Constraints-Relaxed Functional Dependency based Data Privacy Preservation Model

Satish B Basapur, B S Shylaja, *and* $Venkatesh$, *Member, IAENG*

***Abstract*— The data privacy preservation technique must ensure data integrity and prevent the invasion of confidential data from unsolicited or unapproved usage by any authorized or unauthorized user. Meanwhile, genuine users can use data for legal purposes. Confidential data should be excluded from data analysis. Further, the sensitive data resulting from data analysis should not be published if it breaches an individual's data privacy. Numerous methods such as *k*-anonymity, *l*-diversity, and *t*-closeness privacy models, encryption-based methods, and associative rule-based methods have been proposed in the literature to preserve data privacy. However, these methods have more data distortion and less data utility.**

The proposed approach scales down high-dimensional data by finding attributes correlation in the dataset through *Constraints-Relaxed Functional Dependency* (CFDs). If correlated attributes violate privacy according to user requirements or government policies, it finds a minimal set of correlated attributes to be obscured using *heuristic Minimal Vertex Forward Set* (MVFS) and encrypts such attributes using *block cipher* method. The proposed method minimizes the number of attributes to be obscured and enhances data usage while preserving information confidentiality.

All experiments are carried out using Apache Spark on a cloud environment with two different datasets: Heart-Disease, Income-Census (KDD) [39]. The experimental results show the number of attributes to be obscured under different configuration settings of CFDs for *Heart-Disease*, *Income-Census dataset*. The outcome of the experiment illustrates a correlation between attributes in the dataset. The results establish a relation between the number of attributes to be obscured and the level of information confidentiality.

***Index Terms*—Attribute correlation, Information-confidentiality, Block cipher, MVFS, Functional Dependency.**

I. Introduction

IN 2015, the Indian government launched an ambitious program called 'Digital India'. The program empowers people with access to e-governance services and revolutionizes a traditional society into a digitally empowered society, information, and a knowledgeable economy[1][2][3]. A program such as 'Digital India' has encouraged citizens to participate in e-governance, access the services in their locality, and perform digital business transactions. Similar programs such as e-Health-care, e-Commerce, e-Learning, etc. all over the world are successful due to recent advancements in communication technology, distributed computing, cloud computing, and communication devices (i.e., easy usage of smartphones, mobile phones device, small handheld devices, sensors, and actuators). People are directly or indirectly using either of the services; e-Governance, e-Business, e-Health-care, Social networks, Internet-of-Things, Social Internet-of-Vehicle, Autonomous vehicle, etc. With the rapid use of the applications mentioned above, there has been a data deluge and data exploding from these applications. The generated data is complex, huge in volume, and varied with structured, semi-structured, and unstructured formats. It is challenging to store, process, and analyze massive, complex big data. However, significant actionable insights could be extracted from big data if complex and massive data is analyzed promptly on time. The valuable intuition help to take up critical management decisions to enhance revenues in business. The analysis of results also helps us solve economic crises and scientific research problems and improve the services offered. In addition, it explores new opportunities in e-commerce, economy, and research [4][5][6]. Velocity, variety, and volume are the characteristics of big data [7]. Authors in [8] accentuated the significance of cloud computing platforms for processing big data applications. The authors in [9] proposed allocation strategies to enhance QoS and optimize resources in cloud environments for big data applications.

Data analytics give significant and exciting patterns and valuable knowledge from the massive amount of data. The discovered interesting patterns and valuable knowledge are beneficial. However, data analytics or data mining exposed the other side of the coin, particularly the data privacy of individuals. A data privacy breach occurs when private and sensitive information is disclosed. Publishing personal information for purposes other than the purpose of collecting data also results in a privacy breach. Privacy breach also occurs when the purpose of data collection and data analysis are different[10]. Data privacy preservation while data mining is a challenging issue in research and has gained a lot of attention in the research community[11]. Data owners, Data collectors, Data analyzers, and Decision-makers in the data life cycle play a vital role in data privacy preservation. Data owners generate data or own sensitive data. Data collectors collect data from data owners and give it to the Data analyzer to find patterns, trends, or significant value in the data. Decision-makers can make a decision based on the results of the analysis. The value of sensitive data can be determined based on the method proposed in [12]. In this paper, the data privacy model is to protect personal data from Data collectors, publishers, or data analyzers is designed. Data owners try to protect their data in the following situation.

- Data owner thinks that their data is very sensitive; disclosing such sensitive data may reveal critical infor-

Manuscript received May 28, 2021; revised Nov. 10, 2022.

Satish B Basapur is a research scholar and Assistant Professor in the Department of Information Science and Engineering, Dr. Ambedkar Institute of Technology, Bengaluru-560056, INDIA, e-mail: satish.basapur@gmail.com

B S Shylaja is a Professor in the Department of Information Science and Engineering, Dr. Ambedkar Institute of Technology, Bengaluru-560056, INDIA e-mail: shyla.au@gmail.com

Venkatesh is an Associate Professor in the Department of Computer Science and Engineering, University of Visvesvaraya College of Engineering, Bangalore University, Bengaluru-560001, INDIA e-mail: venkateshm.uvce@bub.ernet.in

mation to others. Therefore, the data owner does not want to publish the data. The data owner feels that the data collector may steal their sensitive data.
- Data owner knows that their data gives significant insights, and it is valuable to a data analyzer. Data owners may be willing to exchange their data for certain rewards or benefits.
- Data has to be protected from the data owner because the data owner may distort the sensitive data used by the analyzer to reveal true information.

The data privacy preservation technique must ensure data integrity and invasion of sensitive data from unsolicited or unapproved usage by any authorized or unauthorized user. The sensitive data should be excluded from data analysis, and sensitive data resulting from data analysis should not be published if it breaches the individual's data privacy [13]. Meanwhile, genuine users can use data for legal purposes.

In recent years, numerous methods have been proposed at various stages of the data cycle to safeguard data privacy. The paper reviewed the data privacy preservation methods based on encryption techniques in the big data storage stage. Further, this paper reviewed attribute and identity-based encryption techniques for data privacy preservation. In encryption-based data privacy preservation methods, sensitive data are encrypted using a well-known encryption algorithm and place the encrypted values in the cloud. In the data processing stage, knowledge discovery techniques adopted generalization and suppression approaches to combat data privacy breaches in the cloud. The classification, clustering, and association rule-based techniques preserve data privacy during the data extraction stage [14]. Trends, patterns in data, and the relationship between data is established using the association rule. Data privacy means protecting personal and sensitive information from invasion. From literature, it is found that a set of privacy models; k-anonymity privacy model to combat record linkage invasion, l-diversity data privacy model to combat both record linkage, attribute linkage invasion, t-closeness data privacy models to combat attribute linkage attacks, prevent probabilistic invasion. ϵ-differential data privacy model to avoid table linkage attack and probabilistic invasion [15]. Among all models, the k-anonymity-based privacy model is the most widely used model. However, the k-anonymity based privacy model will have more information loss or data distortion. Data utility is low in k-anonymity based privacy models, and trade-offs exist between privacy and data utility.

II. Literature Survey

In this section, extensive study and reviews on existing approaches on privacy preservation and research issues on data privacy in the context of Big data have been discussed. Many scholars have proposed methods based on cell generalization, clustering, classifications, encryption, and privacy preservation models for various big data life cycle stages. In recent years, a couple of researchers have proposed methods for preventing privacy breaches in a computing environment.

Today, advancements in wireless communication, innovations in cloud computing, ambient intelligence, big data, and wearable devices have leveraged health care systems. The wearable device collects bio-medical data and stores it in the cloud for further analysis. However, data sensitivity and openness in communication channels result in data privacy breaches. Therefore, ensuring data privacy preservation and reducing data distortion is a new noteworthy research issue.

It is paramount to ensure data privacy in wearable devices with knowledge discovery and data analysis requirements. Authors in [16] [17] proposed data aggregation and cooperative-based privacy preservation models for wearable devices. The proposed method is *MinHash* authentication scheme to combat data privacy breaches and find similarity patterns in data without disclosing sensitive data. Similar data is aggregated to preserve data privacy. The author has also proposed attribute encryption for access control of sensitive data. The problem of data privacy breaches in patient-sensitive data due to unauthorized access or excessive authorization is solved in [18].

An extensive research work has been carried out to preserve the privacy of patient-sensitive data in the healthcare sector. Further, comprehensive research work has been carried out to combat data privacy invasion on personal information in the network and cloud. The privacy preservation of sensitive personal data and maintaining the confidentiality of information is challenging for the research community. The statistical agencies must prevent data privacy breaches when data is stored in the cloud. The authors in [19] [20] designed a framework that enforces privacy preservation during data mining on the cloud. The author proposed hiding sensitive rules to balance data confidentiality and data disclosure with a legitimate user.

On cloud storage, data records are merged to extract useful information. While extracting knowledge, the confidential information accumulated may lead to the disclosure of sensitive data. An association mining with privacy-preservation (PPARM) model has been proposed to protect sensitive data from unauthorized access and exploitation. The proposed association mining with the privacy-preservation model determines a set of frequent attributes among data sets distributed vertically without disclosing sensitive data [21] [22]. Generally, privacy preservation is achieved through data anonymization, which anonymizes data and releases data sets in anonymized form. However, data privacy through data anonymization is strong enough to withstand adversaries. The adversary with supplementary information about a particular person in the dataset may begin an individual re-identification and capture sensitive data. An attacker uses permutation and combination to find the relationship between data set records and captures sensitive data through difference invasion. Data anonymization-based privacy models such as k-anonymity [23] and extensions of k-anonymity, l-diversity [24], and t-closeness [25] have been proposed in the literature. Similarly, authors in [26] designed a protocol that identifies encryption to guarantee a record-linkage attack. To overcome difference attacks, semantic-difference-based privacy models have been introduced, which analyze the magnitude of change that took place after the attacker's deduction was drawn on an individual earlier and after records of the individual were published. The semantic difference quantifies privacy loss mathematically.

In literature, there are several semantic-difference-based privacy models have been proposed for knowledge discovery, namely, *PrivBasis* algorithm [27], and *DiffFIM* algorithm

[28]. However, these privacy models suffer from clamorous results due to the large-scale data set. Knowledge discovery in medical records is paramount because it may disclose a correlation between symptoms and diseases. However, there are several infrequent lists of symptoms and diseases in data records. Therefore, use different values of thresholds to represent the frequency of an item. Authors in [29] developed an association rule that preserves privacy using different threshold values for records in the dataset. Similarly, Authors in [30] designed a methodology that performs global computation with minimal communication using association rules. The authors in [31] introduced a semantic layer in the Big data framework suitable for extracting semantic patterns from the data. The usefulness of the semantic layer in the big data framework for event detection, object detection, decision support, and reasoning is highlighted in [31]. Authors in [32] authenticate data owners, a user using bi-linear pairing with a random mask to combat data privacy breaches in the cloud. The privacy risk is determined using a sparse graph with fake edges and encrypting data using additive, somewhat homomorphic encryption [33] and *Spamdoop*[34]. The blockchain-based method is proposed to secure data sharing and restrain unauthorized access[35]. However, storing encrypted data and authenticating data providers or data used is insufficient to preserve privacy in the cloud. The data owners have to decrypt and process data at their location. However, it is not a feasible solution in a big data scenario where the dataset is massive in nature. Government regulation authorities impose stringent rules and penalties on cloud service providers or data collectors to disclose their users' sensitive data. The approach in this paper is to scale down high-dimensional data by finding data correlation through CFD. If correlated data breaches privacy according to user requirements, then such correlated data is obscured or anonymized, or discarded from the dataset.

III. Motivation and background

Several complex operations could be carried out on large-scale datasets on the cloud with privacy preservation constraints. However, data privacy preservation constraints while processing big data may yield more data privacy threats. Data processing activities such as data integration, aggregation, disambiguating records (i.e., entity resolution) across datasets, etc., may disclose individual sensitive data even though sensitive data is encrypted or anonymized or data is obscured form. The motivation factor is that instead of having a data privacy preservation method that guarantees data privacy as expected by the data owner, define privacy preservation policies that cloud storage service providers or statistical agencies or government agencies, or companies must follow to preserve the privacy of users from the initial stage to final stages of services.

In recent years, Constraints-Relaxed Functional Dependencies (with relaxation on comparison and coverage constraints) are widely used to find records matching, the correlation between records, and knowledge discovery in the dataset.

The proposed approach has leveraged Constraints-Relaxed Functional Dependencies to find data correlation that breaches user data privacy. Functional Dependencies with relaxation on some constraints find a correlation between data records across datasets on the cloud.

IV. Preliminaries and Problem definition

A. Relation Scheme R

Relational Scheme in Database D : Let R is relational schema defined on a set of attributes R=$\{A_1, A_2,..., A_n\}$, r is an instance of relational schema R and has set of tuples t. A tuple has set of attributes and a attribute has finite domain $dom(A_i)$, $A_i \in R$, $t[A_i] \in dom(A_i)$ and all tuple $t_i \in r$.

Functional Dependency : A Functional Dependency (φ) between two attributes sets A, B is represented as $A \rightarrow B$. Functional Dependency states constraint on set of tuples that give raise to instance relation $r \in R$, r: $A \rightarrow B$. For pair of tuples (t_1, t_2), constraint of Functional Dependency should be satisfied: if $t_1[A]=t_2[A]$ then $t_1[B]=t_2[B]$, if constraints are satisfied then set A be called LHS and set B be called RHS of FD φ. In this paper, the FD (φ) are free from *attribute-comparison*, *coverage constraint*, therefore, functional dependency FD is *constrints-free* FD φ.

The Attribute-comparison Constraints-Relaxed: use similarity or distance between attribute values in the range of predefined intervals(ε) instead of '*equality*' operator to compare attribute values.

The Coverage-constraints free: it specifies a subset of tuples that satisfies functional dependency constraint. The *coverage degree constraint* quantifies the number of the subset of tuples pairs that satisfy Functional Dependency φ(i.e., degree of satisfiability).

Given relational scheme r with set of attributes, the attributes defined over domains dom=$\{ dom_1, dom_2,... dom_n\}$ and constraint C_i on domain(dom) is given as follows.

$$\text{Dom}_c=\{ \text{ t } \in dom(R) \mid \Sigma_{i=1}^{k} C_i(t[A_i])$$

The constraints domain (Dom_c) is used to define coverage constraints on functional dependency.

B. Constraints-Relaxed Functional Dependency

The constraints-relaxed functional Dependency φ on relation scheme R is defined as follows.

$$[A_{\phi_1} \xrightarrow{\psi \epsilon \varepsilon} B_{\phi_2}]_{Dom_c} \quad (1)$$

The constraints domain is used to filter a set of tuples to get the subset of tuples. The ε represent threshold whose value lies between 0 to 1, the symbol A and B represent attributes and ψ_1, ψ_2 represent constraints defined on attribute A and B respectively (i.e. $A \cap B = 0$, $A, B \subseteq R$).

The instance of relation scheme r satisfies Constraints-Relaxed functional dependency (φ) if $\forall$ (t_1,t_2) ϵ r and $\phi_1 \in \phi$ is True then $\phi_1 \in \phi$ is always True. In other words, if $t_1[A]$ and $t_2[A]$ satisfies the constraints ϕ_1 then $t_1[B]$ and $t_2[B]$ satisfies the constraints ϕ_2 with degree ψ (i.e. degree of tuples dependency) greater than ε. For example, the database may contain authors with the same name, address, and affiliation. Therefore, a Constraints-Relaxed functional dependency φ is defined as:

Author$\rightarrow$ Address, Affiliation.

Further, the database may contain these attributes with different abbreviations, then a Constraints-Relaxed functional dependency φ is defined as:

$\text{Author}_{\phi_1} \rightarrow \text{Address}_{\phi_2}$, $\text{Affiliation}_{\phi_3}$
where ϕ_1,ϕ_2, ϕ_3, are similarity constraints. Moreover, affiliation may change as authors change affiliation during service or remain the same throughout. Therefore, Constraints-Relaxed functional dependency φ is redefined as follows.
$\text{Author}_{\phi_1} \xrightarrow{\psi(Author,Addr,Affil)\leq 0.02} \text{Address}_{\phi_2}$, $\text{Affiliation}_{\phi_3}$

For a given relational schema R, and an instance of relation scheme $r \in R$, Determine set of attributes $A=A_1,A_2....,A_n$ that can breach data privacy call them as data privacy breach attribute or confidentiality-violating attributes. Ensure that such data privacy breach attribute sets are not accessible to the attacker and encrypt data privacy breach attributes. In other words, the functional dependency that sensitive attribute as RHS value is made invalid.

C. Definition: Privacy

In the framework of personal data, data privacy means ensuring confidentiality and integrity of data. For example, user A should not know user Bs age, salary, account number, etc. If user A is adequate to disclose Bs personal information, then data integrity and confidentiality are breached and user Bs data privacy is at risk.

Typically, *R* is relational schema defined on a set of attributes $R=\{A_1, A_2,..., A_n\}$, *r* is instance of relational schema *R* and *t* is tuple representing a record of an individual. The projection of an attribute *B* or group of attributes *B* from a tuple represents sensitive data. To ensure data privacy of tuple *t* containing sensitive attribute, tuple *t* must satisfy: *t*[*B*] is in the obscured manner and a subset of attributes of tuple *t* that has not been declared as sensitive can be used in other operations but not for disclosing the sensitive attribute.

D. Definition: Attributes

The attributes of each transaction record in the dataset are *Identifier*, *Quasi-Identifie*r, *Sensitive* and *Non-Sensitive*.

- *Identifier* attributes are unique and shall be used to distinguish a record from other records in the dataset, For example: driving license number, mobile number.
- *Quasi-Identifier* attributes shall not identify a record in the dataset but it can be used to identify if they are linked with other external records. The identifier attribute values are removed and quasi-identifier attributes are used during data anonymization.
- *Sensitive* attributes are private, contain sensitive information. The sensitive attribute values are to be concealed. For example, disease, ATM pin number, passwords etc., The sensitive attributes are used extensively for data analytics or data mining but not for anonymization.
- *Non-sensitive* attribute value that can be disclosed and no need to protect data privacy.

E. Definition: Data privacy breach attribute set

Let *R* is relational schema defined on a set of attributes $R=\{A_1, A_2,..., A_n\}$, *r* is instance of relational schema *R*, let *A*, *B* be two attributes, $A, B \subseteq R$. Let *B* be set of sensitive attributes $B=\{B_1, B_2,... B_n\}$ and *A* be attribute that breach confidentiality of data if attribute *A* is not key and $B_i \in B$ is RHS of Constraints-Relaxed FD and hold on record *r* and *A* is on LHS (i.e. $A \rightarrow B_i$, where $B_i \in B$).

A relational schema *R* combat information confidentiality by ensuring that all sensitive attributes in an obscured and does not include attribute/s that violate confidentiality.

F. Problem Definition:

Any statistical agency or cloud service provider can use individual sensitive data for data analysis purposes to extract significant insights from it. But, without jeopardizing data privacy and not publishing for commercial gain or sharing individual sensitive data to unauthorized third parties or any individual. If the confidentiality of data is not maintained, then it is a violation of data privacy. Data owners have shared his/her sensitive data with the statistical agency or organization, entrusting them to protect data privacy from unauthorized access. Design a privacy preservation model that finds a minimal set of sensitive attributes to be obscured while preserving information confidentiality in big data applications. The minimal set of attributes is encrypted to enhance sensitive data confidentiality and use non-encrypted attribute values for knowledge discovery and analytics purposes.

V. Methodology

Preserving data privacy or information confidentiality through a cryptographic approach is computationally expensive because the volume and variety of data are massive. Moreover, it is an agonizing task to determine a set of sensitive data and attributes that derive these sensitive data in big data applications. In order to find the solution to this problem, this paper proposes an approach that minimizes the number of obscured attributes that enhance data usage while preserving data privacy or information confidentiality. The proposed approach identifies the correlation between attributes.

The relational databases use functional dependency to find the correlation between attributes and normalization. The extended form of functional dependency is Constraints-Relaxed functional dependency(CFD); it softens some constraints such as attribute comparison and coverage degree. The CFD extended form of functional dependency is used in this paper to identify correlation attributes. The CFD is used to find a set of attributes from which a set of sensitive attributes are derived. The given dataset is considered as relational scheme *R*, instance of a relation scheme is *r*. For a given database relation *R*, find a set of attributes that violate information confidentiality and obscure them from being accessible. For the given instance of a relation scheme is r, find LHSs of CFD that have sensitive data on RHS of CFD.

Section A describes the process of finding Constraints-Relaxed functional dependency(CFD) on relational scheme R. The set of algorithms designed in section B produces a set of CFD of form $Z \rightarrow A$, where *Z* is attributes on LHS that derive a set of sensitive attributes *A* on RHS.

A. Find Constraints-Relaxed Functional Dependency (CFD)

The Constraints-Relaxed functional dependency is determined by discovering a subset of tuples that have the same attribute values on RHS when they have the same attribute

value on LHS. It is accomplished by producing functional dependency candidates and checking for validity minimality. The validation process is performed by splitting the tuples based on LHS and RHS attribute values and then do the cardinalities of partition [36]. Finding Constraints-Relaxed FD means determining subsets of tuples that satisfy the constraints on the LHS attributes and constraints on the RHS attributes. Further, verify the LHS subset pattern contained in at least one RHS subset. First, generate subsets of similar tuples and then use these similar tuple subsets to validate CFDs. To generate similarity subsets, first, determine the difference matrix.

Definition: Difference matrix of an attribute: *r* is an instance of a relation schema *R* and it contains a set of attributes belonging to Attribute domain A (i.e., A_1, A_2,.. A_n ϵ A). Let δ be function of distance defined over the domain of *A*. The difference matrix for attribute *A* is a matrix M_A whose entry (*i*,*j*) contains the distance value $\delta(t_i[A], t_j[A])$ of the projections of tuples t_i and t_j on A.

Definition: Difference matrix of an attribute set: r is an instance of a relation schema *R*, consists of set of attributes $A_1 \epsilon$ A (i.e. A = $\{A_1, A_2 \ldots, A_n\}$ is an attribute set). Let Δ be distance functions defined over the domains of A_1, $A_2 \ldots, A_n$. The difference matrix for *A* is a matrix $M_{X\Delta}$ where entry (*i*, *j*) contains the n-tuple ($d_1, \ldots, d_n$), where $d_k = \delta_k (t_i[A_k], t_j [A_k])$, and t_i , t_j are tuples of *r* .

For example, consider the following sample dataset as shown in Table 1. Using definition of difference matrix of an attribute, the difference matrix for attribute *Height*, *Weight* and *Shoe Size* is computed and it is shown in Table 2.

TABLE I: Sample Dataset

Tuple No.	Height	Weight	Shoe Size
1	175	70	40
2	175	75	39
3	175	69	40
4	176	71	40
5	178	81	41
6	169	73	37
7	170	62	39

In the next step, use the difference matrix of the attribute and the difference matrix of the attribute set to find the similarity of tuples *t*. The similarity pattern of a tuple is used to find similarity patterns among a set of tuples.

Pattern with similarity: Let *r* be an instance of a relation schema *R*, consists of set of attributes $A_1 \epsilon A$. The difference matrix for attribute *A* is M_A and series of constraints on matrix M_A values is $\phi_i \epsilon \phi$. The pattern with similarity of a tuple on matrix M_A is represented by $\tau_A^{t_i}$. The $\tau_A^{t_i}$ is computed as $M[i_1,j_k]=(d_1, d_2..d_n)$ where d_i satisfy constraint ϕ_q, $\forall$ $q \epsilon [1, n]$ and k ϵ[1, h].

Pattern with similarity for subset: Let *r* is an instance of a relation schema *R*, consists of set of attributes $A_1 \epsilon A$. The difference matrix for attribute *A* is M_A and series of constraints on matrix M_A values $\phi_i \epsilon \phi$. A pattern with similarity for the subset is represented as S_x. The pattern with similarity and pattern with similarity for the subset is computed using difference matrix M_{Height}, M_{Weight} and $M_{ShoeSize}$.

For example, for the sample dataset shown in Table 1, a set of constraints are specified. For less than equal operator $\leq$, constraints (i.e. threshold value) for attributes *Height* and *Shoe Size* is set to 1, similarly, constraints for the attribute *Weight* is set to 10. The pattern with similarity for subsets is shown in Table 3.

TABLE II: Difference matrix for an attribute for Height(A), Weight(B), Shoe Size(C) and Height, Weight(D)

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 1 & 3 & 6 & 5 \\ 0 & 0 & 0 & 1 & 3 & 6 & 5 \\ 0 & 0 & 0 & 1 & 3 & 6 & 5 \\ 1 & 1 & 1 & 0 & 2 & 7 & 6 \\ 3 & 3 & 3 & 2 & 0 & 9 & 8 \\ 6 & 6 & 6 & 7 & 9 & 0 & 1 \\ 5 & 5 & 5 & 6 & 8 & 1 & 0 \end{pmatrix} \end{array}$$

A. Difference matrix for an attribute M_{Height}

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 0 & 5 & 1 & 1 & 11 & 3 & 8 \\ 5 & 0 & 6 & 4 & 6 & 2 & 13 \\ 1 & 6 & 0 & 2 & 12 & 4 & 7 \\ 1 & 4 & 2 & 0 & 10 & 2 & 9 \\ 11 & 6 & 12 & 10 & 0 & 8 & 19 \\ 3 & 2 & 4 & 2 & 8 & 0 & 11 \\ 8 & 13 & 7 & 9 & 19 & 11 & 0 \end{pmatrix} \end{array}$$

B. Difference matrix for an attribute M_{Weight}

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 3 & 1 \\ 1 & 0 & 1 & 1 & 2 & 2 & 0 \\ 0 & 1 & 0 & 0 & 1 & 3 & 1 \\ 0 & 1 & 0 & 0 & 1 & 3 & 1 \\ 1 & 2 & 1 & 1 & 0 & 4 & 2 \\ 3 & 2 & 3 & 3 & 4 & 0 & 2 \\ 1 & 0 & 1 & 1 & 2 & 2 & 0 \end{pmatrix} \end{array}$$

C. Difference matrix for an attribute $M_{ShoeSize}$

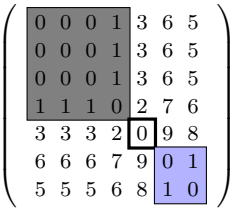
$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 0,0 & 0,5 & 0,1 & 1,1 & 3,11 & 6,3 & 5,8 \\ 0,5 & 0,0 & 0,6 & 1,4 & 3,6 & 6,2 & 5,13 \\ 0,1 & 0,6 & 0,0 & 1,2 & 3,12 & 6,4 & 5,7 \\ 1,1 & 1,4 & 1,2 & 0,0 & 2,10 & 7,2 & 6,9 \\ 3,11 & 3,6 & 3,12 & 2,10 & 0,0 & 9,8 & 8,19 \\ 6,3 & 6,2 & 6,4 & 7,2 & 9,8 & 0,0 & 1,11 \\ 5,8 & 5,13 & 5,7 & 6,9 & 8,19 & 1,11 & 0,0 \end{pmatrix} \end{array}$$

D. Difference matrix for attribute set $M_{Height,Weight}$

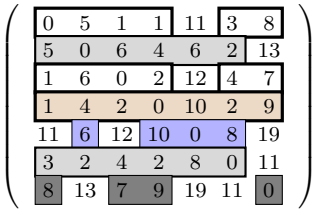
Pattern with similarity for Height = $\{1, 2, 3, 4\}_{1,2,3,4}$, $\{5\}_5$, $\{6, 7\}_{6,7}$ }; Pattern with similarity for Weight = $\{1, 2, 3, 4, 6, 7\}_{1,3}$, $\{1, 2, 3, 4, 5, 6\}_{2,6}$, $\{1, 2, 3, 4, 5, 6, 7\}_4$, $\{2,4, 5, 6\}_5$, $\{1, 3, 4, 7\}_7$; Pattern with similarity for Shoe Size = $\{1, 2, 3, 4, 5, 7\}_{1,3,4}$, $\{1, 2, 3, 4, 7\}_{2,7}$, $\{1, 3, 4, 5\}_5$,$\{6\}_6$.

The pattern with similarity for a subset of tuples is used to validate CFD. The proposed approach reduces patterns with similarity of subsets for the LHS attributes by discarding a single value (i.e., diagonal elements in the matrix) giving a set of striped patterns with similarity subsets. The set of striped patterns with similarity subsets are used to validate CFD based on the refinement of pattern with similarity subsets. The process of finding each pattern with a similar subset is contained in one of the subsets is called refinement. Thus, a Constraints-Relaxed functional dependency holds for the whole database only if a pattern with a similarity subset is contained in one of the subsets patterns with similarity.

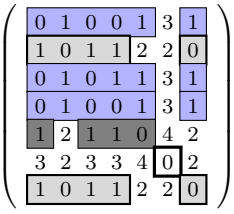
TABLE III: Difference matrix highlighting a set of pattern with similarity of subsets



A. Pattern with similarity for subset set : M_{Height}



B. Pattern with similarity for subset set : M_{Weight}



C. Pattern with similarity for subset set : $M_{ShoeSize}$

The set of constraints is specified for the sample dataset shown in Table 1. For less than equal operator $\leq$, constraints (i.e. threshold value) for attributes *Height* and *Shoe Size* is set to 1, similarly, constraints for the attribute *Weight* is set to 10. The Constraints-Relaxed functional dependency is defined as follows:

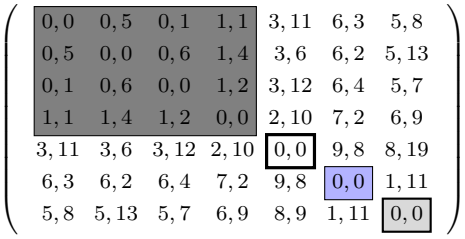
$$\{\text{Height}_{\phi_1}\ , \text{Weight}_{\phi_2}\} \rightarrow \text{Shoe Size}_{\phi_2}$$

The cfd holds for the whole dataset. Thus, refinement on pattern with similarity subsets for attribute Height,Weight = $\{1, 2, 3, 4_{1,2,3,4}\}$ refinement on pattern with similarity subsets for Shoe Size = $\{\{1, 2, 3, 4, 5, 7\}_{1,3,4}$, $\{1, 2, 3, 4, 7\}_{2,7}$, $\{1, 3, 4, 5\}_5$ $\}$, a pattern with similarity of {Height,Weight } is contained in pattern with similarity of Shoe Size. The pattern with similarity subset is contained in one of the subset pattern with similarity is shown in Table 4.

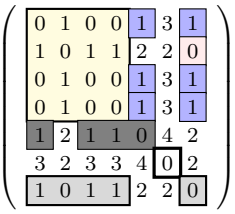
B. Constraints-Relaxed Functional Dependency Algorithm

For a relation scheme *R*, the Constraints-Relaxed functional dependency is constructed by visiting a lattice structure. All *k*-combinations of attributes of the relation scheme are considered while constructing functional dependency. For every iteration, k-attributes are chosen in the range from 2 to the number of attributes of the relation. Only an unpruned set of attributes at each iteration are considered for constructing CFDs. *r* is an instance of a relation schema *R* and contains set of attributes belonging to attribute domain A (i.e., A_1, A_2,.., $A_n \epsilon A$). The difference matrix for attribute *A* is M_A and series of constraints on matrix M_A values $\phi_i \epsilon \phi$. A pattern with similarity for the subset is represented

TABLE IV: The pattern with similarity subset is contained in another subset with similarity pattern



A. Pattern with similarity of subsets for Height and Weight



B. Pattern with similarity of subsets for shoe size, submatrix with yellow color subset indicates inclusion property.

as S_x. The process of generating a subset of pattern with similarity for the attribute is given in algorithm 1.

The subset of pattern with similarity for the attribute is computed using difference matrix M_{Height}, M_{Weight} and $M_{ShoeSize}$ is shown in figure 2.

Algorithm 1 accept relation scheme *r*, attribute and constraints (ϕ_A) on an attribute *A* and return subsets of pattern with similarity for attribute. For each tuple of relation *r* (i.e., $t_i \epsilon$ r) pattern with similarity of a tuple is computed using Algorithm 2 (Pattern with Similarity given Constraint). The pattern with similarity of a tuple on matrix M_A is represented by $\tau_A^{t_i}$. The $\tau_A^{t_i}$ is computed as *M[*i_1*,*j_k*]=(*d_1*, *d_2*..*d_n*)* where d_i satisfy constraint ϕ_q, $\forall$ *q* ϵ*[1, n]* and *k* ϵ[1, h]. The computed pattern with similarity of a tuple $\tau_A^{t_i}$ and tuple t_i is assigned to I_A (Line 4, 5). Algorithm 2 reduces pattern with similarity of subset for the LHS attributes by discarding single value (i.e., diagonal elements in matrix). Algorithm 2 also computes subset of stripped pattern with similarity using a difference matrix on pairs of tuples.

Algorithm 2 accepts relation scheme *r*, attribute *A*, constraints ϕ_A, tuple t_i and returns pattern with similarity of a tuple on matrix M_A $\tau_A^{t_i}$. In algorithm 2, every tuple that belongs to *r* (i.e., $t_j \epsilon$ r) are used to find tuple t_i, t_j values difference for given attribute *A* (Line 3). The difference between tuple t_i, t_j values is computed as per the constraints ϕ_A (Line 4) and the difference value is added to $\tau_A^{t_i}$ when it satisfies the constraints ϕ_A (Line 4-6).

Algorithm 3 computes a subset set that has pattern with similarity for given an attribute set. It uses I_A , I_X and computes $I_X \bigcup A$. For a given pattern with similarity of I_X, it finds matching set of tuples using the function RetriveTuple(Line 3). In next step, for every tuple t_i it finds pattern with similarity $\tau_A^{t_i}$ from I_A using Algorithm 1 (i.e. Subset of Pattern with Similarity)(Line 5). To find a subset of pattern with similarity for a given set of attributes, it performs the intersection of I_x and I_A (Line 6,7). Finally, intersected I_x and I_A set is added to tuple t_i(Line 8).

The obtained set of striped patterns with similar subsets is

used to check the validity of Constraints-Relaxed functional dependency with the help of the refinement process on patterns with similar subsets. The process of finding each pattern with a similarity subset is contained in one of the subsets of patterns with similarity is called refinement. Thus, a Constraints-Relaxed functional dependency holds for the whole database only if a pattern with a similar subset is contained in one of the subset patterns with similarity.

Each striped pattern with similarity subset $\widehat{I_X}$ refines another striped pattern with similarity subset $\widehat{I_{X \bigcup A}}$ if each striped pattern with similarity subset $\widehat{I_X}$ is included in one of the subset of $\widehat{I_{X \bigcup A}}$. The pseudo-code for validating Constraints-Relaxed functional dependency is given in Algorithm 4.

Algorithm 1 Subset of Pattern with Similarity

Input: Relation *r*, Attribute *A*, Constraint:ϕ_A
Output: Subsets with pattern similarity, I_A

```
Initialize I_A=0, similar=0
for every tuple t_i ∈ r do
    similar=PatternWithSim(r, A, t_i, φ_A)
    I_A=I_A ⋃ similar (i.e τ_A^{t_i})
    Assign(similar,t_i)
end for
```

Algorithm 2 Pattern with Similarity given Constraint

Input: Relation *r*, Attribute *A*, t_i, Constraint:ϕ_A
Output: Pattern similarity subset set that satisfies Constraint

```
initialize difference=0, similar=0
for Every tuple t_j ∈ r do
    difference=δ_A(t_i[A], t_j[A])
    if Satisfies(difference, φ_A) then
        add j into similar (i.e τ_A^{t_i})
    end if
end for
return(Pattern similarity subset set that satisfies Constraint)
```

Algorithm 3 Pattern with Similarity Subsets;attribute set

Input: Set I_X, Set I_A
Output: set of subsets with pattern similarity

```
Initialize similar=0, similar_A=0, assigTuples=0, I_{X⋃A}=0
for Every similar_X set ∈ I_X do
    assigTuples=RetrieveTuple(similar_X, I_X)
    for Every tuple t_i ∈ assigTuples do
        similar_A=Subset of Pattern with Similarity(t_i, I_A)
        similar=similar_X ⋂ similar_A
        I_{X⋃A}=I_{X⋃A} ⋃ similar
        Assign(similar, t_i)
    end for
end for
```

C. Heuristic approach to finding minimal set Z

To minimize the number of attributes to be encrypted and preserve information confidentiality find minimal set Z . This paper uses a heuristic approach to find minimal set *Z* and the process of finding minimal set Z is described in this section.

Algorithm 4 Validity of Constraints-Relaxed FD

Input: Set I_X, Set I_A, relation scheme *r*
Output: validity of Constraints-Relaxed functional dependency

```
get Î_A=StripSimPat(I_A)
get Î_{X⋃A}=StripSimPat(I_{X⋃A})
for each Î_X do
    if (disjointness(Î_A,Î_{X⋃A})) then
        return invalid cfd
    else
        return valid cfd
    end if
end for
```

The process of finding minimal set *Z* is formulated as finding a solution to the minimal feedback vertex set (MFVS) problem. The solution to the feedback vertex set problem is determined by identifying a subset of its vertices (i.e. feedback vertex set) that form a cycle in the graph and the removal such vertex set makes the graph as acyclic[37]. This paper adopts an approach that is used to solve the MFVS problem to find a minimal set *Z*. The process begins with identifying attributes that violate information confidentiality and belong to the LHS of CFD. Assign weight (number of times) to each attributes that violate information confidentiality and belongs to LHS of CFD. Further, rank is assigned to an attribute (i.e., belongs to a confidentiality-violating attribute set) based on the number of times it has appeared. Eventually, eliminating confidentiality-violating attributes in decreasing order of attributes appeared.

In the worst case, if the LHS of CFDs has a single attribute, then eliminate such attribute without assigning rank.

For example, consider the following two CFDs. The attribute *B* is sensitive attributes
CFD 1: {A, N, G, O}→ {B}
CFD 2: { A, F, L}→{B}

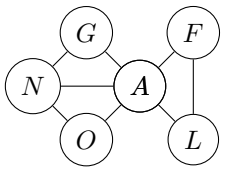


Fig. 1: CFDs with confidentiality-violating attribute

In the above example, assign weight to each attribute that appears on LHS of CFDs, attribute *A* has a weight 2 and all other attributes have weight 1 (i.e., N, G, O, M, D, F, L). Next, arrange weight of attributes in decreasing order. Eventually, eliminate the attribute *A*. The elimination of attribute *A* makes CFDs invalid and the graph becomes acyclic. Moreover, the sensitive attribute on RHS of CFDs is not accessible. To preserve information confidentiality, attribute *A* and attribute *F* are encrypted. The CFDs obtained for the sample dataset considered in Table 1 are as follows.

$$\{\text{Height , Weight}\} \rightarrow \{\text{Shoe Size }\}$$

The CFD has confidential-violating attribute set:{*Height*,*Weight*} and sensitive attributes {*shoe Size*}. To preserve information-confidentiality, attributes{*Height*,*Weight*} on LHS and attributes {*shoe Size*} on RHS is encrypted. The process of encryption is described in the next section.

D. Encryption of Confidentiality-violating Sensitive Attributes

The minimal set *Z* obtained in the previous sections are encrypted using block cipher[38]. The minimal set *Z* contains the set of confidentiality-violating attributes and sensitive attributes. The *k* bits key is used to encrypt 64 bits of plain text and produce cipher text. The process of encryption is specified as function *f*

$$f_K(p) : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$$

The key *K* has length of k and *n* is length of data block to be encrypted. The set of sensitive attributes, confidentiality-violating attributes set *Z* on LHS of CFDs and sensitive attributes on RHS of CFDs are encrypted using different key *K*. The key used to encrypt plain text and decrypt cipher text is shared with the data owner. For example, the sample

TABLE V: Sample customer dataset

Tuple No.	Hieght	Weight	Shoe Size
1	**	**	**
2	**	**	**
3	**	**	**
4	**	**	**
5	178	81	41
6	169	73	37
7	170	62	39

dataset considered in Table 1, it is found that first four rows of *height* and *weight* to be confidential, the attribute *height* and *weight* are confidentiality-violating attribute set and *shoe size* is sensitive attribute. Table 5 shows some attribute values are encrypted using the proposed methodology and the values represented '**' are encrypted values to preserve the information-confidentiality.

E. Robustness of Proposed Methodology

In this section, the critical analysis is performed to check the robustness of the proposed methodology against an attacker who tries to disclose values of confidential attributes. The Constraints-Relaxed functional dependency(CFDs) relaxed some constraints; quantity of attributes on LHS of CFDs and degree of similarity. For example, functional dependency φ: X $\rightarrow$ Y defined on instance of database relation scheme *R* is constraint-free functional dependency, but functional dependency $\varphi^{'}$: A $\rightarrow$ Y, A ϵ X defined on instance of database relation scheme *R* is not constraint-free functional dependency. Thus, the set of CFDs on the unencrypted relation scheme *R* makes the attacker to disclose encrypted attribute values.

Case I: Let us consider a dataset that has an information-confidential attribute (say attribute *D*). As per the methodology, find CFDs with attribute *D* on RHS on the dataset and mask the attribute value of *D* using encryption. That is, find CFD φ: AB $\rightarrow$ D and obscure tuple t_1[D] by encryption. If the tuple t_1[D] is encrypted, attacker can infer value of encrypted attribute through the attributes similarity between two or more tuple (say t_1 and t_2) . To solve this problem, the CFD φ: AB $\rightarrow$ D ascertain to obscure both tuple t_1[A] and tuple t_2[A]. Because, the CFDs, φ_1: A $\rightarrow$ D and φ_2: B $\rightarrow$ D are not validate on dataset and obscure attribute value of *A* or *B*, attacker cannot disclose encrypted attribute values by knowing value of *A*/*B*.

Case II: Consider the scenario, where attribute *B* is not obscured but attribute *A* is obscured to invalidate the correlation between attributes that is represented by φ: AB $\rightarrow$ D. Now, the attacker observes all the values of attribute *B* in all tuples of the dataset and tries to infer the information-confidential attribute value of *D*. To avoid such an attack by the attacker, obscure both attribute *B* attribute *A* to invalidate the correlation between attributes that is represented by φ: AB $\rightarrow$ D.

To guarantee information confidentiality, the proposed methodology must ensure that there is no violation of CFDs by a confidential attribute which is obscured. For confidential attribute *A* and masked tuple *t*[A], the attacker can infer the value of confidential attribute *A* if one of following condition is satisfied.

- By observing minimal set Z of CFDs on the unencrypted dataset and the partially encrypted dataset the attacker infer the value of the confidential attribute.
- There exists cfd φ_1: X $\rightarrow$ A that does not hold on instance of relation scheme *r* but hold on tuple *t* and projection of set of tuples *s* on attribute *X* give similar value of t[X]. All value of *A* is same as tuple value in *t*.

VI. Results and Performance Analysis

A. Language and Hardware used for implementation

Python code is written to find a set of CFDs on datasets, and all CFDs obtained after executing the code are analyzed. The pseudo-code for constructing CFDs is given in Algorithm 1, Algorithm 2, Algorithm 3, and Algorithm 4. Python code is also written MFVS that selects attributes to be encrypted. The paper used standard Block cipher algorithm to encrypt attribute values. All experiments are carried out on an Intel i7 processor at 2.34GHz, 32GB RAM, 6 GB NVIDIA graphic card, Windows 10 operating system, and Python 3.8(Anaconda 3) on the Apache Spark platform.

B. Dataset Used

All experiments are carried out on the Heart-Disease dataset and Census-Income (KDD) dataset[39] and conducted on Apache Spark, an open-source framework for big data applications. Apache Spark provide *driver program* to begin execution with main module, *processing nodes* for parallel execution of CFD algorithms, CFDs formation based on similarity, coverage threshold, attribute encryption and *memory abstraction* for sharing data set. Apache-spark allows us to create Resilient Distributed Dataset after the spark session.

The heart-disease dataset contains 76 attributes, subset of 14 attributes are used for experiments (i.e., *age, chest-pain-location(cpLocation), chest-pain-type(cptype), blood pressure, cholesterol, sex, ecg results, slope, rest-wall(rwall),*

vessels, thal, dnum, maximum-heart-rate(mhrate), exercise-induced, heart-condition(hcondition) etc.,). The Census-income dataset contains 40 attributes relevant to the income of a person, subset of 26 attributes are used for experiments (i.e., *age, class-of-worker, occupation-code, education(edu), wage-per-hour, sex, total-person-earnings, country, citizenship, family-members, weeks-workload, total-person-earnings(totpearning)*, etc.,). Table 6 highlights the number of attributes, number of instances, dataset size, and #CFDs for the given dataset.

TABLE VI: Dataset details

Dataset.	#Attributes	#Instances	Dataset(MB)	#CFDs
Heart-Disease	14	303	60	15
Census-Income	26	9762	103	241

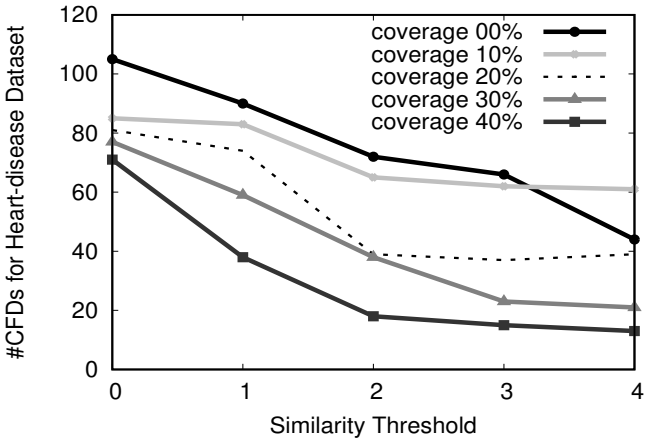


Fig. 2: # CFDs for Heart-Disease Dataset

C. Scenarios of Experiments

The proposed methodology is used to find a minimum number of attributes to be encrypted to preserve information confidentiality. The set of CFDs is obtained for the given dataset according to attribute similarity and coverage threshold value. The sample set of CFDs for Census-Income and Heart-disease datasets are given in Tables 7 and 8.

For the conduction of experiments, this paper supposed attribute to be confidential and derived a set of CFDs for each supposed attribute. Further, the minimum of attributes to be encrypted is derived using *heuristic MVFS* and encrypted using *Block cipher* method. In the experiment, the proposed CFDs algorithm is used to find a set of CFDs according to attribute similarity and coverage degree and *heuristic MVFS* algorithm is used to evaluate the minimum number of attributes to be obscured to preserve information confidentiality. Under different configurations of constraints (i.e., attribute similarity and coverage degree), the experiments are conducted to find a set of CFDs and a minimum of a set of attributes to be encrypted from the set of CFDs. Senario I, both constraints *attribute similarity* comparison and *coverage degree* are considered. Scenario II, considered a set of CFDs with constraint-free on coverage range and reduced value of attribute similarity comparison, setting threshold to 10% on attribute similarity. Scenario III, considered a set of CFDs with constraint-free on attribute comparison and reduced value of coverage range, the threshold value is set to 1 for every attribute. Scenario IV, considered a set of CFDs with constraint-free on attribute comparison and coverage range. The set attributes are encrypted to preserve information confidentiality, and the set of attributes to be encrypted certainly increases as constraints on functional dependencies are relaxed or flexible. Removing constraints on functional dependencies will yield more correlated attributes.

D. # CFDs for Census-Income, Heart-disease Dataset

Table 7 and Table 8 show the set CFDs discovered for the Census-Income, Heart-Disease dataset. The attribute similarity and coverage threshold have a greater influence on the number of CFDs. It is noticed from listed CFDs that attribute sex, age implies occupation, similar values of attributes age, sex, fname,lname, edu imply similar value of occupation. It is also observed from Table 8 that similar values of *age*,*sex* imply *heart-rate*. Further, similar value of attribute `age`, *bp*, *cholesterol*, *dnum* imply *heart-rate*. Another cfd is the hybrid CFDs that uses attribute {cpLocation, cptype, bp, cholesterol} to determine { hrate}. It indicates that heart rate(hrate) is derived using attributes {cpLocation, cptype, bp, cholesterol}. It also indicates that, whenever the cpLocation, cptype, bp, cholesterol} values are similar, then the heart rate is also similar.

TABLE VII: Sample CFDs for Census-Income Dataset

CFDs	Similarity Threshold	Coverage Threshold
{age,sex → occupation}	0	0.2
{age, country→ wages/hour}	0	0.2
{age, fname,lname→ wages/hour}	0	0.2
{age, sex,edu → wages/hour}	1	0.1
{age, sex, fname,lname, edu → occupation}	1	0.1
{age, class,country → wages/hour}	1	0.3
{age, class, totpearning → totearnings}	1	0.3
{age, citizenship, edu,fmemb → totearnings}	1	0.0
{edu, age, wages, totpearning → totearnings}	1	0.2
{edu, age, wages, workload → totearnings}	2	0.2
{age, edu, class → occupation}	2	0.4
{country, citizenship,edu → occupation }	2	0.4

TABLE VIII: Sample CFDs for Heart-Disease Dataset

CFDs	Similarity Threshold	Coverage Threshold
{ age,sex → heart rate }	0	0.2
{ hrate, cplocation, cptype → hcondition }	1	0.1
{ age, sex, exercise → cptype }	1	0.1
{ age, ecg,vessels → hcondition }	1	0.3
{ age, ecg,vessels → hcondition }	1	0.3
{cpLocation, cptype, bp, cholesterol → hrate}	1	0.0
{sex, age, bp, cholesterol → hrate}	1	0.2
{age, bp, cholesterol, dnum → hrate}	2	0.2
{age, rwall, vessels → hrate}	2	0.4
{bp, cholesterol, ecg, slope → hrate }	2	0.4

Figure 2, Figure 3 represent the number of CFDs obtained from the *heart-disease*, *Income-Census* dataset respectively under different configuration settings on constraints. The line in the graph represents different threshold values of coverage extent. It is observed from figure 2 and figure 3 that relaxation on coverage extent results in more CFDs.

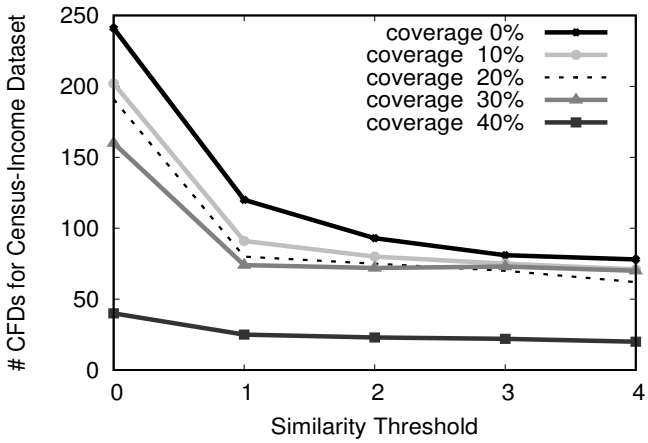


Fig. 3: # CFDs for Income-Census Dataset

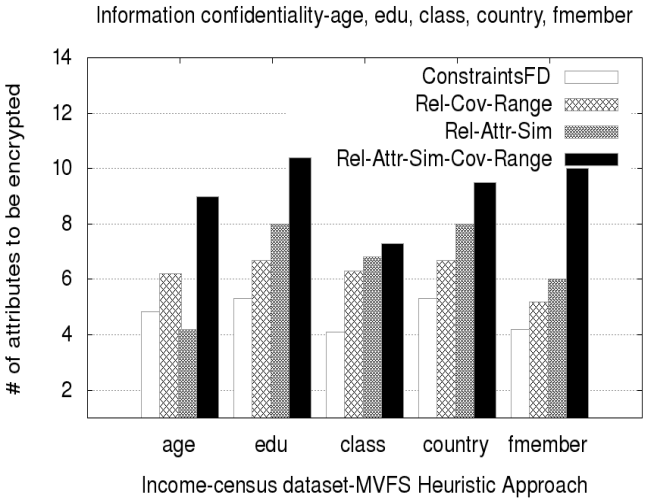


Fig. 4: CFDs with constraints settings *vs*. #attributes obscured

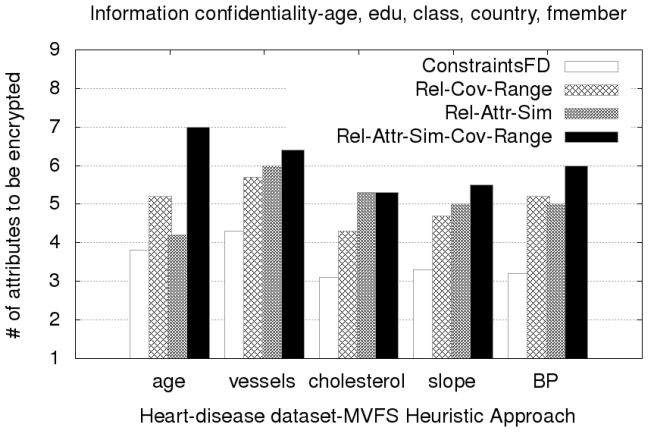


Fig. 5: CFDs with constraints settings *vs*. #attributes obscured

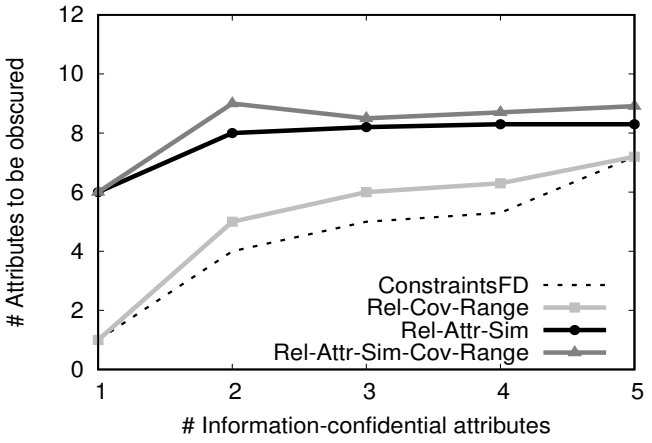


Fig. 6: IC *vs.* # attributes to be obscured:Heart-Disease

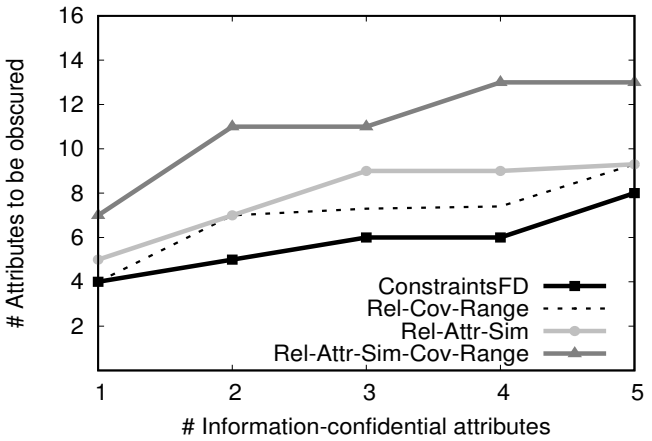


Fig. 7: IC *vs.* # attributes to be obscured:Income-Census

Moreover, a maximum number of CFDs are discovered with a coverage extend threshold value slightly greater than 0% but less than 20%. It is evident from figure 2 that when the attribute similarity threshold value is set to high, then it certainly increases the number of potential CFDs but newly discovered CFDs will invalidate all ready discovered CFDs. Therefore, the number of discovered CFDs is less as attribute similarity and coverage extent threshold values are increased.

It is observed from Figure 2 and Figure 3 that a change in attribute similarity threshold value from 0 to 4 results in a decreased number of discovered CFDs. It is because when the attribute similarity threshold is set to a '*zero*' threshold, then many attributes value are not the same as other attribute values on the LHS attribute, which results in more dependencies. Later, when the attribute similarity comparison threshold value increases, many CFDs are invalidated. When the attribute similarity threshold value is set higher, it might result in new CFDs due to similarity in LHS value. However, it also invalidates existing CFDs due to dissimilarity in RHS attributes.

The number of attributes to be obscured under different configuration settings on CFDs for *Income-Census* and *Heart-Disease* datasets are shown in Fig. 4 and Fig. 5, respectively. The minimal set Z of the attribute to be obscured is determined using *heuristic MVFS*. During experiments,

constraints on functional dependencies are specified using *ConstraintsFD*, *Rel-Cov-Range*, *Rel-Attr-Sim* and *Rel-Attr-Sim-Cov-Range*. The *ConstraintsFD* indicates no constraints on functional dependencies, The variable *Rel-Cov-Range* indicate constraint relaxation on coverage extent, *Rel-Attr-Sim* indicate constraint relaxation on attribute similarity comparison and *Rel-Attr-Sim-Cov-Range* indicate relaxation on attribute similarity comparison and coverage extent.

It is noticed from Figure 4 that number of attributes to be obscured are different under different configuration settings CFDs for *Income-Census* dataset. The intuition from figure 4 is that the relaxation of constraints on CFDs has a direct impact on the number of attributes to be obscured. The relaxation of constraints on CFDs gives more correlated attributes and thwarts information-confidentiality breaches. It is seen from that the number of the attribute to be obscured is more when relaxation on both constraints on CFDs are set. The attributes *edu* and *age* are determine other correlated attributes in the dataset, therefore number of attributes to be obscured is more for attribute *edu* and *age*. The attribute *edu*, *age* and *fmember* are more correlated attributes and can be used to determine total income of family.

The number of attributes to be obscured are different under different configuration settings in CFDs for *Heart-Disease* dataset is shown in Figure 5. A similar observation in Figure 5, the number of the attribute to be obscured is more when relaxation on both constraints on CFDs are set. The attribute *vessels*, *age*, *cholesterol*, *bp* and *slope* are determine other correlated attributes in the heart-disease dataset. Figure 4 and Figure 5 point out the relation between the number of attributes to be obscured and the level of information confidentiality. It indicated obscuring more attributes to thwart information-confidentiality breaches and increase information confidentiality.

In Figure 6, the x-label indicates the information-confidentiality attributes, and the y-label indicates the number of attributes to be obscured to ensure information-confidentiality on attributes. The heuristic MVFS is used for finding a minimal set *Z* to be encrypted. It is obvious that the number of attributes to be obscured is more as the number of information-confidential attributes increases. It is observed from figures 6 and 7 that when the confidential attribute is added, the proposed methodology discovers additional CFDs and applies heuristic MVFS to find a new minimal set *Z* to be obscured. Thus, for every introduction of a new confidential attribute, discovering a new set of CFDs and selecting minimal set Z to be obscured will occur. It is noticed from figure 6 that a linear increase in attributes to be obscured for a given set of information-confidentiality attributes for all configuration settings on CFDs except on *Rel-Attr-Sim-Cov-Range* configurations. For *Rel-Attr-Sim-Cov-Range* configuration, the number of attributes to be obscured decreases for a given set of information-confidentiality attributes.

Figure 7 represents the results of the experiment obtained on the Income-Census dataset. It is noticed from figure 7 linear increase in attributes to be obscured for a given set of information-confidentiality attributes for all configuration settings on CFDs. For configuration *Rel-Cov-Range*, it is a linear increase at the beginning and then remains unchanged for the given information-confidentiality attribute from 2 to 4. Similarly, for *Rel-Attr-Sim* configuration a linear increase

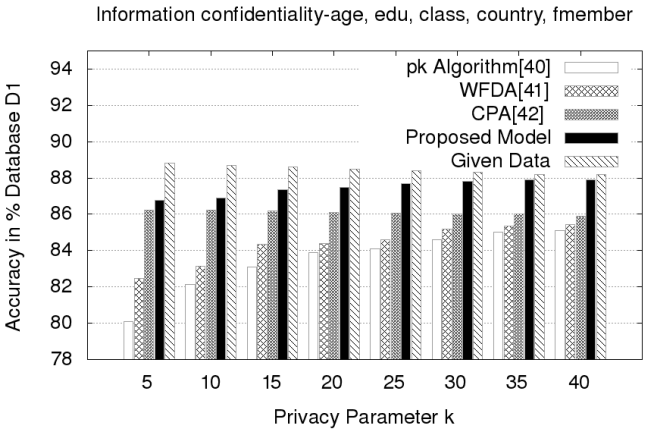


Fig. 8: Accuracy of Proposed Model for Dataset D1

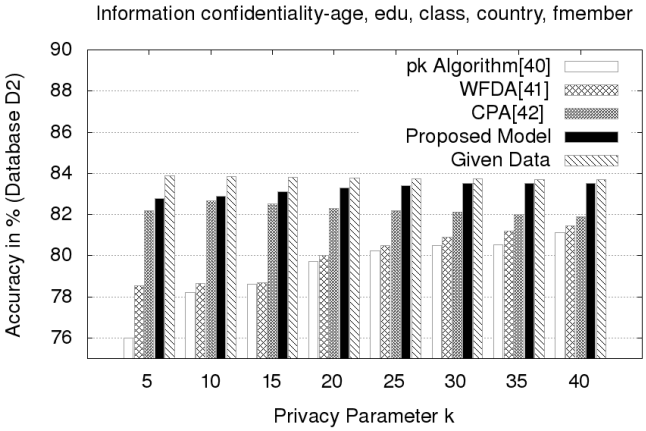


Fig. 9: Accuracy of Proposed Model for Dataset D2

for given information-confidentiality attribute from 1 to 3 and remains unchanged for given information-confidentiality attribute.

Fig. 8 and Fig. 9 show values of accuracy for different values of k and for three different dataset D_1, D_2 . The accuracy is used to compare the data quality before and after data anonymization. If data distortion is more after performing anonymization to the dataset, then accuracy values is less. Otherwise, the accuracy is higher. It is noticed in Fig. 8 and Fig. 9 that the accuracy of the proposed privacy model

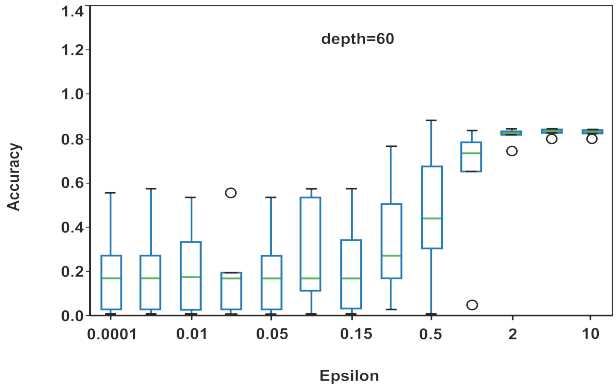


Fig. 10: Quantification of Privacy at depth 60

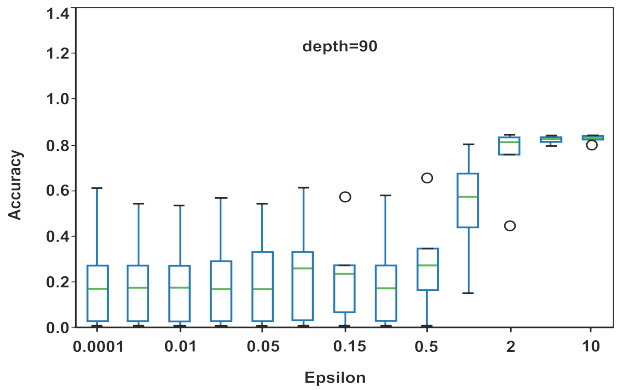


Fig. 11: Quantification of Privacy at depth 90

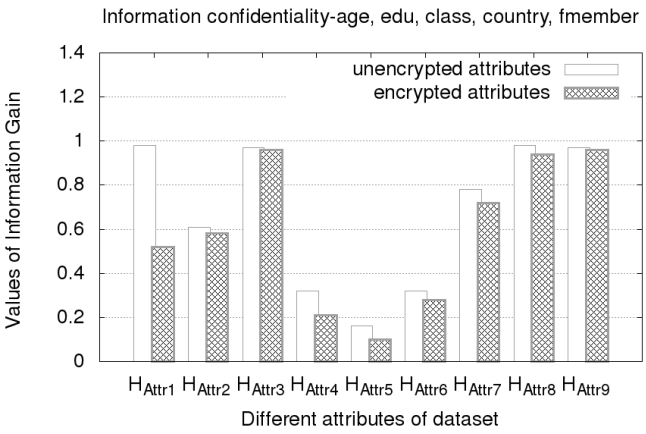


Fig. 12: Information Gain values on *Heart Disease* Dataset

is better than existing privacy models. The rationale beyond better accuracy is the consistency in generalizing the dataset. The data quality after anonymization is marginally lower than the original dataset. The proposed privacy achieves better accuracy because it uses similarity and ensures an approximately equal distribution of sensitive attributes in each equivalence class. The proposed privacy model calculated assailability weights of the quasi-identifier attributes and quasi-identifier attributes value domain consistency during the data anonymization process

The variables d and ϵ indicate the depth of the taxonomy tree and privacy breach at every level of the tree. The effect of depth of taxonomy tree and privacy breach on the accuracy of privacy preservation model is shown in Fig. 10 and Fig. 11. It is observed from Fig. 10 and Fig. 11 that the accuracy increases dramatically when ϵ the value ranges from 0.5 to 2. The value ϵ quantifies the privacy beaches.

The performance of the proposed algorithm is measured according to the information gain. The information gain metric is used to measure the data quality. It is vital to find the data quality in terms of information gain on obscured and non-obscured datasets. The information gain gives data dispersion. The information gain use concepts of entropy. The entropy is defined as

Entropy = $H(X)$=-Σ $p(X)log$ $p(X)$, X represent attribute. H(X) represent entropy of attribute X. p(X) probability of occurrence of attributes X. The information gain downgrades entropy values because it partition the dataset according to the attributes. The formal definition of information gain is formally defined as

Information Gain = $I(X,Y)$ = $H(X)$-$H(X/Y)$, I(X,Y) represent the information gain on X, Y attribute, H(X) represent entropy of attribute X given attribute Y. The information gain for different attributes of datasets has been evaluated. For *Heart-Disease* dataset, *Occupation* attribute is used as the target attribute. Thus, for each attribute, information gain is evaluated. All encrypted values are considered in the computation of information gain. Figure 12 illustrates the results of the experiments. The red bar represents the information gain on non-encrypted attributes, and the blue bar represents information gain obtained after applying the proposed algorithm (i.e., information gain on encrypted attributes). The difference in information gain on non-encrypted and encrypted attributes is small. The small variation of information gain indicates guaranteed privacy preservation without the annoying quality of data. Maximum variation in information gain occurs when an attribute has many values and encryption on these values. Minimum variation in information gain occurs when an attribute has a unique value. In figure 12, attribute H_{Attr1} has maximum variation in information gain, and H_{Attr3} has minimum variation in information gain. The variation in information gain indicates how partial encrypted data attributes help to ensure both privacy and data utility.

The proposed model's execution time depends on the dataset size, application and active nodes. The set of experiments is conducted by fine-tuning parameters, such as the number of executors, memory, number of input splits, number of shuffles, and cores of executors. The proposed model's execution time on the MapReduce framework is based on the number of splits in the datasets. For analysis of the dataset size (i.e., input split) on execution time, initially, the size of the dataset (input split) is set to 128MB, and other default parameters are changed. Later, three different dataset input splits (256MB, 512MB, 1024MB) are selected, and the default parameters are unchanged. It is noticed from Fig. 13a that the execution time increased by 2when a dataset size of 256MB and a 2% increase in execution time continues for a dataset of size 300GB. Moreover, it is observed from Fig. 13a that execution times are similar when data sizes are less than 200 GB.

Fig. 13b demonstrates the execution time on the MapReduce framework with fine-tuning shuffle parameter. It is noticed that the average execution time increases linearly for datasets of sizes up to 450 GB. When the shuffle parameter is fine-tined (i.e. Reduce=150 and task=45), the proposed model's execution time is enhanced by 1%. In general, the proposed model with the default parameter has optimum performance. The execution time of the proposed model on the Apache spark framework is illustrated in Fig. 13c. On the Apache spark framework, we have fixed parameters such as the number of executors and their memory but changed the dataset size to measure the execution time of the proposed model. The default block size of the dataset on Apache spark is 128 MB. However, the block sizes are varied (i.e., 256 MB, 512 MB) to measure the execution time of the proposed model. The results illustrate that block sizes of 512 MB and 1024 MB have better execution times. It is observed that

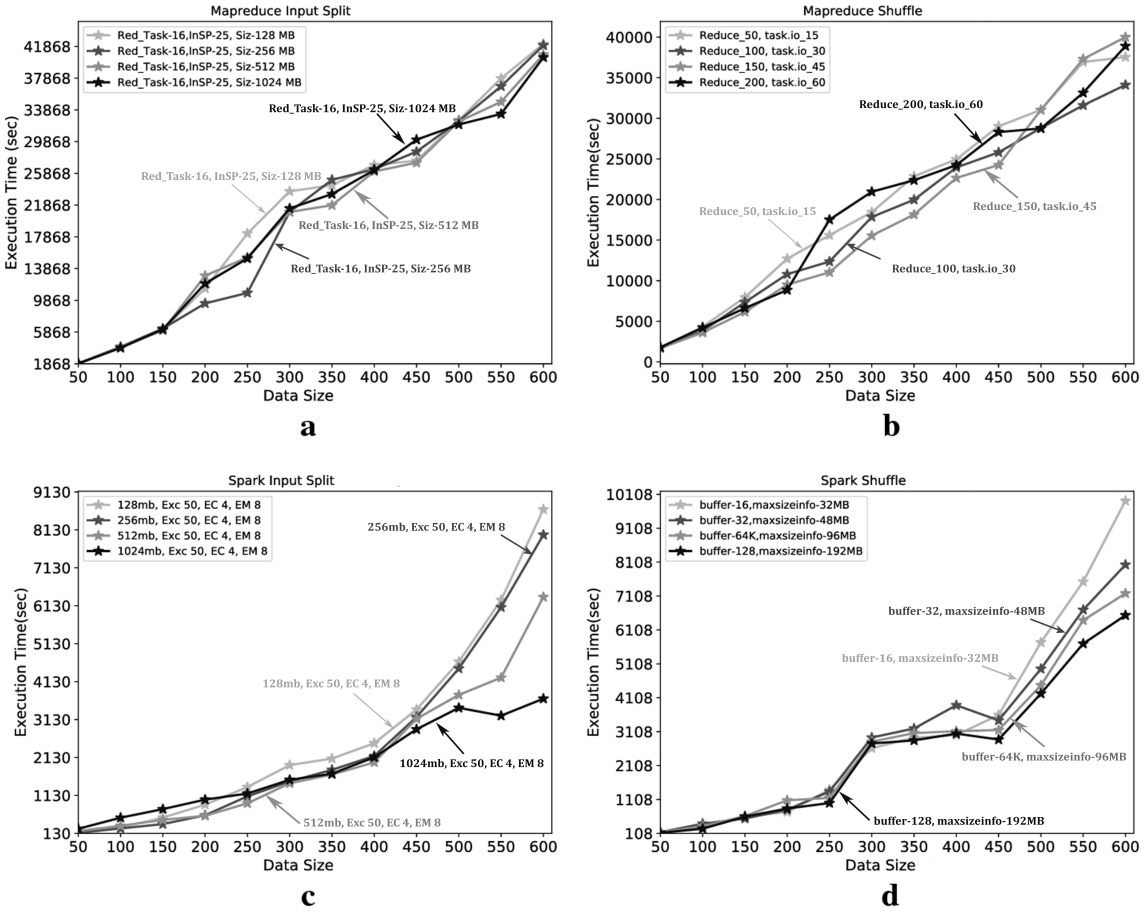


Fig. 13: Execution time of proposed model on a) MapReduce Framework for input split dataset D1, D2 b) MapReduce Framework for shuffle dataset D1, D2, c) Apache Spark framework for input split dataset D1, D2 d) Apache Spark framework for shuffle dataset D1, D2

execution time is improved by 4% when a block size of 1024, and execution time for a data size of 500 GB. Thus, the execution time on the spark framework is improved for a larger dataset size. It is concluded that the performance of the proposed model is scalable for a large dataset on the Apache spark framework.

Fig. 13d shows the performance of the proposed model for shuffle parameters setting. In an experimental setup, initially, the default parameters such as *buffer* and *spark.reducer.maxSizeInFligh* is set at 32 and 48MB respectively. During the experiment, parameter *buffer* and *maxSizeInFlight* are set at 128 and 192 respectively, and the execution time for the proposed model proportionally increases. An increase in the execution time is the more number of input splits for distinct executors on the apache spark.

An extensive set of experiments are conduct to evaluate the impact of l, d parameters on information loss incurred after anonymization. This research work uses a real and synthetic dataset. The CENSUS is the real dataset that contains information about adults, *U-dis* is a uniformly distributed synthetic dataset and *S-dis* is a skewed distributed synthetic dataset. The parameter l indicate set of distinct sensitive values in quasi-identifier group, the parameter d indicate close frequency of sensitive values. For example, two sensitive values S_1, S_2 are said to be d-close if $| f_1 - f_2 | \leq d$. Since d has direct impact on anonymization. Thus,

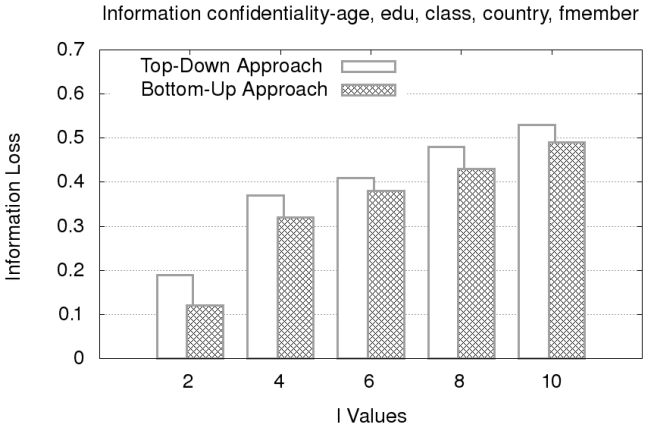


Fig. 14: Total Information loss by Top-Down and Bottom-Up approach

frequency distance is calculated as $d = \sum_{1=1}^{n} | f_i - f_{i-1}|/n$, where n indicate distinct sensitive values, f_i and f_{i-1} are frequency of neighboring sensitive values. Fig. 14 shows comparison results of information loss incurred by top-down and bottom-up anonymization approaches on the CENSUS dataset. It is observed that information loss incurred by the bottom-up anonymization approach is much better than the top-down anonymization approach. It is also observed

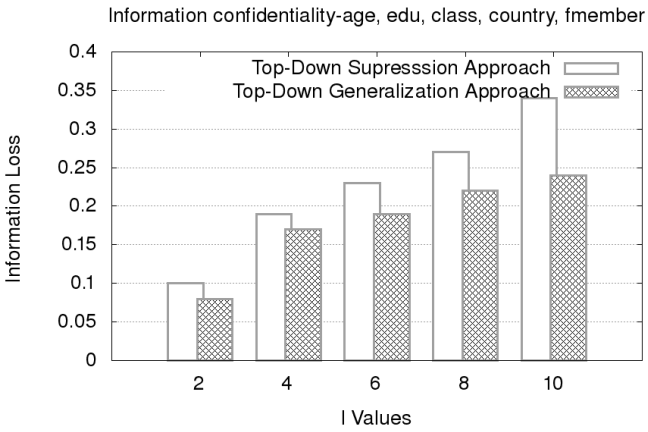


Fig. 15: Total Information loss by Top-Down Suppression, Top-down Generalization on uniformly distributed dataset(U-dis) with 500k tuples, d=10

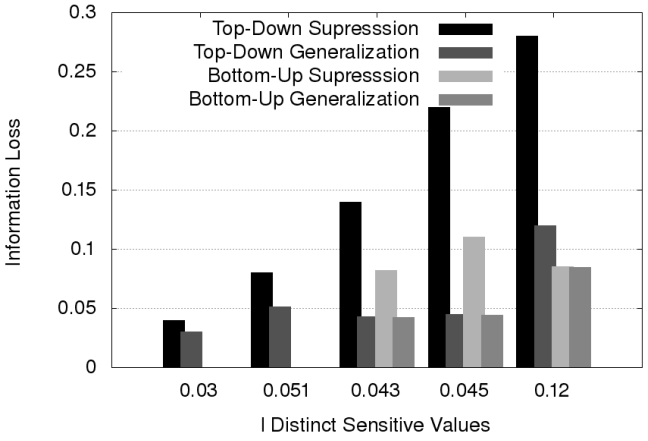


Fig. 16: Total Information loss by Top-Down Suppression, Top-down Generalization, Bottom-Up Suppression, Bottom-Up Generalization for skewed dataset(S-dis)with 500k tuples, d = 150.

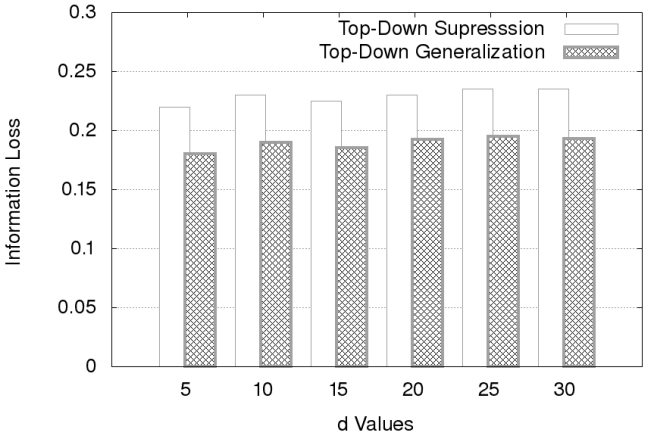


Fig. 17: Total Information loss by Top-Down Suppression, Top-down Generalization for uniformly distributed dataset(U-dis)with 500k tuples, l = 5.

that information loss is more for larger values of l. It is because of two reasons: first, splitting of distinct sensitive values is more for a larger value of l, which causes more tuples to be eliminated to satisfy frequency of closeness d; Second, a larger value of l yields more data distortion. Moreover, anonymization through generalization leads to more information loss.

Fig. 15 illustrate impact of l (i.e. set of distinct sensitive attributes) on information loss on *U-dis* dataset. it is noticed from the figure that more information loss by top-down tuple suppression with increasing l values. It happens for two reasons: first, distant and more sensitive values are partitioned. Consequently, more tuples are removed from the dataset to comply d-closeness. Second, the procedure is repeated on all partitions until no further distinct set of sensitive values partition. If the partition size is less than $2l$, then no further split. Fig. 16 shows the impact of l (i.e. set of distinct sensitive attributes) on information loss on *s-dis* dataset. The information loss on *U-dis* dataset is slightly stable than s-dis dataset, since the *u-dis* dataset has set of sensitive values with similar frequencies. Fig. 16 shows that larger value of *d=150* yields less information loss by tuple generalization. It happens because they are not required to remove more tuples to compliance d-closeness. Second, CFD introduces less information loss.

Fig. 17 demonstrate the impact of d on the performance of Top-Down suppression, Top-down generalization anonymization. The performance of Top-down generalization anonymization is relatively stable with increasing d value since sensitive values with similar frequencies are grouped.

VII. Conclusions

The proposed methodology exploits the correlation between attributes using Constraints-Relaxed functional dependencies (CFDs) in the dataset to thwart information confidentiality in big data applications. It finds a minimal set of sensitive attributes to be obscure using heuristic MVFS from a set of CFDs and then encrypts a minimal set of sensitive attributes using a block cipher. This paper aims to find the minimal set of sensitive attributes to be obscured while preserving information confidentiality in big data applications. The minimal set of attributes is encrypted to enhance sensitive data confidentiality and use non-encrypted attribute values for knowledge discovery and analytics purposes. The paper examined different threats and analyzed the robustness of the proposed methodology.

The experimental results show the number of attributes to be obscured under different configuration settings in CFDs for *Heart-Disease*, *Income-Census* dataset. The results of the experiments show a correlation between attributes in the dataset. The experimental results established a relation between the number of attributes to be obscured and the level of information confidentiality. From experimental results, it is observed that the relaxation of constraints on CFDs directly impacts the number of attributes to be obscured.

Acknowledgment

The authors would like to express their gratitude to the professors of the department for sharing their useful input with them during this research.

REFERENCES

[1] Mohanta, Giridhari, Sathya Swaroop Debasish, and Sudipta Kishore Nanda, 'A Study on Growth and Prospect of Digital India Campaign', *Saudi Journal of Business and Management Studies*, vol. 2, no. 7, pp. 727-731, 2017.

[2] Ambika.S, Agalya.P, Sneha K.K, Emiliya V.R, 'A Study on Digital India-Impacts', *International Journal of Advanced Science and Technology,* vol. 29, no. 02, pp. 2067-2073, 2020.

[3] L. Sharma and V. Singh, 'India Towards Digital Revolution (Security and Sustainability)', *'2018 Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)',* pp. 297-302, doi: 10.1109/WorldS4.2018.8611564.

[4] S. S. Alrumiah and M. Hadwan, 'Implementing Big Data Analytics in E-Commerce: Vendor and Customer View', *IEEE Access*, vol. 9, pp. 37281-37286, 2021, doi: 10.1109/ACCESS.2021.3063615.

[5] M. Li, T. Li, D. Quan and W. Li, 'Economic System Simulation with Big Data Analytics Approach', *IEEE Access*, vol. 8, pp. 35572-35582, 2020, doi: 10.1109/ACCESS.2020.2969053.

[6] Bhaskar, R and Shylaja, BS, 'Dynamic Virtual Machine Provisioning in Cloud Computing Using Knowledge-Based Reduction Method', *Next Generation Information Processing System*, pp. 193–202, Springer, 2021. .

[7] Mikalef, I. O. Pappas, J. Krogstie, and M. Giannakos, 'Big Data Analytics Capabilities: A systematic Literature Review and Research Agenda', *Information Systems and e-Business Management Volume,* vol. 16, no. 3, pp. 547-578, Aug. 2018.

[8] Ma, Junkuo Cao Mingcai Lin Xiaojin 'A Survey of Big Data for IoT in Cloud Computing' *IAENG International Journal of Computer Science*, vol. 47, no. 3, pp. 585-592, 2020.

[9] Rawas, Soha, and Ahmed Zekri 'EEBA: Energy-Efficient and Bandwidth-Aware Workload Allocation Method for Data-intensive Applications in Cloud Data Centers', *IAENG International Journal of Computer Science*, vol. 48, no. 3, pp. 703-715, 2021.

[10] S. G. Teo, J. Cao and V. C. S. Lee, 'DAG:A General Model for Privacy-Preserving Data Mining', *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 1, pp. 40-53, 1 Jan. 2020, doi: 10.1109/TKDE.2018.2880743.

[11] L. Xu, C. Jiang, J. Wang, J. Yuan and Y. Ren, 'Information Security in Big Data: Privacy and Data Mining', *IEEE Access*, vol. 2, pp. 1149-1176, 2014, doi: 10.1109/ACCESS.2014.2362522.

[12] Belhadaoui, Hicham, and Reda Filali, 'A Mathematical Model to Calculate Data Sensitivity in Hadoop Platform Using the Analytic Hierarchy Process Method' *IAENG International Journal of Computer Science*, vol. 47. no. 4, pp. 765-774, 2020.

[13] R. Mendes and J. P. Vilela, 'Privacy-Preserving Data Mining: Methods, Metrics, and Applications', *IEEE Access*, vol. 5, pp. 10562-10582, 2017, doi: 10.1109/ACCESS.2017.2706947.

[14] A. Mehmood, I. Natgunanathan, Y. Xiang, G. Hua and S. Guo, , 'Protection of Big Data Privacy', *IEEE Access*, vol. 4, pp. 1821-1834, 2016, doi: 10.1109/ACCESS.2016.2558446.

[15] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, 'Privacy-preserving Data Publishing: A Survey of Recent Developments', *ACM Computer Surveys*, vol. 42, no. 4, Jun. 2010, Art. ID 14.

[16] J. Zhou, Z. Cao, X. Dong, and X. Lin , 'PPDM: Privacy-preserving Protocol for Dynamic Medical Text Mining and Image Feature Extraction from Secure Data Aggregation in Cloud-assisted e-Healthcare Systems', *IEEE Journal on Selected Topics Signal Process,* vol. 9, no. 7, pp. 13321344, Oct. 2015.

[17] H. Liu, X. Yao, T. Yang and H. Ning, 'Cooperative Privacy Preservation for Wearable Devices in Hybrid Computing-Based Smart Health', *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1352-1362, April 2019, doi: 10.1109/JIOT.2018.2843561.

[18] Z. Hui, H. Li, M. Zhang, and D. G. Feng, 'Risk-adaptive Access Control Model for Big Data in Healthcare', *Journal on Communication*, vol. 36, no. 12, pp. 190-199, Jul. 2017.

[19] S. Qiu, B. Wang, M. Li, J. Liu and Y. Shi, 'Toward Practical Privacy-Preserving Frequent Itemset Mining on Encrypted Cloud Data', *IEEE Transactions on Cloud Computing*, vol. 8, no. 1, pp. 312-323, 1 Jan.-March 2020, doi: 10.1109/TCC.2017.2739146.

[20] V. S. Verykios, A. K. Elmagarmid, E. Bertino, Y. Saygin and E. Dasseni 'Association rule hiding', *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 4, pp. 434-447, April 2004, doi: 10.1109/TKDE.2004.1269668.

[21] K. Nomura, Y. Shiraishi, M. Mohri and M. Morii, 'Secure Association Rule Mining on Vertically Partitioned Data Using Private-Set Intersection', *IEEE Access,* vol. 8, pp. 144458-144467, 2020, doi: 10.1109/ACCESS.2020.3014330.

[22] B. Wang, Y. Zhan, and Z. Zhang, 'Cryptanalysis of a Symmetric Fully Homomorphic Encryption Scheme', *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 6, pp. 1460-1467, Jun. 2018.

[23] L. Sweeney 'k-anonymity: A Model for Protecting Privacy', *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557-570, 2002.

[24] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, 'l-diversity: Privacy beyond k-anonymity', *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol, 1, no. 1, pp. 1-24, 2007.

[25] N. Li, T. Li, and S. Venkatasubramanian, 't-closeness: Privacy beyond k-Anonymity and l-Diversity', *in Proceedings of IEEE 23rd International Conference on Data Engineering*, pp. 106-115, 2007.

[26] R. Schnell, T. Bachteler, and J. Reiher, 'Privacy-preserving Record Linkage using Bloom Filters', *BMC Medical Informatics and Decision Making*, vol. 9, no. 1, pp. 41-50, Dec. 2009, doi: 10.1186/1472-6947-9-41.

[27] N. Li, W. Qardaji, D. Su, and J. Cao, 'PrivBasis: Frequent Itemset Mining with Differential Privacy', *Proceeding of VLDB Endowment*, vol. 5, no. 11, pp. 1340-1351, Jul. 2012,

[28] C. Zeng, J. F. Naughton, and J.-Y. Cai, 'On Differentially Private Frequent Itemset Mining', *Proceedings of VLDB Endowment*, vol. 6, no. 1, pp. 25-36, Nov. 2012.

[29] Y. -T. Tsou, H. Zhen, X. Jiang, Y. Huang and S. -Y. Kuo, 'DPARM: Differentially Private Association Rules Mining,' *IEEE Access*, vol. 8, pp. 142131-142147, 2020, doi: 10.1109/ACCESS.2020.3013157.

[30] A. M. Khedr, Z. A. Aghbari, A. A. Ali and M. Eljamil, 'An Efficient Association Rule Mining From Distributed Medical Databases for Predicting Heart Diseases,' *IEEE Access*, vol. 9, pp. 15320-15333, 2021, doi: 10.1109/ACCESS.2021.3052799.

[31] ElDahshan, K., E. K. Elsayed, and H. Mancy, 'Enhancement Semantic Prediction Big Data Method for COVID-19: Onto-NoSQL', *IAENG International Journal of Computer Science*, vol. 47, no. 4, pp. pp. 613-622, 2020.

[32] Y. Sun, Q. Liu, X. Chen and X. Du, 'An Adaptive Authenticated Data Structure With Privacy-Preserving for Big Data Stream in Cloud,' *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3295-3310, 2020, doi: 10.1109/TIFS.2020.2986879.

[33] S. Sharma, J. Powers and K. Chen, 'PrivateGraph: Privacy-Preserving Spectral Analysis of Encrypted Graphs in the Cloud,' *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 5, pp. 981-995, 1 May 2019, doi: 10.1109/TKDE.2018.2847662.

[34] A. AlMahmoud, E. Damiani, H. Otrok and Y. Al-Hammadi, 'Spamdoop: A Privacy-Preserving Big Data Platform for Collaborative Spam Detection,' *IEEE Transactions on Big Data*, vol. 5, no. 3, pp. 293-304, 1 Sept. 2019, doi: 10.1109/TBDATA.2017.2716409.

[35] Razali, Noor Afiza Mat 'Secure Blockchain-Based Data-Sharing Model and Adoption among Intelligence Communities', *IAENG International Journal of Computer Science*, vol. 48, no. 1, pp. 18-31, 2021.

[36] Caruccio, Loredana and Deufemia, Vincenzo and Polese, Giuseppe, 'Mining Relaxed Functional Dependencies from Data', *Data Mining and Knowledge Discovery*, vol. 34, no. 2, pp.443-477, 2020.

[37] Schwikowski, Benno and Speckenmeyer, Ewald, 'On Enumerating All Minimal Solutions of Feedback Problems' *Discrete Applied Mathematics*, vol. 117, no. 1-3, pp. 253-265, 2015.

[38] William Stallings, 'The Offset Codebook (OCB) Block Cipher Mode of Operation for Authenticated Encryption, ' *Cryptologia*, vol.42, no. 2, pp. 135-145, 2018.

[39] C. L. Blake and C. J. Merz, 'UCI Repository of Machine Learning Databases,' *Accessed*: May. 11, 2021. [Online]. Available: http://archive.ics.uci.edu/ml/index.php

[40] Liu, Peng, Yan Bai, Lie Wang, and Xianxian Li, 'Partial k-anonymity for Privacy-preserving Social Network Data Publishing' *International Journal of Software Engineering and Knowledge Engineering*, vol. 27, no. 01, pp. 71-90, 2017.

[41] Eyupoglu, C., Aydin, M. A., Zaim, A. H., Sertbas, A '"An Efficient Big Data Anonymization Algorithm based on Chaos and Perturbation Techniques', *Entropy*, vol. 20, no. 5, pp.373-383, 2018.

[42] Han, Jianmin, Juan Yu, Jianfeng Lu, Hao Peng, and Jiandang Wu 'An Anonymization Method to Improve Data Utility for Classification', *Proceedings of International Symposium on Cyberspace Safety and Security*, pp. 57-71. Springer, Cham, 2017.

Satish B Basapur is currently working as an Assistant Professor in the Department of Information Science and Engineering, Dr. Ambedkar Institute of Technology, Bengaluru-560056. He completed a Bachelor of Engineering degree in Computer Science and Engineering from Kuvempu University in

1999 and a Master of Technology degree in Computer Network Engineering from Visvesvaraya Technological University in 2009. His research interest includes Cloud Computing, Big Data analytics, and Data Mining.

B S Shylaja is currently working as a Professor in the Department of Information Science and Engineering, Dr. Ambedkar Institute of Technology, Bengaluru-560056. She has Bachelor of Engineering degree in Computer Science and Engineering from Mysore University in 1987 and Masters of Science degree in Computer Science and Engineering from Birla Institute of Technology and Science Pilani, Rajasthan, India in 1994. She obtained her Ph.D. degree in Information and Communication Engineering in the Area of Vehicular Ad hoc Networks from Anna University Chennai in 2011. Her research interest includes Wireless Sensor Networks, Cloud Computing, and Big Data Analytics.

Venkatesh (M'18) is currently working as an Associate Professor in the Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bangalore-560001. He has obtained his Bachelor of Engineering, Masters of Technology, and Ph.D. in Computer Science and Engineering in 2000, 2004, and 2018 respectively. His research interests include Wireless Sensor Networks, Ad-hoc networks, Data Science, Web Recommendation Systems, and Data Privacy in Big Data.