


Received December 29, 2020, accepted January 27, 2021, date of publication March 2, 2021, date of current version March 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3063603

On Differential Privacy-Based Framework for Enhancing User Data Privacy in Mobile Edge Computing Environment

JHILAKSHI SHARMA¹, DONGHYUN KIM² , (Senior Member, IEEE),
AHYOUNG LEE¹, (Member, IEEE), AND DAEHEE SEO³, (Member, IEEE)

¹Department of Computer Science, Kennesaw State University, Marietta, GA 30060, USA

²Department of Computer Science, Georgia State University, Atlanta, GA 30303, USA

³Faculty of Artificial Intelligence and Data Engineering, Sangmyung University, Seoul 03016, South Korea

Corresponding author: Daehee Seo (daehseo@smu.ac.kr)


ABSTRACT The potential growth in data mining has an important aspect on security due to the consideration of the data as an asset. The provisioning of protection in a public infrastructure fails to ensure privacy disclosure of an individual's information. Differential Privacy (DP) is a promising solution for assuring privacy protection by injecting noise using the Laplace mechanism or Exponential mechanism. The access of data by analysts is performed via edge devices. A common problem identified from previous research work is the leakage of privacy at the edge layer and data accessed by unauthorized people. To address the problem, this paper proposes DP-FCNN, that implements Differential Privacy using a Fuzzy Convolution Neural Network (FCNN) with Laplace Mechanism for injecting noise. The processes handled here are data processing and query processing. The dataset is uploaded by the data owner to the data provider, who is responsible for injecting noise and then encrypting with Piccolo encryption before uploading it into the cloud. Based on the uploaded dataset, the data owner constructs a hash index from the extracted key attributes by using the BLAKE2s algorithm for performing hashing. The hash index is fed into the edge server to form a Merkle hash tree due to the data leakage at the edge is eliminated. On the other hand, requests/queries by the data analyst are authenticated by the data provider. The hash tree in the edge server then searches for the corresponding data, extracting it from the cloud and delivers it to the data analyst in an encrypted format. Every authenticated data analyst is provided with a decryption key for retrieving the query result. This is implemented using Java and the results show better efficiency in terms of scalability, processing time and accuracy.

INDEX TERMS Differential privacy, fuzzy, convolution neural network, Merkle hash tree.

I. INTRODUCTION

In the modern industrial era, the need for preserving an individual's privacy has been brought to attention by the numerous data breaches that have been plaguing even the most successful organisations and companies. The traditional ways of protecting the confidentiality of data, such as cryptography, would destroy the utility of the data as it would prevent any algorithm from accessing the data at all. Differential Privacy (DP) is a guaranteed standard solution that offers privacy for a dataset by holding on to the individual's personal information. In recent days, data is collected

from different environments. It also includes the Internet of Things (IoT). Hence, the data can be related to healthcare, smart home, vehicle communication, smart grid and other applications [1]–[4]. The need for privacy exists in every real-time application. In these applications, the private data collected from individuals suffers from differential attacks. DP is generally categorized into two types as centralized DP (CDP), local DP (LDP) [5], [6]. In CDP, the gathered data is stored into a trusted entity which is responsible to perform DP in order to deliver the processed data to analysts. In DP, a privacy parameter ϵ plays a vital role in determining the security of the data. The properties in DP are guaranteed to provide full assurance for safeguarding the personal information. The major reasons for using DP are illustrated below:

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

- 1) It provides data privacy with simpler computations that enable support with the construction of blocks of privacy.
- 2) In post-processing by analysts the retrieved result cannot be extracted without knowing the information regarding the database in which it is uploaded.
- 3) The tradeoff that exists between data privacy and accuracy is effectively managed.

In DP, privacy for a dataset is obtained by adding noise into the uploaded data. The adding of larger noise tends to improve privacy while the accuracy is reduced. Hence the adding up of noise is required to be a correct ratio and so the balance between accuracy and privacy could be achieved. DP is presented by incorporating clustering and machine learning algorithms as Support Vector Machine (SVM) and deep learning [7]–[10]. Laplace random noise is the most common mechanism that is applied into a dataset for privacy. The selection of ϵ using different algorithms reflects on privacy of the data. Similarly, the addition or removal of a bit of data from the entire dataset should not impact the query results. Meanwhile, the performance of DP makes it harder for any type of attacker to guess the sensitive attributes that are present in the dataset. The process of DP is essential to address all the security aspects from the query till the response from the database.

The data analyst submits their request to the database for the required data via edge computing. Here edge computing plays a vital role in connecting front-end people with the back-end database. For years, edge computing has been used for providing better transfer rates and response times to the end user. This has been done by distributing the computation load to the edge servers, especially in the case of data centers, such that the edge servers are closer to the end user. However, this ease in computation in edge computing comes at the cost of security and privacy. The edge servers, being closer to the end user, come with a need to preserve the privacy of the end user's data and are more prone to data leakages. At the edge, the devices used can be gateway, routers, fog, switches, access points and others. These edge devices are not enabled with the assurance of security and hence they are untrusted in this system. The edge, being an intermediate entity, is vulnerable to leaking information. In case the communication link between the user and the edge is broken or hacked, then the privacy for personal information is not assured. Due to these reasons, DP is also incorporated into edge computing where the security risk is large [11]–[13]. The existence of common challenges in the provisioning of DP here is,

- The adjustment of parameters for improving the data utility based on the size of the datasets that is present in the database.
- The structure of each dataset is different from one another and hence the use of the same parameters for each dataset reduces security strength.
- Required to respond to (support server response for) queries from multiple analysts at a time.

- Data dimensionality variations for the dataset are not able to provide efficient privacy in case of a non-adaptable privacy parameter.
- Untrusted edge devices leak out the information either with intention or without intention.

In this paper, differential privacy is presented with a combination of artificially intelligent and deep learning methods for efficient addition of noise. This model combines fuzzy logic and convolutional neural networks such that the fuzzy membership functions are fed into the convolutional networks in order to produce noise. The entities at the edge are untrusted and hence they are appointed only for forwarding the request and searching results for the given query. In order to make the searching secure, this work presents a Merkle hash tree using which the edge can search and retrieve corresponding data from the cloud, which has been encrypted using Piccolo encryption, a lightweight encryption method. Additionally, the data analyst is authenticated with credentials due to increased security threats. Therefore, a completely refined security system is designed. The organization of this research paper is further elaborated in the following subsections.

The key contribution of the proposed research work is summarized as follows:

- Two-fold process of uploading the data owner's dataset and serving analysts queries with the improvement of accuracy as well as privacy protection.
- The ϵ parameter is effectively added from a combination of Fuzzy Convolution Neural Network (FCNN) using the Laplace mechanism. This is performed by analyzing the sensitivity and attributes of the dataset, so that it is applicable to datasets of different dimensions.
- Public cloud is untrusted, so the data is encrypted using the lightweight Piccolo algorithm and then it is uploaded into the cloud.
- Untrusted edge devices are equipped to search analyst queries in a Merkle hash tree that ensures the edge devices have no knowledge about the data. The search results are extracted from the cloud in an encrypted format and delivered.
- The analyst is authenticated using individual credentials before they are provided with a decryption key. Here hashing is performed with the lightweight BLAKE2s algorithm.
- The incorporation of lightweight algorithms ensures that lesser resources are consumed and that they also perform faster. The use of the public cloud and an untrusted edge device is effectively presented by holding the data in an encrypted format and searching in the hash values.

The rest of this paper is organized as follows. Section II gives the initial knowledge of Differential Privacy for understanding the idea of our paper, Section III details the previous research work that has been done in DP for provisioning privacy protection, Section IV highlights the common problems that exists in DP, Section V describes the proposed solutions that are defined to solve the identified problems, Section VI illustrates the experimental evaluation of this proposed work

that justifies improvements and Section VII concludes this research along with future directions.

II. PRELIMINARY KNOWLEDGE

DP is a privacy provisioning method that is employed with the property of including noise into the dataset. The added noise ensures the protection for the private information that is present in the uploaded dataset. In this section the common steps that are followed in DP and their definitions are discussed. In this section, Figure 1 illustrates the process where data analysts query into a differently private framework. Here n data analysts request the database for receiving their query response. The mechanisms for adding noise into the dataset are Laplace mechanism and exponential mechanism. The data gathered from different environments is outsourced into the public database. The sharing of data in this way meets the security constraints since outsourcing sensitive data with personal information is not advisable in recent days due to the increased vulnerabilities and threats [14], [15]. Let D and D' represent two neighboring datasets that are not similar since they differ by one dataset entry. In order to secure the private information in the dataset, noise ϵ is added. By adding this noise, it is not able to predict whether the particular entity exists in the database or not.

$$R_r[R(D) \in S_s] \leq \exp(\epsilon) \times R_r[R(D') \in S_s] \quad (1)$$

The term R_r indicates randomness of the R algorithm, S_s is the subset of P_r where P_r represents the possible output sets obtained from R . The ϵ is the privacy budget that defines the level of privacy protection provided to the data. The lower the value of ϵ selected, the stronger the privacy [16]. The protection in DP is provided by two categories as local and global sensitivity. The maximum changing value for the adjacent dataset is expressed as per the following equation,

$$\Delta F = \text{MAX}\{D, D'\} \|F(D) - F(D')\|_1 \quad (2)$$

ΔF is the varying output result that is formulated from F represents the function of global sensitivity and the first order distance between D and D' is given in the function of $F(D)$ and $F(D')$ as $\|F(D) - F(D')\|$.

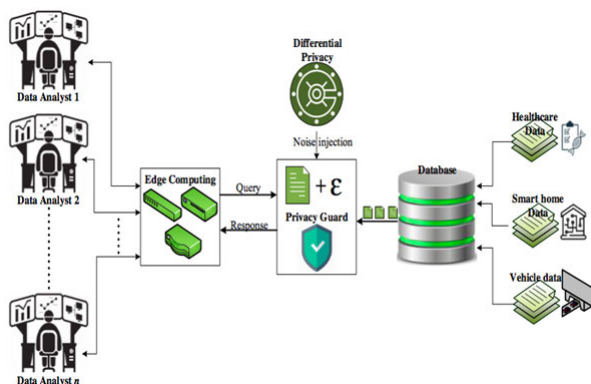


FIGURE 1. Process of differential privacy.

The Laplace Mechanism [17] is performed as per the Laplace distribution, let Q be the query from analyst. This Laplace mechanism satisfies the chosen ϵ parameter and the expression is given as,

$$R(x) = Q(x) + \text{Lap}\left(\frac{\Delta Q}{\epsilon}\right) \quad (3)$$

The Laplace mechanism is defined with a scale factor $\frac{\Delta Q}{\epsilon}$, where the given Q is mapped into the dataset which is present in the form of string, strategy or tree. The exponential mechanism is defined based on the exponential distribution.

III. PROBLEM DESCRIPTION

The major problems identified in the field of DP are highlighted in this section. A personalized differential privacy scheme was implemented to secure a smart home model based on fog computing [18]. The privacy level was estimated from the trust distance, measured from the Markov process. Then, in machine learning algorithms two approaches were developed to ensure privacy based on the estimation of weight using classification and regression tree (CART) method [19]. The noise to be added is computed using the attribute weights by decision tree. Later, if the ϵ value was poor, the deep NN was processed. The problems defined in these works can be defined as follows: the initial ϵ value has to be efficient in order to reduce repeated computation and the use of decision tree results in inaccurate output due to poor estimation of weight, which in turn leads to poor selection of ϵ . Then the parameter used for DP has to be significant. The distance-based privacy level estimation was not optimal, since the distance depends on the device's mobility. Even the devices at a short distance may contain larger sensitive data. In the Markov model, only three privacy levels were defined. The distance between the device and the fog server was hard to confine within three different privacy levels. Hence, the privacy level has to be determined based on the information of the data. The process of authentication and secure data storage was also employed in [20], [21]. The problems that existed while providing authentication and secure data storage were: The data miner was authenticated using her ID which is not a strong credential, since the ID of the user can be retrieved easily. Secondly, the decryption key was provided only after authentication, so anyone with an authenticated ID would act as a data miner and decrypt the received data. Kd-trees based cannot support the increase in the dimensionality of the dataset. Advanced Encryption Standard (AES) based secure data storage increases computations and hence requires more resources for encrypting and decrypting the data. To solve these overwhelming problems, we propose a framework, which ensures differential privacy along with authentication and secure storage as a solution in this research.

IV. PROPOSED FRAMEWORK

The proposed DP-FCNN is presented in this section with a detailed solution for the problems defined in DP. This section discusses three subsections as System Design, Data

Processing and Query Processing. The common entities that are involved in the proposed system are elaborated and the algorithms used are explained in detail.

A. SYSTEM DESIGN

The proposed DP-FCNN system is modeled with the major definitions that are required to build the system. The entities that are involved in this system are data owner, data provider, edge server, data analyst and cloud. There exist multiple data analysts that try to access the data and multiple data owners that upload the data. The definition for each entity used in the system is illustrated below:

- **Data Owner:** The data owner uploads a dataset of any type. The complete dataset is uploaded along with the index for the dataset which is required for searching purpose of data analyst. The data owner is assumed to be trusted in this system.
- **Data Provider:** This entity is the most trusted in the system which is responsible to inject noise, encrypt data, hash the index values and then authenticate the data analyst.
- **Edge Server:** This entity is connected with the data provider and the cloud to provide access for the data analyst query. They are responsible for forwarding the query to the data provider, searching the query from the hash tree and retrieving files from the cloud.
- **Cloud:** As known, the cloud is a public database used in this work, which stores the data incoming from the data provider and delivers it to data analysts via edge server.
- **Data Analyst:** The data analyst is also known as a data miner who submits their query to edge and receives a decryption key and response from edge.

This proposed system is categorized into two processes: one for the data owner and the other for the data analyst. Firstly, the data owner uploads the dataset into the data provider. Then the data provider determines the ϵ value using FCNN with the Laplace mechanism and then encrypts the data using the Piccolo algorithm. The encrypted data is given to public cloud for storage. On the other hand, the index terms received from the data owner are converted into hash using the BLAKE2s algorithm and constructed as a Merkle hash tree in edge server. In this work, the data provider is the only trusted entity that participates in the system for privacy protection and authentication of the data analysts.

Secondly, the data analyst submits a query to the edge server, which is forwarded to the data provider. The data provider verifies the security credentials of the data analyst and then allows access to search query. The query is in the form of a hash and so it can be searched for in the Merkle hash tree, after which the encrypted results are obtained from the cloud. Meanwhile, the authenticated data analyst will receive its decryption key for decrypting the data. The entire process handled in this proposed system is depicted in figure 2. As shown in the figure, the data analyst operates on a set of 4 processes whereas the data owner operates with a set of 5 processes.

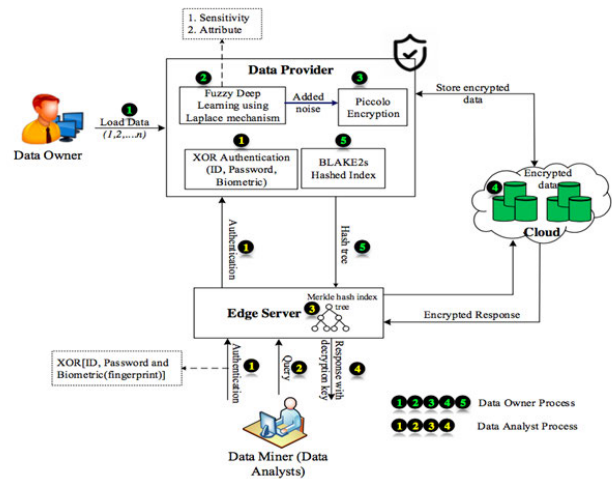


FIGURE 2. Proposed DP-FCNN framework.

B. DATA PROCESSING

The data owner begins to upload the dataset into the data provider, then the provider estimates ϵ based on the dataset using FCNN algorithm. The parameters involved in fuzzy logic are sensitivity and the dataset attributes.

Definition 1 (Sensitivity): This is a significant measure that specifies the amount of noise that is required to be added into the dataset in order to ensure privacy. This is defined based on the variation of the output as a result of the addition of noise into the dataset.

Definition 2 (Attributes): The attributes are the fields that are present in a dataset. The attributes differ based on the type of dataset. For instance, if the dataset is related to health disease the attributes will be of patient name, age, gender and other health issues. These attributes also play a vital role in the dataset.

These two constraints are taken into account for defining the noise ϵ parameter that is to be included into the dataset. This determination is carried out using the FCNN algorithm.

The CNN is composed of five layers as input layer, convolutional layer, pooling layer, fully connected layer and output layer. The fuzzy membership functions are generated from sensitivity and the dataset attributes, further these functions are fed into the convolutional layer and then ϵ is estimated from the Laplace mechanism and added into the dataset. As a result we obtain a dataset with noise added into it. Consider $\rho(P)$ and $V(P)$, the standard deviation and the variance of the probability density function \mathcal{P} . \mathcal{P} is given as,

$$(\mathcal{P}) = \sqrt{V(\mathcal{P})}, V(\mathcal{P}) = 2S^2 \tag{4}$$

where $S = \sqrt{\frac{\Delta G_s}{\epsilon}}$, where G_s denotes the global sensitivity. Therefore the output of the standard deviation and variance is formulated as follows:

$$(\rho\mathcal{P}) = \sqrt{2V\Delta\frac{G_s^2}{\epsilon^2}} \tag{5}$$

$$(\rho\mathcal{P}) = 2\left(\frac{G_s^2}{\epsilon^2}\right)^2 \tag{6}$$

From the estimated ϵ values the noise is added to protect privacy. In FCNN, the input layer extracts the entities and the attributes present in the dataset. By looking at the membership functions in the convolution layer, the importance of each record is predicted. The Laplace mechanism is then applied for determining what is added into the data and then retrieved at the output layer. The resulting probability output from the fuzzy is taken into account for predicting the ϵ using the Laplace mechanism. After prediction, the noise is added and the output will be data with noise added to it.

Here the fuzzy logic system consists of three key components: the fuzzifier, the inference engine and the defuzzifier. The first component is employed for converting the input into a fuzzy set so that it can be processed in the inference engine. So, the input attributes in the dataset are transformed into processable fuzzy sets. Then, the defuzzifier converts back the fuzzy set into crisp values. As per the obtained crisp values, the Laplace mechanism is applied for determining ϵ .

INPUT				OUTPUT
Sensitivity	Attributes			
	Attr ₁	Attr _k	
0	1		0	0
0	0	⋮	1	0
0	0	⋮	0	0
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
1	1	1	1

FIGURE 3. Fuzzy membership functions from fuzzy rules.

The fuzzy rules in FCNN are defined as shown in Table 3, by which the security parameter is defined. Further, using the output value from the convolution layer, the Laplace mechanism is applied using which the specified amount of noise is added into the dataset.

As shown in Figure 4, DP is applied for the uploaded dataset to ensure privacy for personal information. However here the other sensitive attributes are not hidden. Thus a lightweight piccolo algorithm is applied for encrypting the dataset. Piccolo is a block cipher lightweight algorithm that is stronger against differential attacks. Due to this property, the piccolo block cipher is used for encrypting the data before uploading it into the cloud. The input block of 64-bit is processed with a key of 80-bit or 128-bit. As per the used key length, the number of rounds for processing are defined. The steps followed in piccolo encryption are given below:

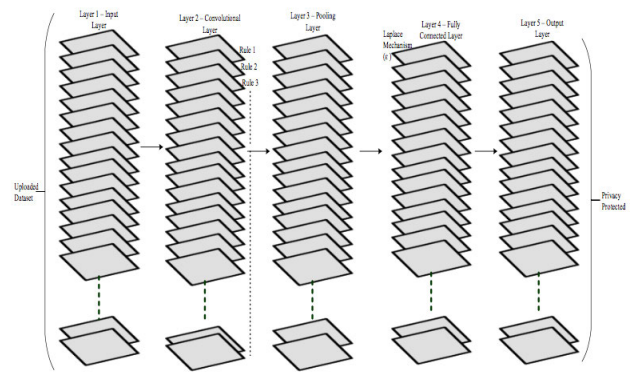


FIGURE 4. FCNN design.

- Step 1: Consider X_{64} as the input block that is given into a lightweight piccolo block cipher.
 - Step 2: From the master keys generate whitening keys and round keys.
 - Step 3: Deploy four Feistel round functions i.e. F-functions having 16-bit level.
 - Step 4: This level consists of a 4-bit S-box that is enabled to process in parallel.
 - Step 5: Then MixColumns is performed that combines the columns of the state by using a particular transformation. This transformation is performed using nibble level.
 - Step 6: On completion of F-function, the whitening sub-key is added and then the plain text is encrypted as Y_{64} .
- The encryption procedure is depicted in Figure 5 using which the input noise added dataset is encrypted. Since the

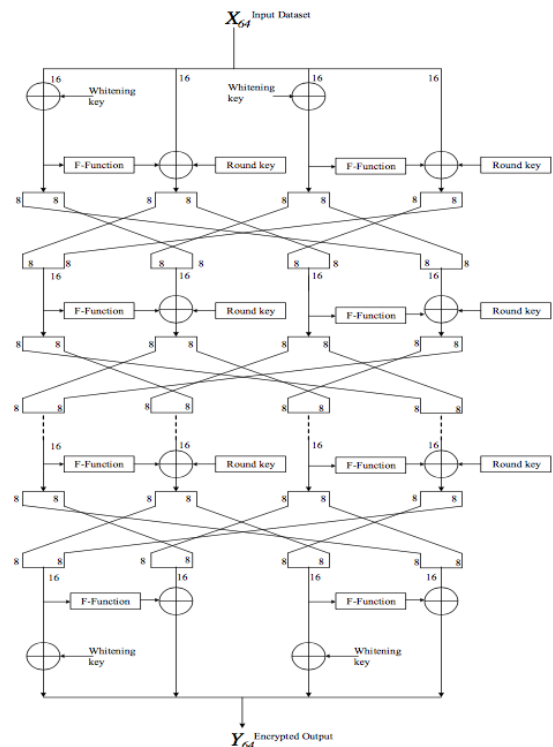


FIGURE 5. Piccolo encryption.

public cloud is not trustable, after encrypting the data, it is stored into the cloud environment. The stored data is also composed of the hashed index terms with which it gives only the required content as response to edge server. On the other hand, the index terms extracted from the dataset is also given to data providers for efficient search results. These index terms are converted into hash values for the purpose of providing security while searching. For hashing the index terms, BLAKE2S is applied which is also a lightweight algorithm. BLAKE2S is suitable to be operated on platforms of 8-bit to 32-bit. A size of 32-bit index term will be converted into 256-bit hash value. Initially, the input index term is divided into 512 blocks. The round function of BLAKE2S is defined as 4×4 matrix that is represented as,

$$M = \begin{pmatrix} v_0 & v_1 & v_2 & v_3 \\ v_4 & v_5 & v_6 & v_7 \\ v_8 & v_9 & v_{10} & v_{11} \\ v_{12} & v_{13} & v_{14} & v_{15} \end{pmatrix} \quad (7)$$

The terms $\{v_1, v_2, \dots, v_{15}\}$ represent the 32-bit index words that are present, M denotes the intermediate state that is initialized. 12 rounds are performed in BLAKE2S. Then the G functions are expressed as

$$\begin{aligned} &G_0(v_0, v_4, v_8, v_{12}), G_1(v_1, v_5, v_9, v_{13}), G_2(v_2, v_6, v_{10}, v_{14}), \\ &G_3(v_3, v_7, v_{11}, v_{15}), G_4(v_0, v_5, v_{10}, v_{15}), G_5(v_1, v_6, v_{11}, v_{12}), \\ &G_6(v_2, v_7, v_8, v_{13}), G_7(v_3, v_4, v_9, v_{14}). \end{aligned} \quad (8)$$

The following set of 8 steps is performed for determining G_i functions: Step 1: $a = a + b + m(\rho_r(2i))$, Step 2: $d = (d \oplus a) \gg 16$, Step 3: $c = c + d$, Step 4: $b = (b \oplus c) \gg 12$, Step 5: $a = a + b + m(\rho_r(2i+1))$, Step 6: $d = (d \oplus a) \gg 8$, Step 7: $c = c + d$, Step 8: $b = (b \oplus c) \gg 7$, where ρ_r indicates the permutation, \oplus denotes the XOR operator for lightweight processing. From these hash values, the privacy is preserved in the edge device, thus resolving the issue of leakage in the edge. The hashed index terms are constructed into a Merkle hash tree in which each node has two children. The key hash is at the root and it generates left and right subtrees. Then each subtree creates further leaf nodes. These hash tree structures are efficient in providing security and also applicable for larger sized datasets. Let a dataset have n index terms that are hashed into h_n values respectively. As discussed earlier, the h_n is generated from the BLAKE2S hashing algorithm.

This tree creates new hash values from which the leaf nodes are linked. Let the root node A have two leaf nodes B and C with the hash values of h_A, h_B and h_C respectively. These hash values are performed with concatenation operation and the root node's hash is $h_A = h(h(n_B)|h(n_C))$. In this way, the Merkle hash tree is constructed. This tree is present in the edge server and the data corresponding to these hash values will exist in the cloud in encrypted format. Hereby the privacy is also achieved in the edge server.

Merkle hash tree structure is depicted in Figure 6 using which the query by the data analyst retrieves the exact result. The construction of the tree is an effective solution for searching. Additionally, privacy is achieved at the edge server by

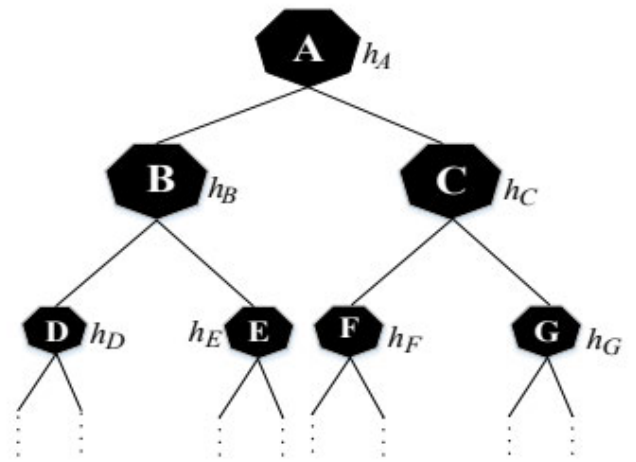


FIGURE 6. Merkle hash tree.

employing a hash based tree structure. Hence the edge server has no knowledge about the data present in the cloud. In data processing, the dataset is uploaded along with the key terms. Then DP is applied into dataset by predicting the values of ϵ as 0.01, 0.1, 0.2, \dots , 0.8, 1. As per the sensitivity and the importance of the attribute, noise is added accordingly. Meanwhile, the key terms are encrypted for efficient secure searching at edge server. Therefore, security is efficiently achieved even when the cloud and edge servers in use are untrusted.

C. QUERY PROCESSING

Query processing is the process of the data analyst submitting his/her query to the available edge server. The initial authentication request from the data analyst consists of her identity and biometric. The biometric used here is the fingerprint, which is unique for every individual and hence is considered as a worthy security credential. Once the authentication request arrives at the edge server, it is forwarded to the data provider. The data provider maintains a list of authenticated data analysts, information with their identity, password and biometric. Based on this stored information, the authentication is performed. In the authentication request, the security credentials identity, password and fingerprint are represented as D_{id} , p_w and F_p respectively. The request R for the data analyst is given as follows,

$$R = [D_{id} \oplus p_w \oplus F_p] \quad (9)$$

If the received $R = 1$, then the data analyst sending the authentication request is legitimate. If $R = 0$, then she is illegitimate. Only legitimate data analysts are allowed to submit a hashed query for searching. The XOR operator is performed between each of the security credentials as the edge server is not trustable. Therefore, the XOR operator makes the credentials of the data analyst secure. Due to the simplicity of the XOR operator, the resources, as well as the time consumed during this operation are limited.

After the authentication, the data analyst submits a hashed query to the edge server. The given hash query traverses

throughout the tree from the root to their left and right nodes. The searching results with matched hash values using which the requested query data is extracted from the cloud. Data extracted from the cloud is in the form of cipher text which is decrypted using a decryption key. This key is also provided by the data provider after the data analyst is authenticated.

D. JUSTIFICATION BEHIND CHOOSING THE COMPONENTS FOR THE PROPOSED FRAMEWORK

1) USING FUZZY LOGIC WITH CONVOLUTIONAL NEURAL NETWORKS (CNNs)

Whenever we use fuzzy logic with neural networks, this is also known as a neural-fuzzy system. A neural-fuzzy system can have a custom design according to the needs of the framework. In our framework, since we are focused on providing differential privacy, which in turn requires a tradeoff between accuracy and privacy and thus an accuracy loss, we have chosen to use a Convolutional Neural Network with fuzzy logic as input. Although CNNs are predominantly used for data with spatial features, i.e. image and video data, we have used it for dealing with tabular data, specifically the adult and heart disease dataset. This is because, even when we are dealing with image data, the idea is to preserve the structure while condensing/downsampling using convolutions by CNN. Since the application of Differential Privacy involves an addition of noise, there is a need for retaining the structure even after the noise has been added. Generally, the architecture of a fuzzy-neural system involves the following:

- Input layer for the input variables
- Fuzzy rules for training the dataset (usually present in the second layer), with fuzzy sets used as the fuzzy connection weights
- The output variables given by the third layer

Since we are using a CNN, we have five layers in the neural network, the architecture of which we have already explained in Section IV-C. The second layer, that is, the convolutional layer, symbolizes the fuzzy rules, as the fuzzy membership functions are fed into this layer. These rules can be seen as examples of the training data. Here, the crisp values of confidential attributes, such as the age and salary attributes in the adult dataset, get fuzzified. This fuzzification leads to data perturbation which in turn preserves privacy of the original data. The reason we have chosen fuzzy logic over other methods such as *k*-anonymity, *l*-diversity, etc. for randomization of data is because it can be used for both numeric (continuous) and discrete attributes [22]. Previous work such as [24] shows how using a Fuzzy Convolutional Neural Network gives more accuracy for the task at hand than a standard CNN. Reference [22] shows more accuracy (99.10%) for handwriting recognition with a fuzzy CNN than with a standard CNN (97.35%) for the training dataset. Inferring from these results, in addition to the other work we mentioned in the literature survey, we chose to apply a Fuzzy CNN for our framework.

2) USING FEISTEL CIPHER BASED PICCOLO ENCRYPTION

Since the aim of the paper was to find a lightweight solution for secure storage, we had to focus on choosing a lightweight encryption algorithm. Reference [23] has proven that Piccolo encryption is ultralightweight, requiring just 60 gates for decryption, making it suitable for extremely constrained environments. Moreover, [23] also shows Piccolo’s effectiveness against different types of differential attacks, such as truncated and higher order attacks. The degree of the lightweightness of Piccolo algorithm can also be measured from the fact it is based on the Feistel Cipher. Feistel structure is used in construction of block ciphers. It is symmetric in nature, with the decryption and encryption process being almost identical, the difference being the reversal in the key schedule. Due to this property, the code required to implement the cipher is almost half the size of other ciphers, proving its lightweightness.

3) USING BLAKE2s HASHING AND MERKLE HASH TREE

In order to ensure fast conversion of the key attributes from the data owner to hashed indices, we have chosen the BLAKE2s hashing algorithm to do so. This is because it provides security as strong as the SHA-3, while also being as fast as the MD5 [24]. The diagram in Figure 7 from [24] shows how fast BLAKE2s in comparison with other algorithms:

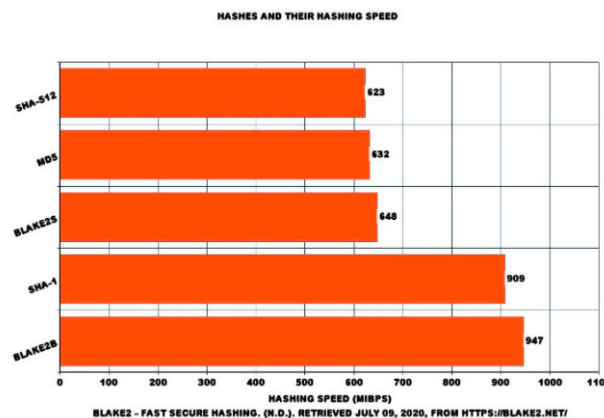


FIGURE 7. BLAKE2s hashing speed comparison.

For storing these BLAKE2s hashes, we have chosen the Merkle Hash tree data structure, over other storage structures such as hash chains and lists. This is because the lookup time for a hash tree is proportional to the logarithm of the number of leaf nodes in the hash tree, while for a hash list it is proportional to the number of leaf nodes in the tree. Therefore, Merkle hash trees provide the fastest access for searching a query.

V. EXPERIMENTAL EVALUATION

In this section, the complete implementation environment is discussed by including the dataset used for processing and comparative results. The graphical plots and their reason for improvements are detailed in this section.

A. IMPLEMENTATION SETUP

This system is designed using Java Development toolkit (JDK) version 1.8 and Weka. The proposed DP-FCNN is modeled by developing data processing and query processing performed with data owner and data analyst. Here the data owner uploads their completed dataset, whereas the data analyst extracts the required data from the database. The WAMP server 2.0 is used to manage the security credentials of the data analysts.

The proposed system is developed in a JDK environment and the other configurations used for the implementation are shown in Table 9. The JDK-1.8 is installed into the Netbeans 8.2 tool that is executed on the Windows 7 operating system. The key goal of this work is security, which is achieved using Differential Privacy, encryption and hashing. The ϵ parameter is determined using the Laplace mechanism in FCNN to add noise and then data is encrypted with the lightweight piccolo algorithm. Then the key terms in the dataset are secured using BLAKE2s hashing. Hereby the specifications used in security algorithms of this proposed work are illustrated in Table 10. The use of machine learning presents higher accuracy with stronger security in the system. For testing purposes, two datasets are used and then the results are evaluated. The proposed implementation structure is illustrated in Figure 8 based on which DP is applied. In this proposed work, all the security aspects are covered while uploading and retrieving the data.

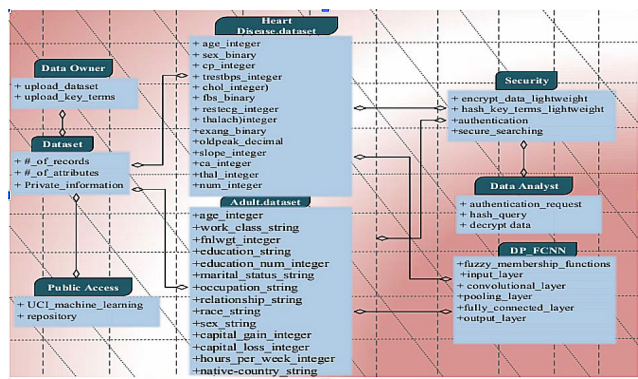


FIGURE 8. Implementation model and dataset.

Parameter	Specification
Operating System	Windows 7
Processor	Pentium Dual Core
System Type	32-bit
Speed	2.50 GHz above
RAM	4 GB
Database	MySQL-5 1.36 (WAMP Server 2.0)

FIGURE 9. Proposed DP-FCNN system implementation configurations.

B. DATASET DESCRIPTION

The proposed DP-FCNN is experimented on two datasets from UCI knowledge discovery Archive database for machine learning systems. This dataset is a public and is

Parameter	Specification
Piccolo Block Cipher	
Key Size	80-bit
Block Size	64-bit
Number of rounds	25
BLAKE2s Hashing	
Key Size	32-bytes
Block Size	64-bytes
Hash Size	32-bytes
Number of Rounds	10

FIGURE 10. Parameters used for encryption and hashing Algorithm.

available for open access [25], [26]. From this database, the Adult and Heart disease dataset are used for this work. The Adult dataset has 48842 records with 14 attributes. Similarly, the Heart Disease dataset contains 3030 records with 75 attributes, among which 14 key attributes are considered. The attributes used are illustrated in Table 11.

Attributes in the Adult Dataset	
Age	Relationship
Sex	Race
Work_class	Capital_Gain
Education	Capital_Loss
Education_num	Hours_per_week
Occupation	Native_Country
Marital_status	Final_weights
Attributes in the Heart Disease Dataset	
Age	Maximum_Heart_Rate
Sex	Exercise
Chest_Pain	ST_Depression
Resting_Blood_Pressure	Peak_Exercise_ST_Segment
Serum_Cholesterol	Number_of_Major_Vessels
Fasting_Blood_Sugar	Defect
Resting_Electrocardiograph	Diagnosis_of_heart_disease

FIGURE 11. Attributes in the adult and heart disease dataset.

In the UCI repository there exist 497 different datasets among which two are used for evaluating this work. These datasets are effective in evaluating the improvements of the used machine learning algorithms. The heart disease dataset specifically consists of four datasets as Cleveland, Hungary, Switzerland and VA Long Beach. The Cleveland dataset with 303 records (with 14 attributes each) are used for testing the DP-FCNN. On the other hand, we have taken 500 records from the Heart Disease dataset for testing this system. Apart from these two datasets, there exist many datasets that are required to be confidential. Here, heart disease is a health care related dataset collected from ill persons.

C. COMPARATIVE RESULTS

The comparative analysis presents the comparison of the proposed DP system with previously existing systems. The key metrics that are taken in account for comparison are,

Focused On	Method Used	Limitations
Personalized DP scheme [18]	Markov process	Distance based privacy level estimation is not suitable for all instances. Markov model was enabled to predict only confined distances
Machine learning based DP [19]	CART method. Deep neural network	Poor ϵ selection at initial stage due to the use of machine learning at first. Tree based weight computation is complex
Data analyst authentication [20]	Kd-tree. Elgamal encryption	Insecure credential (identity) for authentication. Tree fails to support a high dimensional dataset
Secure Storage [21]	AES algorithm	AES encryption involves multiple computations and hence consumes larger time. Absence of privacy at edge fails to protect private information

FIGURE 12. Previous research work in DP.

Scalability, Processing time, and Accuracy. Here, the major research papers that concentrate on providing DP are studied in this section and the existence of their problematic issues are solved with machine learning and lightweight algorithms. The common idea of DP is to protect private information with the assistance of adding noise.

A comparative study on prior research work in the field has been illustrated in Table 12. The limitations are problematic issues that the proposed research work aims to solve. Each parameter is evaluated and compared with previous work in the field.

1) SCALABILITY

Scalability is defined as an efficient measure that denotes the ability of the designed system to support an increasing number of input elements. The capacity of the system is improved based on the methods that are used for processing. The field of DP is applied for larger sized datasets that consist of thousands of records. DP is a security assisting topic that is envisioned to provide privacy for the personal details that exists in the dataset. For instance, a health oriented dataset includes private information of the patient along with their health problems. Hereby, the private information of a patient like name, gender, location and address are too confidential to be in public. So, a specified amount of noise is added in order to provide privacy for the personal information of the patient which will not reveal any information of the patient. The process responsible for performing Differential Privacy affects the system scalability. Hence, machine learning algorithms were used for improving scalability. This enabled faster processing and absolute decision making in the system even for increasing records.

The DP-FCNN achieves higher scalability and efficiency than the previous research work, as can be seen in Figure 13 and Figure 14, respectively. The scalability is measured in terms of the increasing number of records in the dataset with respect to the runtime. Similarly, the efficiency is measured by increasing the number of data analysts with respect to the runtime. The improvement in scalability and efficiency when compared to previous work is due to the following reasons:

- Selection of significant parameters that are relevant to the assurance of privacy is taken in account for applying DP

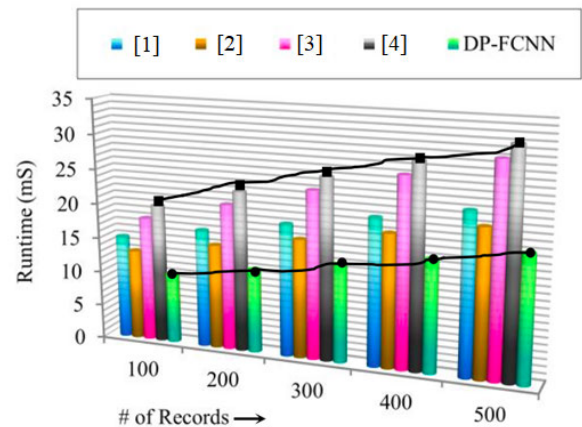


FIGURE 13. Comparison of scalability.

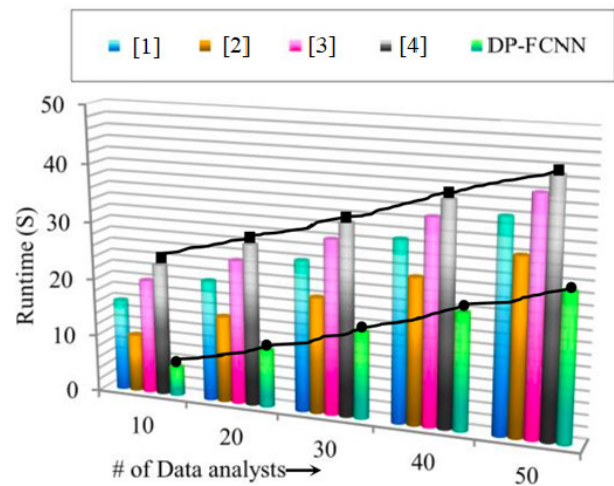


FIGURE 14. Comparison of efficiency.

- Efficient prediction of ϵ value using fast performing machine learning algorithms
- Faster searching by the appointment of the tree along with the assurance of security
- Use of lightweight hashing and cryptography algorithms are faster in processing and capable of processing many inputs

Work	Runtime (ms/s)	
	With respect to number of records	With respect to number of the data analysts
[30]	19	26
[31]	17	20
[32]	24	30
[33]	26	33
DP-FCNN	14	15

FIGURE 15. Average runtime.

As mentioned above, the use of machine learning and lightweight algorithms such as Piccolo are the key processes that reduce runtime, even with the increase in number of records and number of data analysts. The major issues in the previous work that cause larger runtime are,

- Involvement of algorithms with complex computations, consuming larger amounts of time for processing an individual data point and query
- Fails to predict exact ϵ value at the first attempt due to the absence of machine learning
- Incorporation of high convergence cryptography algorithm for security
- Lack of authentication makes the system less secure, which also increases the unnecessary runtime of the system

From this comparison, the proposed DP-FCNN shows improvements in scalability and efficiency. Hence this gradual increase in runtime is also capable of supporting further increase in records and analysts.

The average runtime that is taken in account for processing is illustrated in Table 6. The time taken to upload the dataset into the cloud is given with respect to increase in records and the increase in number of data analysts respectively. The previous work [19] using the CART method and deep neural network have reached nearer to DP-FCNN, where their difference is 5ms and 5s in their runtime for data processing and query processing. In contrast, all the other papers have higher runtime than this due to the absence of machine learning algorithms and use of poor parameters for preserving privacy.

2) PROCESSING TIME

The overall processing time of the system is measured for individual datasets to evaluate the performances. Figure 9 depicts the total processing time that is taken for processing the two datasets individually. Hereby the proposed DP-FCNN has lesser processing time than the previous research work due to the use of machine learning algorithms and lightweight methods for privacy.

The use of ML algorithms makes the computation faster even with the increase in the number of inputs. Thus the proposed DP-FCNN framework shows a shorter processing time than previous works. The processing time of dataset 1 is comparatively higher than dataset 2 since the number of

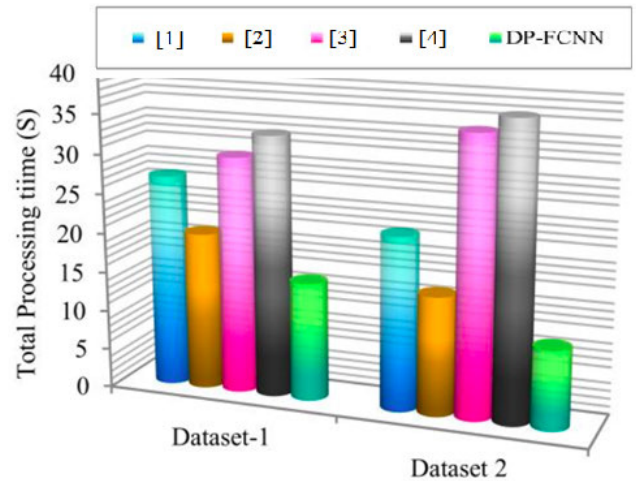


FIGURE 16. Comparison of Processing Time.

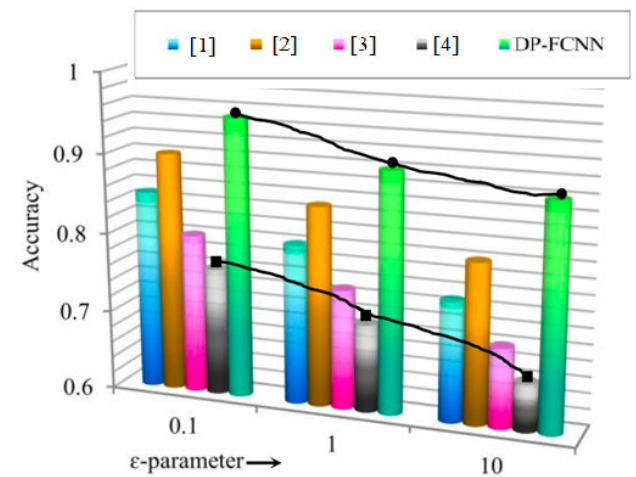


FIGURE 17. Comparison of accuracy.

ϵ	Accuracy (%)				
	RF	Linear SVM	RBF SVM	[32]	DP-FCNN
0.1	91.5	37.6	43.9	75.4	97
0.5	93.5	41.5	52.3	76	98
1	94.5	50.6	60	76.1	98

FIGURE 18. Comparison of accuracy with other machine learning Algorithms.

records in dataset 2 is lesser. The key benefits of minimizing the processing time of the system are:

- Capable of supporting thousands of datasets with considerable amount of processing time that mitigates the possibility of system failure.
- Enables access to incoming queries into the system from legitimate data analysts who have registered previously.
- Reduction in processing time also requires ensuring the provision of stronger privacy for the stored data and the query.

Issue	Solution Defined
Privacy for Sensitive Information	Use Differential Privacy in combination with Machine Learning for adding noise into the dataset. Incorporated fuzzy-CNN with the Laplace Mechanism
Insecure storage in public cloud	Use lightweight encryption that is stronger and efficient. Incorporated Piccolo Encryption Algorithm
Untrusted Edge Server	Secure request forwarding and searching. Incorporated XOR operator for credentials. Incorporated BLAKE2s hashing based Merkle hash tree construction.
Untrusted Data Analysts	Perform authentication using unique credentials. Credentials used are identity, password and fingerprint.

FIGURE 19. Proposed research solutions.

DP-FCNN is an efficient machine learning algorithm that combines fuzzy logic with CNN for adding noise by the Laplace mechanism. The addition of noise to the dataset, encryption before data storage, hashing index terms for searching gives an assurance of security and privacy as well as faster processing. Therefore the use of machine learning in adding noise for Differential Privacy is a promising solution for improving security and processing time of the system.

3) ACCURACY

Accuracy is a significant parameter that is computed for evaluating the efficiency of DP with respect to the ϵ value. The accuracy is higher when the ϵ value is smaller and as it increases the accuracy also decreases.

The comparison of accuracy between the proposed framework and previous works is illustrated in Figure 17. As per this result the accuracy of the proposed DP-FCNN is higher than previous work in the field, since Laplace mechanism is added based on the obtained fuzzy results that are determined from sensitivity and attributes present in the dataset. The increase in accuracy ensures the presented machine learning algorithm is stronger in providing security.

The above table depicts the comparison of accuracy with other commonly used machine learning methods in prior research work. From these results, the proposed DP-FCNN achieves 97-98% of accuracy which is not attained in any of the previous works. While using RF the accuracy is 91-94%, however it was not able to reach the results of DP-FCNN. All the other papers were able to attain the accuracy of 37-76%.

The performance metrics shown in the comparative analysis section shows the achievement of the proposed DP-FCNN over previous work in the field. This superior performance is also witnessed while processing with a real-time application.

D. KEY FINDINGS OF THE DP-FCNN FRAMEWORK

In this section, the key findings of the DP-FCNN are summarized:

- The provisioning of security with differential privacy requires selection of the amount of noise to be added which is achieved by the incorporation of a fuzzy convolution neural network that takes into account the sensitivity attribute for weight estimation. This is done right

after data is loaded, therefore ensuring no excess noise is added to the data.

- Edge devices are vulnerable to different attacks while using the edge server for searching the query in a hash tree. Due to this, the edge server has zero knowledge about the miner's query and the data extracted from the cloud. The data from the cloud is in an encrypted form which also increases data security
- Stronger credentials, which are difficult to be forged, have been used for the authentication of the data miner. This ensures only authenticated data miners are allowed into the system

On behalf of this DP-FCNN, the addition of noise provides privacy for the sensitive information followed by secure data storage and data searching.

The solutions defined in this paper are depicted in the above table. The framework is all-rounded as it concentrates on both data and query processing.

VI. CONCLUSION

In this paper, security, being a vital property to most frameworks, is ensured by using machine learning in collaboration with Differential Privacy. The proposed work focuses on data processing, that is, secure data uploading by the data owner and query processing, that is, secure data access by the data analyst. This system is designed with the participation of the following entities: the data owner, data provider, data analyst, edge server and the cloud. Differential Privacy (DP) is performed by the data provider, which is considered to be a trusted entity in this system. The FCNN is applied with the Laplace mechanism for addition of noise into the dataset in order to ensure privacy for personal information. The parameter is determined on the sensitivity attribute. The lightweight Piccolo algorithm is used for encrypting the data before uploading it to the public cloud. The public cloud is not trustable and hence the data is encrypted before storing. The Merkle hash tree is presented with the hashed key terms using the BLAKE2s algorithm for searching. The security credentials of the data analyst are validated before she is provided access to decrypt the data. The system design is tested and their results show better efficiency in terms of scalability, accuracy and processing time than previous work

in the field. In the future, the DP-FCNN is planned will be further developed to resolve single point failures by incorporating multiple trusted entities. The work can also be extended by the use of hybrid machine learning algorithms to add DP for privacy protection.

REFERENCES

- [1] R. Mo, J. Liu, W. Yu, F. Jiang, X. Gu, X. Zhao, W. Liu, and J. Peng, "A differential privacy-based protecting data preprocessing method for big data mining," in *Proc. 18th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Aug. 2019, pp. 693–699.
- [2] L. Zhou, L. Yu, S. Du, H. Zhu, and C. Chen, "Achieving differentially private location privacy in edge-assistant connected vehicles," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4472–4481, Jun. 2019.
- [3] H. Bao and R. Lu, "A lightweight data aggregation scheme achieving privacy preservation and data integrity with differential privacy and fault tolerance," *Peer Peer Netw. Appl.*, vol. 10, no. 1, pp. 106–121, Jan. 2017.
- [4] G. Eibl, K. Bao, P.-W. Grassal, D. Bernau, and H. Schmeck, "The influence of differential privacy on short term electric load forecasting," *Energy Informat.*, vol. 1, p. 48, Oct. 2018.
- [5] R. Civino, C. Blondeau, and M. Sala, "Differential attacks: Using alternative operations," *Des., Codes Cryptogr.*, vol. 87, nos. 2–3, pp. 225–247, Mar. 2019.
- [6] A. R. Chowdhury, C. Wang, C. He, A. Machanajhala, and S. Jha, "Crypte: Crypto-assisted differential privacy on untrusted servers," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, Jun. 2020, pp. 603–619.
- [7] L. Ni, C. Li, X. Wang, H. Jiang, and J. Yu, "DP-MCDBSCAN: Differential privacy preserving multi-core DBSCAN clustering for network user data," *IEEE Access*, vol. 6, pp. 21053–21063, 2018.
- [8] Y. Zhang, Z. Hao, and S. Wang, "A differential privacy support vector machine classifier based on dual variable perturbation," *IEEE Access*, vol. 7, pp. 98238–98251, 2019.
- [9] J. Zhao, Y. Chen, and W. Zhang, "Differential privacy preservation in deep learning: Challenges, opportunities and solutions," *IEEE Access*, vol. 7, pp. 48901–48911, 2019.
- [10] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 308–318.
- [11] Z. Song, Z. Li, and X. Chen, "Local differential privacy preserving mechanism for multi-attribute data in mobile crowdsensing with edge computing," in *Proc. IEEE Int. Conf. Smart Internet Things (SmartIoT)*, Aug. 2019, pp. 283–290.
- [12] M. Du, K. Wang, X. Liu, S. Guo, and Y. Zhang, "A differential privacy-based query model for sustainable fog data centers," *IEEE Trans. Sustain. Comput.*, vol. 4, no. 2, pp. 145–155, Apr. 2019.
- [13] Q. Miao, W. Jing, and H. Song, "Differential privacy-based location privacy enhancing in edge computing," *Recent Adv. Cloud Data Centers Toward Fog Data Centers*, vol. 31, no. 8, 2018, Art. no. e4735.
- [14] S. Yu, "Big privacy: Challenges and opportunities of privacy study in the age of big data," *IEEE Access*, vol. 4, pp. 2751–2763, 2016.
- [15] P. Jain, M. Gyanchandani, and N. Khare, "Differential privacy: Its technological prescriptive using big data," *J. Big Data*, vol. 5, no. 1, p. 15, Dec. 2018.
- [16] Z. Lv, L. Wang, Z. Guan, J. Wu, X. Du, H. Zhao, and M. Guizani, "An optimizing and differentially private clustering algorithm for mixed data in SDN-based smart grid," *IEEE Access*, vol. 7, pp. 45773–45782, 2019.
- [17] C. Yin, J. Xi, R. Sun, and J. Wang, "Location privacy protection based on differential privacy strategy for big data in industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3628–3636, Aug. 2018.
- [18] Y. Zhang, Y. Qu, L. Gao, T. H. Luan, X. Zheng, S. Chen, and Y. Xiang, "APDP: Attack-proof personalized differential privacy model for a smart home," *IEEE Access*, vol. 7, pp. 166593–166605, 2019.
- [19] Z. Sun, Y. Wang, M. Shu, R. Liu, and H. Zhao, "Differential privacy for data and model publishing of medical data," *IEEE Access*, vol. 7, pp. 152103–152114, 2019.
- [20] G. G. Dagher, B. C. M. Fung, N. Mohammed, and J. Clark, "SecDM: Privacy-preserving data outsourcing framework with differential privacy," *Knowl. Inf. Syst.*, vol. 62, pp. 1923–1960, May 2020.
- [21] T. Wang, Y. Mei, W. Jia, X. Zheng, G. Wang, and M. Xie, "Edge-based differential privacy computing for sensor-cloud systems," *J. Parallel Distrib. Comput.*, vol. 136, pp. 75–85, Feb. 2020.
- [22] M. Sridhar and B. Babu, "A fuzzy approach for privacy preserving in data mining," *Int. J. Comput. Appl.*, vol. 57, pp. 1–5, 2012.
- [23] K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita, and T. Shirai, "Piccolo: An ultra-lightweight blockcipher," in *Proc. Int. Conf. Cryptograph. Hardw. Embedded Syst. (CHES)*, 2011, pp. 342–357.
- [24] E. Popko and T. Weinstein, "Fuzzy logic module of convolutional neural network for handwritten digits recognition," in *Proc. 5th Int. Conf. Math. Modeling Phys. Sci. (IC-MSquare)*, May 2016, pp. 23–26.
- [25] *Dataset*. Accessed: Mar. 2, 2021. [Online]. Available: <https://archive.ics.uci.edu/ml/machine-learning-databases/adult/>
- [26] *Dataset*. Accessed: Mar. 2, 2021. [Online]. Available: <https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/>



JHILAKSHI SHARMA received the M.S. degree in computer science from Kennesaw State University, in July 2020, under the supervision of Dr. Donghyun Kim. Her research interests include cybersecurity and data science.



DONGHYUN KIM (Senior Member, IEEE) received the B.S. degree in electronic and computer engineering and the M.S. degree in computer science and engineering from Hanyang University, Ansan, South Korea, in February 2003 and February 2005, respectively, and the Ph.D. degree in computer science from The University of Texas at Dallas, Richardson, TX, USA, in May 2010. He is currently an Assistant Professor with the Department of Computer Science, Georgia State University (GSU), Atlanta, GA, USA. He is a Senior Member of ACM. He has served as the TPC Co-Chair for several international conferences, most recently IPCCC 2020 and COCOON 2020.



AHYOUNG LEE (Member, IEEE) received the M.S. and Ph.D. degrees in computer science and engineering from the University of Colorado at Denver, Denver, in 2006 and 2011, respectively. From 2013 to 2015, she was a Postdoctoral Fellow with the Broadband Wireless Networking Lab, Georgia Institute of Technology, under the supervision of Prof. Ian F. Akyildiz with a research project focused on software defined networking (SDN). She is currently an Assistant Professor with the Department of Computer Science, Kennesaw State University. Her current research interests include modeling and analysis with applications in SDN, mobile wireless networks, cyber-physical systems, future internet architecture for improving big data centers, the Internet of Things (IoT), and internet-centric technologies in cloud for network management.



DAEHEE SEO (Member, IEEE) received the B.S. degree in electronic and electrical engineering from Dongshin University, Naju, South Korea, in February 2001, and the M.S. degree in computer science and engineering and the Ph.D. degree in computer science from Soonchunhyang University, South Korea, in February 2003 and February 2006, respectively. He is currently an Assistant Professor with the Faculty of Artificial Intelligence and Data Engineering, Sangmyung University (SMU), Seoul, South Korea.

...