Secure Data Sharing in Smart Homes: An Efficient Approach Based on Local Differential Privacy and Randomized Responses

Amr T. A. Elsayed[1], Almohammady S. Alsharkawy[2], Mohamed S. Farag[3], S. E. Abo-Youssef[4]
Faculty of Science, Al-Azhar University, Cairo, Egypt[1,2,4]
Obour Heigh Institute for informatics, Cairo, Egypt[3]

Abstract—**Smart homes are smart spaces that contain devices that are connected to each other, collecting information and facilitating users' comfortable living, safety, and energy management features. To improve the quality of individuals' life, smart device companies and service providers are collecting data about user activities, user needs, power consumption, etc.; these data need to be shared with companies with privacy-preserving practices. In this paper, an effective approach of securing data transmission to the service provider is based on local differential privacy (LDP), which enables residents of smart homes to provide statistics on their power usage as disturbances bloom filters. Randomized Aggregatable Privacy-Preserving Ordinal (RAPPOR) is a privacy technique that allows sharing of data and statistics while preserving the privacy of individual users. The proposed approach applies two randomized responses: permanent random response (PRR) and instantaneous random response (IRR), then applies machine learning algorithms for decoding the perturbation bloom filters on the service provider side. The simulation results show that the proposed approach achieves good performance in terms of privacy-preserving, accuracy, recall, and f-measure metrics. The results indicate that, the proposed LDP for smart homes achieved good utility privacy when the value of LDP $\epsilon = 0.95$. The classification accuracy is between 95.4% and 98% for the utilized classification techniques.**

Keywords*—*Smart homes; security; privacy-preserving; differential privacy; RAPPOR; randomized responses

I. Introduction

As more people seek to automate their homes and improve their quality of life, the popularity of smart homes increases. A smart home is a residence that contains remote-controllable devices, such as smart thermostats, security systems, lighting, and entertainment systems. These devices are internet-connected, allowing homeowners to control them remotely using smartphones or other internet-connected devices [1]. Convenience is the main advantage of a smart home it enables users to control the temperature, lighting, and security of your smart home from anywhere in the world. The ability to turn off lights, adjust the temperature, and view security cameras from your smartphone makes it simple to keep your home comfortable and secure. Smart homes can also reduce your energy costs, smart thermostats can automatically adjust your home's temperature based on your preferences and your presence, saving you money on heating and cooling expenses. Similarly, intelligent lighting systems can turn off lights automatically when no one is in a room, thereby reducing energy consumption [2].

A smart home also provides better protection; with intelligent security systems, you can monitor your house from anywhere and receive warnings if suspicious behavior is detected [3]. You can also lock and unlock doors remotely, allowing you to let guests or service personnel in without being present. Smart homes can also improve your entertainment experience. You can control your television, music, and other entertainment systems from anywhere in your smart home. Even your smart home can be integrated with your voice assistant, making it simple to control your entertainment with voice commands.

Data gathered from smart homes can be used in a variety of ways to improve services across a range of industries. These services such as smart home activity prediction [4], smart healthcare for patient treatment [5], disorder assessment, and smart city pedestrian monitoring [6], energy management. In this context, businesses have discovered the potential of using the data gathered from smart homes to improve their products and services.

However, data collectors must consider the confidentiality of these data. If data is not correctly managed, it could cause significant issues. So, to address these concerns, a new system that maintains both privacy and utility has been proposed. Remote health systems necessitate the collection, disclosure, and utilization of personal health information, which raises grave privacy concerns. For many individuals, the household is their most private environment. A glucometer measuring the blood sugar level, a spirometer tracking the air entering and exiting the lungs, and a sleep monitoring sensor recording the sleep conditions can potentially reveal whether a resident has diabetes, seasonal allergy-induced asthma, or a depressive disorder. Patients are inclined to restrict access to these data to a small group, such as their personal physicians, out of concern for their privacy.

Differential Privacy [7] is a privacy preservation mechanism that has gained popularity. The main idea behind differential privacy is that a user is given plausible deniability by adding random values to their input. This approach provides strong privacy guarantees for users, protecting their data against adversary entities, such as service providers and outsiders. In the centralized differential privacy setting, noise is added to the database and apply a differential privacy aggregation algorithm. RAPPOR [8] is a privacy preservation technology that allows for the sharing of statistics while preserving the privacy of individual users. By using randomized response, RAPPOR ensures that no individual's data is disclosed to the data collector. This approach has shown great promise, as it allows for the sharing of valuable data while protecting users' privacy.

This paper aims to present an approach for securely transmitting household data to the aggregator, while accounting for the presence of malicious aggregator nodes. To address this concern, we apply LDP to the real-time data collected from residences. Prior to transmission, the data is subjected to a process of privacy preservation. The proposed model utilizes the RAPPOR algorithm to encrypt the data, thereby ensuring that the aggregator cannot ascertain the identity of the householder, thereby preserving anonymity. To achieve the goal of secure data transmission, we propose a three-step approach. Firstly, bloom Filter mechanism is applied to the raw data collected from the residences. Next, the data is privatized using the RAPPOR algorithm to ensure that the identity of the householder remains unknown. Finally, the aggregator employs machine learning algorithms to decode the data into a form that is acceptable and useful.

The proposed model has several advantages over existing approaches. By employing RAPPOR, we are able to ensure that the data is secure and anonymous, thereby preventing malicious aggregator nodes from accessing sensitive information. Moreover, the use of machine learning algorithms by the aggregator allows for efficient decoding of the data, making it more accessible and user-friendly. The remainder of this paper is organized as follows. Section II surveys existing privacy preservation techniques used in smart homes and highlighting their deficiencies. Section III gives the useful background about Local Differential Privacy and RAPPOR. Then our approach is described in Sections IV and introduces the system model. In section V, the performance of the scheme is analyzed from two aspects of security and efficiency.

II. Related Work

In recent years, LDP has emerged as a promising technique for privacy-preserving data analysis in various domains. This section provides an overview of some well-known LDP use cases and privacy-preserving systems that have used LDP. In [8] Google proposed RAPPOR as an LDP-based system for collecting aggregate statistics from users without compromising their individual privacy. It randomizes user responses to a question with a bloom filter and randomized response, allowing the server to compute meaningful statistics about the aggregate responses while ensuring individual privacy. The authors in [9] proposed an approach called a differential privacy-based system to guarantee thorough security for data produced by smart houses. At the aggregator level, they used the Hidden Markov Model (HMM) technique and applied differential privacy to the personal information obtained from smart homes.

In healthcare field the authors in [10] proposed an improved approach based on k-anonymity and differential privacy to enhance privacy protection by mitigating re-identification risks through generalization and suppression techniques. This study [11] concentrates primarily on identifying the security issues that can arise from the use of a large number of Internet of Things (IoT) devices connected to provide a smart home facility in Saudi Arabia. [12] proposed an approach called LATENT, suggests an intermediate layer in deep learning models that satisfies LDP. LATENT allows a data owner to perturb the data on their device before it reaches an untrusted machine learning service, thereby protecting the privacy of the owner's data. By adding noise to the data in a controlled manner, LATENT ensures that the machine learning model can still provide useful insights while preserving the privacy of the individual data points. In Microsoft, LDP is used to collect data about the time users spend in different applications, which enables the identification of their favorite ones and improves their user experience [13]. This approach still preserves user privacy while providing valuable insights for application developers. LDP has also been used to reduce potential privacy leakage in deep learning models.

Differential privacy is a privacy-preserving technique that has been extensively researched for various applications in computer science. One of the most popular applications is in recommendation systems, where differential privacy is used to protect the privacy of user preferences and behavior while still allowing the system to make accurate recommendations [14], [15]. Data mining is another field that benefits from differential privacy, as it allows for the analysis of sensitive data without revealing individual records [16]. Differential privacy is also used in crowd-sourcing [17]. In network measurements, differential privacy is used to ensure that the privacy of individuals' network traffic data is protected while still allowing for useful aggregate network measurements to be obtained [18]. In intelligent transportation systems, differential privacy is used to protect the privacy of users and their data [19].

These approaches have a few disadvantages or limitations compared to our approach, they uses a trusted third party to collect data from users, applies some algorithms, and takes some privacy-preserving data analysis by adding "noise". This lead to a reduction in the accuracy of data analysis and inference. The noise introduced to protect privacy may make it challenging to obtain precise information or draw accurate conclusions from the collected data. Our solution uses differential privacy at the data source, thereby providing greater privacy. In addition to the use of LDP, researchers have also put forth schemes that employ data masking techniques [20–24]. These approaches involve masking the data submitted by users with a specific masking value, ensuring that other entities cannot access the actual value unless they possess knowledge of the masking value. By incorporating data masking alongside LDP, these schemes offer an extra layer of privacy protection and enhance the security of sensitive information in the context of data sharing and analysis. In each of these applications, differential privacy should provide a way to perform valuable computations on sensitive data while ensuring that the privacy of individual users is protected. By adding controlled noise to the data, differential privacy makes it difficult for attackers to identify any specific individual in the dataset, while still allowing for meaningful analysis and insights to be drawn from the data.

III. Preliminaries

This section provides background information on LDP and the randomized response approach. It also discusses RAPPOR, which is a method for implementing the randomized response strategy. In addition to this, it investigates the machine learning methods that have been implemented, such as K-nearest neighbours (KNN), Support vector machines (SVMs) and XGBoost. In the final part of this discussion, we will examine the performance and assessment measures that are utilized in this paper to evaluate the performance of the proposed scheme.

A. Local Differential Privacy (LDP)

LDP is a privacy-preserving technique that aims to protect the privacy of individual data contributors while enabling statistical analysis and inference on the aggregated data. Unlike other privacy-preserving methods that rely on centralizing and anonymizing data, LDP allows data contributors to locally perturb their data before sharing it.

Definition: A randomized algorithm T satisfies the ϵ-local differential privacy where $\epsilon > 0$ if for all pairs of the client 's values a and b and for all $S \subseteq range(T)$:

$$Pr[T(a) \in S] \leq e^{\epsilon} Pr[T(b) \in S]. \tag{1}$$

The definition introduces ϵ, called the privacy budget. It quantifies the level of privacy protection provided. By satisfying ϵ-Differential Privacy, a mechanism provides a strong privacy guarantee, indicating that an adversary cannot significantly differentiate between the presence or absence of an individual's data based on the mechanism's output, thereby safeguarding individual privacy during data analysis or release.

B. Randomized Response

The Randomized Response (RR) method, introduced by H. Warner et al. in 1965 [7]. With RR, when an end user is asked a binary question (e.g., "yes" or "no"), a coin is flipped with a probability of p for heads. To maintain the user's privacy, RR allows the user to provide the opposite response when heads are shown. Consequently, the data aggregator is unable to confidently ascertain the true response for a specific user, ensuring their privacy is preserved.

Definition: The RR mechanism is a mapping with $X = Y$ that satisfies the following equality:

$$Q(x|y) \begin{cases} \frac{e^{\epsilon}}{|Y|-1+e^{\epsilon}}, & \text{if } x = y \\ \frac{1}{|Y|-1+e^{\epsilon}}, & \text{if } x \neq y \end{cases} \tag{2}$$

Here, $Q(x|y)$ is the conditional probability, Y is the true dataset, X is the privatized dataset, $y \in Y, x \in X$, $|Y|$ is the size of set Y, and ϵ is the privacy parameter.

C. RAPPOR

Privacy-Preserving Aggregatable Randomized Response is a real world application of LDP has been made by Google for collecting statistics from the end user, and client side software, in a way that provides robust privacy protection using randomize response techniques [8]. RAPPOR's applies randomized response to bloom filters [25] with strong ϵ-differential privacy guarantees. Bloom filter is a simple space-efficient randomized data structure for representing a set in order to support membership queries.

The RAPPOR algorithm takes in the client's true value v and parameters of execution k, h, f, p, q and is executed locally on the client's machine performing the following steps:

1) Signal: Hash client's value v onto the bloom filter B of size k using h hash functions.
2) Permanent randomized response: For each client's value v and bit $i, 0 \leq i < k$ in B, create a binary reporting value $B_i^{'}$ which equals to

$$B_i^{'} = \begin{cases} 1, & \text{with probability } \frac{1}{2}f \\ 0, & \text{with probability } \frac{1}{2}f \\ B_i, & \text{with probability } 1-f \end{cases} \tag{3}$$

where f is a user-tunable parameter controlling the level of longitudinal privacy guarantee. Subsequently, this $B^{'}$ is memoized and reused as the basis for all future reports on this distinct value v.
3) Instantaneous randomized response: Allocate a bit array S of size k and initialize to 0. Set each bit i in S with probabilities

$$P(S_i = 1) = \begin{cases} q, & \text{if } B_i^{'} = 1. \\ p, & \text{if } B_i^{'} = 0. \end{cases} \tag{4}$$

D. Machine Learning Techniques

The K-nearest neighbors (KNN) classifier is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining, and intrusion detection [26]. KNN algorithm helps us identify the nearest points or the groups for a query point. But to determine the closest groups or the nearest points for a query point we need some metric. For this purpose, we use below distance metrics:

$$d(x, y) = \left(\sum_{i=1}^{n} (x_i - y_i)^p \right)^{\frac{1}{p}} \tag{5}$$

Support vector machines: the support vector machines, is a powerful supervised learning algorithm used for classification tasks. It works by finding an optimal hyperplane in a high-dimensional feature space that separates different classes of data points. The hyperplane is chosen in such a way that it maximizes the margin, which is the distance between the hyperplane and the closest data points of each class [27]. This helps to achieve better generalization and robustness of the model.

Hyperplane Equation: The SVMs algorithm seeks to find a hyperplane in the feature space that separates the data points. The hyperplane equation can be written as:

$$w.x + b = 0 \tag{6}$$

where:

w is a weight vector orthogonal to the hyperplane. x is the feature vector of a given data point. b is the offset or distance of the hyperplane from the origin along the normal vector w.

Classification:

$$f(x) = sign(w.x + b) \tag{7}$$

where

$sign(.)$ is the sign function that returns *-1* or *1* depending on the sign of its argument. If $f(x) < 0$, the point is classified as one class, and if $f(x) > 0$, it is classified as the other class.

XGBoost is an implementation of gradient boosted decision trees. XGBoost models majorly dominate in many Kaggle Competitions. In this algorithm, decision trees are created in sequential form. Weights play an important role in XGBoost [28]. XGBoost objective function:

$$Obj^{(t)} = \sum_{i=1}^{n} L\left(y_i, \hat{y}_i^{(t-1)}\right) + \sum_{j=1}^{T} \Omega\left(f_j\right) \tag{8}$$

where

$Obj^{(t)}$ is the objective function at the tth iteration. n is the total number of training examples. y_i is the true label of the ith training example. $\hat{y}_i^{(t-1)}$ is the predicted value of the ith example at the $t-1$th iteration. T is the total number of trees in the ensemble. f_j is the jth tree in the ensemble. $\Omega\left(f_j\right)$ is the regularization term that penalizes the complexity of the tree.

E. Performance Evaluation Measurements

In this paper, the classifiers performance has been analyzed by using Precision, Recall and F-measure, which are obtained from the confusion matrix as shown in Table I. These metrics are described as follows.

TABLE I. CONFUSION MATRIX

	P‘(Predicted)	N‘(Predicted)
P (Actual)	TP	FN
N (Actual)	FP	TN

- Precision: measures the relevant actions found against all actions found i.e. the percentage of selected actions that are correct and is defined by the following equation.

$$Precision = \frac{TP}{(TP + FP)} \tag{9}$$

- Recall: measures the relevant actions found against all relevant actions i.e. the percentage of correct actions that are selected and is defined by the following equation.

$$Recall = \frac{TP}{(TP + FN)} \tag{10}$$

- F-measure: is weighted harmonic mean between precision and recall and is defined by the following equation.

$$F - measure = \frac{2 * Precision * Recall}{(Precision + Recall)} \tag{11}$$

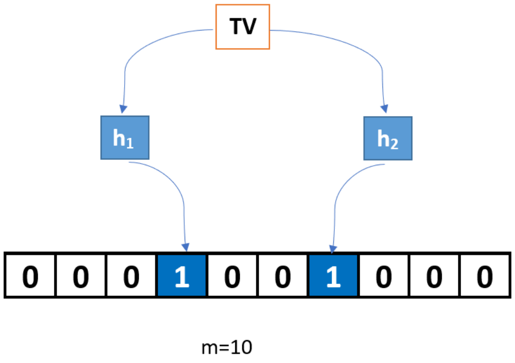

Fig. 1. Encoding algorithm maps C_i devices into bits in a bloom filter.

IV. THE PROPOSED PRIVACY-PRESERVING MODEL IN SMART HOMES

This section introduces and describe the proposed model and methodology. It outlines the key concepts and principles that underpin our approach, as well as explain the data collection process, preprocessing techniques used, and any specific algorithms or techniques used within the model.

A. Assumptions

Assume that there is a smart home which contains a set of devices and each device has its own sensor to measure power consumption, and there is a set of classes $C_i, i \in \{1, \ldots, l\}$ where l denotes the maximum number of supported classes, these classes represent the priority of each device, each class is composed of groups of devices $D_{ij}(i.e.\ j \in \{1 \ldots g\})$ where g represents the maximum number of groups of interests for class i, devices like (lights, TVs, laptops, sound system, alert systems, air condition systems, laundry devices, camera systems, garden lights, garage lights/motors, fans). RAPPOR is used to send power consumption statistics to service providers/electricity products companies.

B. System Model and Overview

Based on the basic idea of bloom filter and RAPPOR, the proposed approach consists of two phases: data perturbation phase and decoding phase. These two phases are described as follows:

1) Smart home data perturbation: In this phase, the proposed approach determines the set of devices and each device has its own sensor to measure power consumption, and there is a set of classes $C_i, i \in \{1, \ldots, l\}$ where l denotes the maximum number of supported classes, classes represent the priority of each device, each class is composed of groups of devices $D_{ij}(i.e. j \in \{1 \ldots g\})$ where g represents the maximum number of devices of interests for class i. The proposed approach uses the following steps.

1) *Encoding* is the first step of the data perturbation process, the encoding algorithm maps C_i devices into bits in a bloom filter. Fig. 1 illustrates the bloom filter implementation using *2* hash functions, h_1 and h_2, on the TV class.

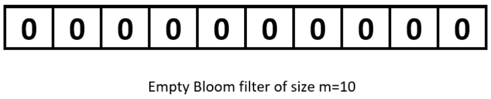

Fig. 2. Bloom filter B is initialized with all “0” values and size $m = 10$.

To compute the optimal bloom filter size m, given the maximum number of devices encoded into the bloom filter ($i.e.$ $n = l * g$ if each class includes the same number of groups), and a fixed false positive rate f_p, according to Eq. 12:

$$m = -\frac{n \times ln(f_p)}{(ln\,2)^2} \tag{12}$$

Then, the optimal number of hash functions is computed as given by Eq. 13

$$k = \frac{m}{n} \times ln\,2 \tag{13}$$

After selection of m and k appropriate values, the bloom filter B is initialized with all “0” values, as given in Fig. 2. For feeding B with the set of devices values v, C first applies the k hash functions to v, and feeds B with the hash output providing indices. Example: let us consider a bloom filter of size $m = 10$ bits, with $k = 2$ hash functions (h_1, h_2), and two devices {TV, Reception lights} to be included into the bloom filter. As given in Fig. 1, C first computes the two hashes of the TV device, and gets the results $h_1(TV) = 4$ and $h_2(TV) = 7$, thus leading to positioning the 4^{th} and the 7^{th} bits of the bloom filter to value 1. The same applies for the reception lights as depicted in Fig. 2 where h_1 (Reception lights) = 2 and h_2 (Reception lights) = 8.

2) *Permanent randomized response (PRR):* This first level perturbation applies over the bloom filter B obtained through the encoding phase. This step is executed once over a set of devices v. A noisy bit is derived from each bit of B thus resulting in a perturbed bloom filter vector B'. The derivation is compliant with the RAPPOR works and considers the following probabilistic processing:

$$B'[i] = \begin{cases} 1 & \text{with probability } \frac{1}{2}f \\ 0 & \text{with probability } \frac{1}{2}f \\ B[i] & \text{with probability } 1 - f \end{cases} \tag{14}$$

3) *Instantaneous randomized response (IRR):* To guarantee stronger privacy, this second level perturbation is executed for each request done by P Providers. After getting B', the user initializes a bit vector S with all zeros and then applies the following probabilistic processing Eq. 15:

$$P(S[i] = 1) \begin{cases} q & \text{if } B'[i] = 1 \\ p & \text{if } B'[i] = 0 \end{cases} \tag{15}$$

Where p denotes the probability of flipping a bit that equals to *0* into *1* whereas q represents the probability of keeping bits equal to *1*. This second level perturbation IRR algorithm is $\epsilon - differential$ privacy with the following quantified ϵ_2 privacy budget Eq. 16:

$$\epsilon_2 = k\,ln\left(\frac{q'(1-p')}{p'(1-q')}\right) \tag{16}$$

Where p', resp. q' is the probability of observing 1 given that the same bloom filter bit was set to *0*, resp. *1*, as defined in the following Eq. 17 and 18.

$$p' = \frac{1}{2}fq + (1 - \frac{1}{2}f)p \tag{17}$$

$$q' = (1 - \frac{1}{2}f)(1 - q) + \frac{1}{2}f(1 - p) \tag{18}$$

4) Algorithm 1, shows the steps of this recognition phase:

Parameters:

- hash functions k: This is the number of hash functions used in the bloom filter. The specific value is determined by Eq. 13.
- bloom filter size m: This is the size of the bloom filter, which is determined by Eq. 12.
- privacy budget ϵ: This is a parameter related to the privacy level. Its specific value is determined by the user or the privacy configuration.

Input:

- x: This is a single row of data from a smart home, representing a specific event or measurement.
- f: This represents the privacy level configured by the homeowner. Its specific value is not mentioned in the code.

Output:

- Perturbed bloom filter vector S: This is the resulting vector after applying perturbations to the bloom filter, calculated using Eq. 15.

Steps:

- Set $x \in \mathcal{U}$ this indicates that the smart home data, represented by x, belongs to the set $\mathcal{U}$, which includes all available data in the smart home.
- Convert x to bloom filter vector B of size m: This step involves converting the smart home data, x, into a bloom filter vector B, of a specified size m.
- Apply permanent randomized response on B and get vector B' of size m.
- Apply instantaneous randomized response on B' and return vector S of size m.

C. Decoding Phase

In this phase, three machine learning algorithms KNN, SVMs and XGBoost were selected for their ability to work on perturbed data. Those algorithms are calibrated to fit the specification the following datasets, thus resulting into *3* configurations as detailed below:

Algorithm 1: Data perturbation

```
Parameter: hash functions $k$ given by Eq. 13, bloom
    filter size $m$ given by Eq. 12, privacey
    budget $\epsilon$
Input: Row of smart home data $x$, $f$ is the privacy
    level configured by home owner.
Output: Perturbed bloom filter vector $S$ given by Eq.
    15
Data: set $x \in \mathcal{U} : \mathcal{U}$ is the set of all avilable data in
    smart home
/* Encoding                                        */
Convert $x$ to bloom filter vector $B$ of size $m$
/* PRR function                                    */
Initialize an empty vector $B'$ of size $m$ and set all
  $bits = 0$
for $i = 0$ to $BloomFilterSize$ do
    $B'[i] = 1$ with probability $\frac{1}{2}f$
    $B'[i] = 0$ with probability $\frac{1}{2}f$
    $B'[i] = B[i]$ with probability $1 - f$
/* IRR function                                    */
Initialize an empty vector $S$ of size $m$ and set all
  $bits = 0$
for $j = 1$ to $NumberOfhashfunctions$ do
    for $i = 1$ to $BloomFilterSize$ do
        if $B'[i] = 1$ then
            $S[i] = 1$ with probability $\frac{e^{\frac{\epsilon}{2k}}}{e^{\frac{\epsilon}{2k}}+1}$
        else
            $S[i] = 1$ with probability $\frac{1}{e^{\frac{\epsilon}{2k}}+1}$
Return vector $S$:
```

1) The K-nearest neighbors (KNN) classifier: is a versatile algorithm that classifies data based on the majority class of its K nearest neighbors in a training set, making it suitable for both classification and regression tasks.
2) Support vector machines: The Support vector machines works by finding an optimal hyperplane in a high-dimensional feature space that separates different classes of data points. The hyperplane is chosen in such a way that it maximizes the margin, which is the distance between the hyperplane and the closest data points of each class. This helps to achieve better generalization and robustness of the model.
3) XGBoost configuration: XGBoost is a gradient boosting algorithm. Table III gives the parameters calibrated for each dataset to optimize the model's performances. As can be shown, the configuration is slightly the same, except for parameter Subsample.

V. Experimental and Analysis

To evaluate the proposed approach, a real dataset The MHEALTH [29] is used. It is a data file consisting of approximately 1 million records. The data primarily consists of numerical values. Specifically, it is referred to as the "Mobile HEALTH" dataset, which captures body motion and vital signs recordings. The dataset encompasses measurements from ten volunteers with diverse profiles while engaging in various physical activities. Also colab notebook is used. Colab [30] is a research initiative for prototyping machine learning models on powerful hardware such as GPUs and TPU. Tables II and III provide the SVMs and XGBoost parameters, as well as the KNN with $K = 3$. These machine learning algorithms are used in the proposed method to test shred data in smart home environments.

TABLE II. SVMs Configuration

Parameter	Value
SVM Type	rbf
C	1000
Gamma	0.4

TABLE III. XGBOOST Configuration

Parameter	Value
N estimators	55
Max depth	6
Min child weight	7
num rounds	10
Gamma	0.4

A. Classification Evaluation

This subsection analyses the influence of different parameters on the classification results, including the privacy budget value ϵ, the bloom filter size M and the number of hash functions k. Knowing that the accuracy for the dataset without applying LDP were KNN: 97.8%, SVMs: 98.5% and XGBoost: 98%.

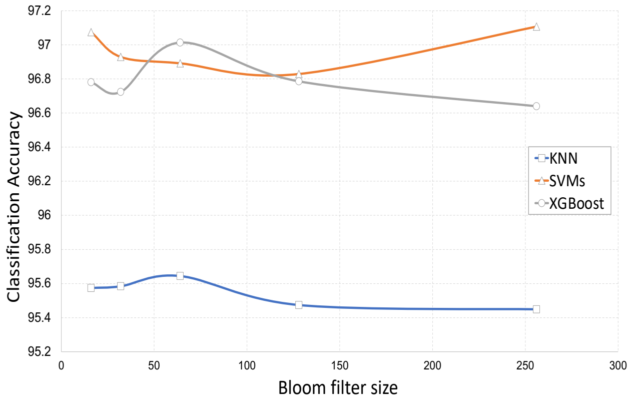

Fig. 3. Bloom filter size for ϵ =0.95 and k = 5].

Fig. 3 demonstrate the relationship between the accuracy of various classification methods and the size of the bloom filter. The bloom filter size ranges from 8 to 256, and it achieves accuracy within the following ranges: KNN (95.4%-95.7%), SVMs (96.8%-97.15%) and XGBoost (96.6%-97.0%). When comparing these results with the accuracy obtained without applying LDP using the same machine learning algorithms (97.8%, 98.5%, and 98% respectively), there is an error margin of approximately 2%. However, this level of error does not significantly impact the overall accuracy.

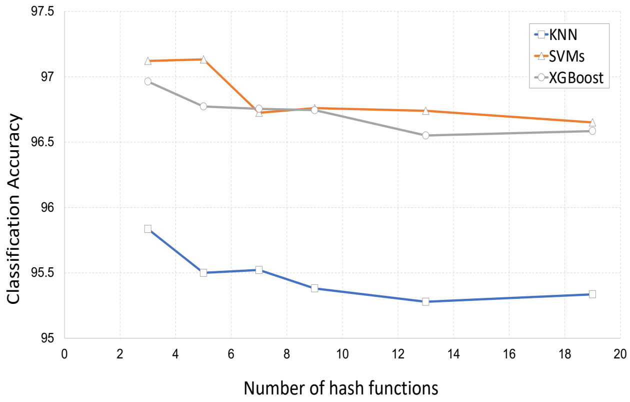

Fig. 4. Number of hash functions for ϵ =0.95 and M = 128].

Fig. 4 shows the relationship between the accuracy of various classification methods and the number of hash functions. The number of hash functions ranges from *3* to *19*, and it achieves accuracy within the following ranges: KNN (95.2%-95.8%), SVMs (96.65%-97.13%) and XGBoost (96.55%-96.9%). Also, the error margin is approximately (1%-2%) between the proposed LDP approach and without applying LDP using the same machine learning algorithms

Our experiment considers a minimum value of hash functions of *5*, which corresponds to the optimal number of hash functions for M = *128*, according to the Eq. 13. As depicted in the Fig. 4, the classification accuracy decreases when the number of hash function increases. This stems from an increasing number of hash collisions.

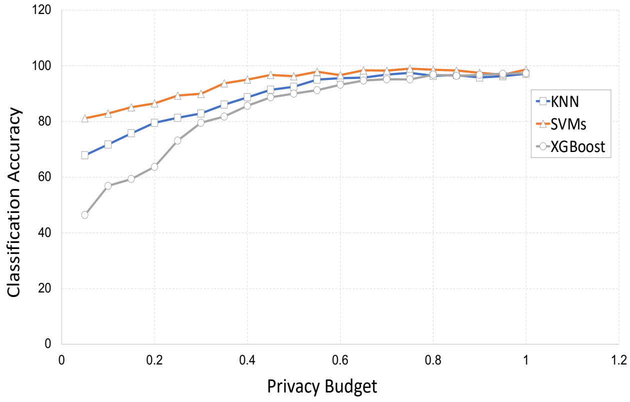

Fig. 5. Privacy budget for M =128 and K = 5].

The relation between the accuracy of different classification algorithms and the Privacy budget is depicted in Fig. 5. The privacy budget varies from *0.1* to *1*, and it achieves accuracy within the following ranges: KNN (67.2%-97.4%), SVMs (81.65%-99.13%) and XGBoost (46.42%-97.3%). When comparing these results with the accuracy obtained without applying LDP using the same machine learning algorithms (97.8%, 98.5%, and 98% respectively).

As expected in Fig. 5, the classification accuracy is an increasing function of the privacy budget. Indeed, the higher the privacy budget, the lower the perturbation level, and the higher the accuracy. The preference dataset achieves better classification results.

B. Decoding Algorithms Evaluation

Table IV shows the accuracy of various classification methods on perturbed dataset using bloom filter size $M = 128$, privacy budget $\epsilon = 0.95$ and number of hash functions $k = 5$. Table V shows the accuracy of the same classification methods on the main dataset. As shown in both tables, an analysis of accuracy comparisons for main and perturbed data utilizing KNN, SVMs and XGBoost algorithms. This study evaluates the accuracy performance of KNN, SVMs and XGBoost algorithms when applied to a dataset consisting of 24,000 records and encompassing 12 distinct activities. The comparison focuses on the accuracy of predictions made using both the original dataset and a perturbed version.

Table VI illustrates the error margin between the accuracy of various classification methods on perturbed data and main dataset. The findings of this analysis indicate that the application of LDP techniques on the dataset did not introduce any significant impact on the decision-making process. The accuracy levels observed for the main dataset and the perturbed data remained consistent across the evaluated algorithms, namely KNN, SVMs, and XGBoost.

C. Security Analysis

In this part, we undertake security analysis using the fundamental adversary model. This model assumes that the attacker has access to the altered data disclosed by different individuals through the Local Differential Privacy (LDP) method. The primary objective is to ensure that despite the adversary's access to the perturbed data and some knowledge of the noise introduced during the LDP procedure, inferring sensitive information about any individual remains computationally infeasible or statistically improbable. The success rate of basic adversary can directly be obtained from the probability of Eq. 19 [31]

$$Pr(B'[i]=1)\begin{cases}\frac{e^{\frac{\epsilon}{2k}}}{e^{\frac{\epsilon}{2k}}+1} & \text{if } B[i]=1\\ \frac{1}{e^{\frac{\epsilon}{2k}}+1} & \text{if } B[i]=0\end{cases} \tag{19}$$

Fig. 6 illustrates the relation between the success rate of basic adversary and ϵ and k values. The privacy budget ranges from *0.1* to *4*, and the number of hash functions are ($k = 2, k = 7, k = 12$). This figure indicates that as the privacy budget increases, the probability that the adversary will win in the game also increases. However, as the number of hash functions increases, more wrong guesses occur. As expected, the probability of winning the game decreases when ϵ and k increase.

TABLE IV. Accuracy for Perturbed Data using Bloom Filter Size $M = 128$, Privacy Budget $\epsilon = 0.95$ and Number of Hash Functions $k = 5$, KNN, SVMs and XGBoost, 24,000 Records and 12 Activities

	Precision			Recall			F1-score			Support		
	KNN	XGBoost	SVMs	KNN	XGBoost	SVMs	KNN	XGBoost	SVMs	KNN	XGBoost	SVMs
Standing still (1 min)	1	1	1	1	1	1	1	1	1	1121	1121	1121
Sitting and relaxing (1 min)	1	1	1	1	1	1	1	1	1	542	542	542
Lying down (1 min)	1	1	1	1	0.99	1	1	1	1	544	544	544
Walking (1 min)	0.97	0.93	0.97	0.94	0.99	0.94	0.96	0.96	0.96	567	567	567
Climbing stairs (1 min)	0.97	0.97	0.97	0.95	0.89	0.95	0.96	0.93	0.96	610	610	610
Waist bends forward (20x)	0.99	0.99	0.99	0.98	1	0.98	0.98	0.99	0.98	567	567	567
Frontal elevation of arms (20x)	0.98	0.96	0.98	0.98	0.99	0.98	0.98	0.98	0.98	543	543	543
Knees bending (crouching) (20x)	0.97	0.95	0.97	0.98	0.96	0.98	0.98	0.96	0.98	569	569	569
Cycling (1 min)	1	1	1	1	1	1	1	1	1	581	581	581
Jogging (1 min)	0.92	0.83	0.92	0.92	0.92	0.92	0.92	0.87	0.92	576	576	576
Running (1 min)	0.96	0.93	0.96	0.94	0.88	0.94	0.95	0.9	0.95	596	596	596
Jump front & back (20x)	0.9	0.91	0.9	0.94	0.84	0.94	0.92	0.87	0.92	594	594	594
Weighted Avg	0.972	0.956	0.972	0.969	0.955	0.969	0.971	0.955	0.971	7410	7410	7410

TABLE V. Accuracy for Main Data using KNN, SVMs and XGBoost Algorithms, 24,000 Records and 12 Activities

	Precision			Recall			F1-score			Support		
	KNN	XGBoost	SVMs	KNN	XGBoost	SVMs	KNN	XGBoost	SVMs	KNN	XGBoost	SVMs
Standing still (1 min)	1	1	1	1	1	1	1	1	1	922	922	922
Sitting and relaxing (1 min)	1	1	1	0.99	0.99	0.99	1	1	1	488	488	488
Lying down (1 min)	1	1	1	1	1	1	1	1	1	450	450	450
Walking (1 min)	1	1	1	1	1	1	1	1	1	449	449	449
Climbing stairs (1 min)	1	1	1	1	0.96	1	1	0.98	1	443	443	443
Waist bends forward (20x)	1	1	1	1	1	1	1	1	1	444	444	444
Frontal elevation of arms (20x)	1	0.99	1	0.99	0.99	0.99	1	0.99	1	438	438	438
Knees bending (crouching) (20x)	1	0.96	1	1	0.99	1	1	0.97	1	432	432	432
Cycling (1 min)	1	1	1	1	1	1	1	1	1	447	447	447
Jogging (1 min)	0.98	0.92	0.98	1	0.98	1	0.99	0.95	0.99	425	425	425
Running (1 min)	1	0.98	1	0.98	0.94	0.98	0.99	0.96	0.99	458	458	458
Jump front & back (20x)	1	0.97	1	1	0.96	1	1	0.96	1	454	454	454
Weighted Avg	0.998	0.985	0.998	0.997	0.984	0.997	0.998	0.984	0.998	5850	5850	5850

TABLE VI. The Error Margin between the Accuracy of Various Classification Algorithms KNN, SVMs and XGBoost on Perturbed Data and Main Dataset, 24,000 Records and 12 Activities

	Precision			Recall			F1-score		
	KNN	XGBoost	SVMs	KNN	XGBoost	SVMs	KNN	XGBoost	SVMs
Standing still (1 min)	0	0	0	0	0	0	0	0	0
Sitting and relaxing (1 min)	0	0	0	0	0	0	0	0	0
Lying down (1 min)	0	0	0	0	0.01	0	0	0	0
Walking (1 min)	0.03	0.07	0.03	0.06	0.01	0.06	0.04	0.04	0.04
Climbing stairs (1 min)	0.03	0.03	0.03	0.05	0.07	0.05	0.04	0.05	0.04
Waist bends forward (20x)	0.01	0.01	0.01	0.02	0	0.02	0.02	0.01	0.02
Frontal elevation of arms (20x)	0.02	0.03	0.02	0.01	0	0.01	0.02	0.01	0.02
Knees bending (crouching) (20x)	0.03	0.01	0.03	0.02	0.03	0.02	0.02	0.01	0.02
Cycling (1 min)	0	0	0	0	0	0	0	0	0
Jogging (1 min)	0.06	0.09	0.06	0.08	0.06	0.08	0.07	0.08	0.07
Running (1 min)	0.04	0.05	0.04	0.04	0.06	0.04	0.04	0.06	0.04
Jump front & back (20x)	0.1	0.06	0.1	0.06	0.12	0.06	0.08	0.09	0.08
Weighted Avg	0.026666667	0.029166667	0.026666667	0.0275	0.029166667	0.0275	0.0275	0.029166667	0.0275

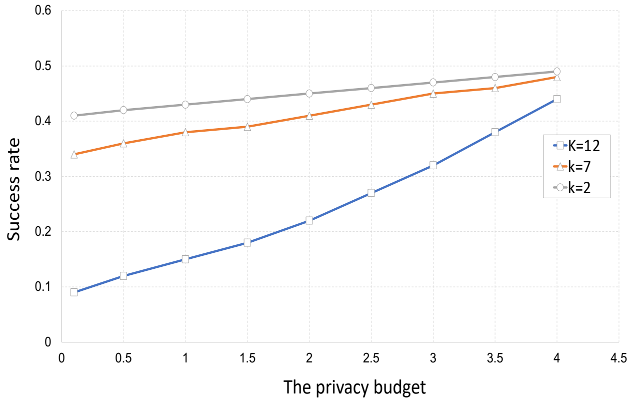

Fig. 6. Success rate over one record of perturbed data by a Basic Adversary.

VI. Conclusion

In this study, we investigated the problem of sharing data in a smart home environment while preserving user privacy. The main contribution of this research is the development of an efficient method for secure data sharing in smart homes using local differential privacy and the Randomized Aggregatable Privacy-Preserving Ordinal technology. Individual users' privacy is protected while data sharing with service providers is facilitated by the proposed method. The simulation results demonstrate that the technique performs well in terms of privacy preservation, accuracy, recall, and f-measure metrics, achieving utility privacy with high classification accuracy of 95.4% to 98% when the privacy budget is set to 0.95. This research helps to improving data privacy and utility in the context of smart homes, as well as providing a valuable direction for privacy-preserving practices in the IoT domain. In future research, we aim to extend the research to consider privacy-preserving techniques for multi-modal data, such as combining data from various sensors and devices within a smart home environment.

References

[1] T. Denning, T. Kohno, and H. M. Levy, "Computer security and the modern home," *Communications of the ACM*, vol. 56, no. 1, pp. 94–103, 2013.

[2] H. Youssef, S. Kamel, M. Hassan, and L. Nasrat, "Optimizing energy consumption patterns of smart home using a developed elite evolutionary strategy artificial ecosystem optimization algorithm," *Energy*, p. 127793, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360544223011878

[3] O. Taiwo, A. Ezugwu, O. Oyelade, and M. Almutairi, "Enhanced intelligent smart home control and security system based on deep learning model," *Wireless communications and mobile computing*, vol. 2022, pp. 1–22, 2022.

[4] S. Zhang, W. Li, Y. Wu, P. Watson, and A. Zomaya, "Enabling edge intelligence for activity recognition in smart homes," in *2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 2018, pp. 228–236.

[5] G. Muhammad, M. F. Alhamid, M. Alsulaiman, and B. Gupta, "Edge computing with cloud for voice disorder assessment and treatment," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 60–65, 2018.

[6] J. Lwowski, P. Kolar, P. Benavidez, P. Rad, J. J. Prevost, and M. Jamshidi, "Pedestrian detection system for smart communities using deep convolutional neural networks," in *2017 12th System of Systems Engineering Conference (SoSE)*. IEEE, 2017, pp. 1–6.

[7] C. Dwork, "Differential privacy. automata, languages and programming-icalp 2006, lncs 4052," 2006.

[8] Ú. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, 2014, pp. 1054–1067.

[9] N. Waheed, F. Kha, M. Jan, A. Z. Alalmaie, and P. Nanda, "Privacy-enhanced living: A local differential privacy approach to secure smart home data," *arXiv preprint arXiv:2304.07676*, 2023.

[10] R. Ratra, P. Gulia, and N. S. Gill, "Evaluation of re-identification risk using anonymization and differential privacy in healthcare," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 2, 2022. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2022.0130266

[11] O. Almutairi and K. Almarhabi, "Investigation of smart home security and privacy: Consumer perception in saudi arabia," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 4, 2021. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2021.0120477

[12] P. Arachchige, P. Bertok, I. Khalil, D. Liu, S. Camtepe, and M. Atiquzzaman, "Local differential privacy for deep learning," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5827–5842, 2019.

[13] B. Ding, J. Kulkarni, and S. Yekhanin, "Collecting telemetry data privately," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[14] F. McSherry and I. Mironov, "Differentially private recommender systems: Building privacy into the netflix prize contenders," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 627–636.

[15] S. Rahali, M. Laurent, S. Masmoudi, C. Roux, and B. Mazeau, "A validated privacy-utility preserving recommendation system with local differential privacy," in *2021 IEEE 15th International Conference on Big Data Science and Engineering (BigDataSE)*. IEEE, 2021, pp. 118–127.

[16] A. Friedman and A. Schuster, "Data mining with differential privacy," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 493–502.

[17] J. Hamm, A. C. Champion, G. Chen, M. Belkin, and D. Xuan, "Crowd-ml: A privacy-preserving learning framework for a crowd of smart devices," in *2015 IEEE 35th International Conference on Distributed Computing Systems*. IEEE, 2015, pp. 11–20.

[18] A. Mani and M. Sherr, "Histor varepsilon : Differentially private and robust statistics collection for tor." in *NDSS*, 2017.

[19] F. Kargl, A. Friedman, and R. Boreli, "Differential privacy in intelligent transportation systems," in *Proceed-*

ings of the sixth ACM conference on Security and privacy in wireless and mobile networks, 2013, pp. 107–112.

[20] P. Gope and B. Sikdar, "An efficient data aggregation scheme for privacy-friendly dynamic pricing-based billing and demand-response management in smart grids," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3126–3135, 2018.

[21] W. Jia, H. Zhu, Z. Cao, X. Dong, and C. Xiao, "Human-factor-aware privacy-preserving aggregation in smart grid," *IEEE Systems Journal*, vol. 8, no. 2, pp. 598–607, 2013.

[22] H. Bao and R. Lu, "Ddpft: Secure data aggregation scheme with differential privacy and fault tolerance," in *2015 IEEE International Conference on Communications (ICC)*. IEEE, 2015, pp. 7240–7245.

[23] C. Castelluccia, A. C. Chan, E. Mykletun, and G. Tsudik, "Efficient and provably secure aggregation of encrypted data in wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 3, pp. 1–36, 2009.

[24] L. Lyu, K. Nandakumar, B. Rubinstein, J. Jin, J. Bedo, and M. Palaniswami, "Ppfa: Privacy preserving fog-enabled aggregation in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3733–3744, 2018.

[25] B. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[26] V. Prasatha, H. Alfeilate, A. Hassanate, O. Lasassmehe, A. Tarawnehf, M. Alhasanatg, and H. Salmane, "Effects of distance measure choice on knn classifier performance-a review," *arXiv preprint arXiv:1708.04321*, p. 56, 2017.

[27] S. Yue, P. Li, and P. Hao, "Svm classification: Its contents and challenges," *Applied Mathematics-A Journal of Chinese Universities*, vol. 18, pp. 332–342, 2003.

[28] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[29] A. S. Oresti Banos, Rafael Garcia, "Mhealth dataset," UCI Machine Learning Repository, 2014, dOI: https://doi.org/10.24432/C5TW22.

[30] E. Bisong and E. Bisong, "Google colaboratory," *Building machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners*, pp. 59–64, 2019.

[31] M. E. Gursoy, A. Tamersoy, S. Truex, W. Wei, and L. Liu, "Secure and utility-aware data collection with condensed local differential privacy," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 2365–2378, 2019.