for this study.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical standards The manuscript has not been submitted to more than one journal for simultaneous consideration. The manuscript has not been published previously. The Research not involved human participants and/or animals.

References

Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mane D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viegas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X (2015) TensorFlow: large-scale machine learning on heterogeneous systems. Software available from tensorflow.org. [Online]. https://www.tensorflow.org/

Alsheikh-Ali AA, Qureshi W, Al-Mallah MH, Ioannidis J (2011) Public availability of published research data in high-impact journals. PLoS ONE 6:e24357

Bajaj S, Sion R (2013) TrustedDB: a trusted hardware-based database with privacy and data confidentiality. IEEE Trans Knowl Data Eng 26:752–765

Balmford A, Crane P, Dobson A, Green RE, Mace GM (2005) The 2010 challenge: data availability, information needs and extraterrestrial insights. Philos Trans R Soc B Biol Sci 360:221–228

Chollet F et al (2015) Keras. https://github.com/fchollet/keras

Dumoulin J. Nasa http webserver logs. [Online]. http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html

Federer LM, Belter CW, Joubert DJ, Livinski A, Lu Y-L, Snyders LN, Thompson H (2018) Data sharing in PLOS ONE: an analysis of data availability statements. PLoS ONE 13:e0194768

Gherghina S, Katsanidou A (2013) Data availability in political science journals. Eur Polit Sci 12:333–349

Krishnan P, Jain K, Achuthan K, Buyya R (2021a) Software-defined security-by-contract for blockchain-enabled MUD-aware industrial IoT edge networks. IEEE Trans Ind Inf. https://doi.org/10.1109/TII.2021.3084341

Krishnan P, Jain K, Jose PG, Achuthan K, Buyya R (2021b) SDN enabled QoE and security framework for multimedia applications in 5G networks. ACM Trans Multimed Comput Commun Appl 17(2):1–29. https://doi.org/10.1145/3377390

Kumar RS, Saxena A (2011) Data integrity proofs in cloud storage. In: 2011 third international conference on communication systems and networks (COMSNETS 2011). IEEE, pp 1–4

Laptev N, Amizadeh S, Flint I (2015) Generic and scalable framework for automated time-series anomaly detection. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining (2015). ACM, pp 1939–1947

Lavin A, Ahmad S (2015) Evaluating real-time anomaly detection algorithms—the numenta anomaly benchmark. In: 2015 IEEE 14th international conference on machine learning and applications (ICMLA). IEEE, pp 38–44

Lee YW, Pipino L, Strong DM, Wang RY (2004) Process-embedded data integrity. J Database Manag: JDM 15:87–103

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in python. J Mach Learn Res 12:2825–2830

Puttaswamy KP, Kruegel C, Zhao BY (2011) Silverline: toward data confidentiality in storage-intensive cloud applications. In: Proceedings of the 2nd ACM symposium on cloud computing, pp 1–13

Ranganathan K, Iamnitchi A, Foster I (2002) Improving data availability through dynamic model-driven replication in large peer-to-peer communities. In: 2nd IEEE/ACM international symposium on cluster computing and the grid (CCGRID'02). IEEE, pp 376–376

Sandhu RS (1993) On five definitions of data integrity. DBSec. Citeseer, pp 257–267

Sivathanu G, Wright CP, Zadok E (2005) Ensuring data integrity in storage: techniques and applications. In: Proceedings of the 2005 ACM workshop on storage security and survivability, pp 26–36

Tiwari T, Turk A, Oprea A, Olcoz K, Coskun AK (2017) User-profile-based analytics for detecting cloud security breaches. In: 2017 IEEE international conference on Big Data (Big Data), pp 4529–4535. https://doi.org/10.1109/BigData.2017.8258494

Wu C-P, Kuo C-CJ (2001) Fast encryption methods for audiovisual data confidentiality. In: Proc. SPIE 4209, multimedia systems and applications III