IJAMEC
e-ISSN: 2147-8228

INTERNATIONAL JOURNAL OF APPLIED MATHEMATICS ELECTRONICS AND COMPUTERS

www.dergipark.org.tr/ijamec

International Open Access

Volume 08
Issue 04

December, 2020

Research Article

A hybrid approach of homomorphic encryption and differential privacy for privacy preserving classification

Ezgi Zorarpacı [a,*], ***Selma Ayşe Özel*** [b]

[a]*Faculty of Aeronautics and Astronautics, Iskenderun Technical University, Iskenderun, Hatay, Turkey*
[b]*Department of Computer Engineering, Cukurova University, Adana, Turkey*

ARTICLE INFO

Article history:
Received 28 September 2020
Accepted 11 October 2020

Keywords:
Differential privacy
Homomorphic encryption
Privacy preserving classification
One Rule
Naïve Bayes

ABSTRACT

Privacy preserving data mining is a substantial research area that aims at protecting the privacy of individuals while enabling to perform data mining techniques. In this study, we propose a secure protocol that fulfils the privacy restriction by combining homomorphic encryption with differential privacy and integrate this protocol into Holte's One Rule which is a simple, but accurate and efficient classification algorithm. The proposed method allows a researcher to get the answers of his/her queries to build One Rule classifier by processing the encrypted training dataset under Paillier's cryptosystem and also applies differential privacy to minimize the privacy leakage of individuals as much as possible in this training dataset. Therefore, both of security and privacy of the individuals in the training dataset for classification are provided thanks to our proposed method; since neither the parties, nor the researcher attain any information about the individuals in the database. Besides the One Rule classifier, we apply our proposed privacy preservation model to Naïve Bayes classification algorithm for the performance comparison, and show the efficiency of the proposed method through experiments on real data from UCI repository.

1. Introduction

Data mining is the process of discovering beneficial information from big quantity of data. The extracted information can be patterns, rules, clusters or a classification model. Throughout the data mining process, sensitive information of individuals are subject to several parties such as data collectors, data miners and users of data mining operations. Consequently, privacy preserving data mining has emerged as a significant sub-field of the data mining to protect sensitive personal information from various data mining parties. Privacy preserving data mining is interested in maintaining data mining techniques without disclosing the privacy of individual data or sensitive information [1].

In this study, we propose a privacy preserving version of Holte's One Rule (1R) [2] algorithm, which is a simple and short, but efficient and accurate classifier. We combine 1R algorithm with homomorphic encryption and differential privacy which are the two most popular privacy preservation techniques to develop privacy preserving 1R classification algorithm. Homomorphic encryption is a variant of encryption, which enables making calculations on ciphertext domain and produces an encrypted result, such that the obtained encrypted results match with the result of the computations as if these computations were fulfilled on the plaintext domain. Differential privacy, which is a strong privacy guarantee, has been proposed to perform data mining algorithms over data that contain sensitive information. Differential privacy determines the rate of private information leakage by using an ϵ parameter. Differential privacy perturbs the results of queries that run on the sensitive data by adding a noise (such as Laplace), and

* Corresponding author. E-mail address: *ezgi.zorarpaci@iste.edu.tr*
DOI: 10.18100/ijamec.801157

the amount of noise to be added is determined by the size of the ϵ parameter. There is an inverse proportion between the amount of noise added and the size of the ϵ parameter. The lower values of ϵ parameter increases the noise, therefore lower classification accuracy values are observed with higher privacy levels. On the other hand, the higher the values of ϵ parameter are, the higher classification accuracy results are obtained but with the less privacy.

1R is a classification model induction algorithm that finds the most informative attribute in the data and classifies instances with the values of this single attribute. 1R generates very few classification rules that are very easy to interpret for humans [2]. During the construction of 1R classifier, frequencies of predictors (i.e., attributes) for each class in the database are needed. Despite this necessity, the individuals in the database are concerned that their confidential information may be disclosed, misused or abused. Therefore, we propose a privacy preserving data mining approach which employs a combination of homomorphic encryption (i.e., Paillier's cryptosystem) [3] and differential privacy [4, 5, 6] to extract the classification rules by using 1R algorithm in this study. In our approach, the training data for classification is kept in an encrypted database to provide data confidentiality, and by the help of both homomorphic encryption and differential privacy, frequencies of attribute values (i.e., results of count queries) are computed from this encrypted database without seeing the original data. As we employ two parties (i.e., homomorphic encryption and differential privacy) at the same time for the computation of the attribute value frequencies, even if a malicious user disrupts one of these parties, he/she will not see the original data and will not know the actual frequency values since differential privacy adds some noise on the results of count queries used for finding the frequencies.

Basically, our proposed method supports the following two vital aspects of security:

i) Data privacy. The training dataset for classification task is securely stored by employing Paillier's cryptosystem. Thanks to the two-party computation, the confidentiality of the training data is maintained even if one of the parties is hacked.

ii) Output privacy. Differential privacy provides a strong privacy guarantee by perturbing the actual query result while maintaining the utility of the result. Thanks to the proposed two-party computation, neither the parties nor the researcher won't learn the actual query results as long as these two parties cooperate.

In the literature, the differentially private implementations of well-known classification algorithms such as decision trees, random forests, random trees, Naïve Bayes, and k-NN have been proposed. However, any implementation of private Holte's 1R classifier [2] which is a simple and short, but efficient and accurate algorithm has not been studied so far to our best knowledge. To cover this gap, we propose to develop a private 1R algorithm. At the same time, none of the existing techniques take into account of data privacy while ensuring output privacy.

The contributions of this study can be given briefly as follows: We present a privacy preserving protocol for building of 1R classification algorithm. This protocol combines two most popular privacy preservation techniques that are homomorphic encryption and differential privacy. Our proposed method ensures the data privacy and output privacy simultaneously for classification task unlike existing private algorithms in the literature. To our best knowledge, the proposed method is the first study to perform a private 1R classification algorithm by using homomorphic encryption and differential privacy together.

Our proposed privacy protocol is also incorporated into Naïve Bayes classification algorithm as well to evaluate the classification performance of the proposed 1R classifier, since Naïve Bayes is a baseline differentially private classifier in the literature and requires attribute value frequencies (i.e., count queries) to calculate the class probabilities during construction of the classifier. Through the experiments, we demonstrate the effectiveness and applicability of our proposed approach in terms of classification accuracy, run-time complexity, and storage requirement.

The rest of the paper is organized as follows: the previous studies in the related fields are summarized in Section 2, the basic concepts of Paillier's homomorphic cryptosystem, differential privacy, and 1R classification algorithm are given in Section 3. In Section 4, the proposed method is explained in detail. In Section 5, the experimental results including the security analysis and run-time analysis of the proposed method are presented; also the storage requirements of the method for the encryption of training data for classification are given. Finally, the paper is concluded in Section 6.

2. Related works

2.1. Securely computing count queries

One of the most prevalent obligation in data mining is querying the number of instances in the dataset that satisfy the condition given by the data miners (i.e., researchers). Accordingly, it becomes a necessity to obtain such knowledge from data without revealing the confidential information of individuals. One of the earliest method to answer such count queries securely without seeing the data of individuals in a database of DNA sequences has been proposed by Kantarcioglu et al. [7], which is based on a cryptographic model. In this study, two parties are considered to compute frequencies

(i.e., count queries). One party (i.e., data storage site) is liable for storing encrypted data acquired from various data sources and then operating the queries requested by a researcher on this encrypted data; while another party (i.e., key holder site) is in charge of keys made use of encryption and decryption. This key holder site decrypts the encrypted values obtained from data storage site to get the result of a count query and sends this result to the researcher. This protocol enables a strong privacy such that any malevolent user can never see the encrypted information of individuals unless two parties disagree.

Canim et al. [8] have proposed a protocol to securely store, share, and query clinical genomics data using secure cryptographic hardware. This protocol utilizes tamper-resistant cryptographic hardware to simplify secure genomic data storage and processing by removing the necessity of multiple parties. According to this study, all the encrypted records taken from multiple sources are kept in data storage site. This site can securely operate queries on this encrypted data by using the secure co-processor settled in the data storage site. The proposed method in [8] uses an encryption method that can only support count queries unlike this new solution which can directly promote any algorithm in which the intermediate outcomes could be stored in the memory of the co-processor.

Faramarzi et al. [9] have proposed a privacy preserving solution for the bipartite ranking problem. The RIMARC (Ranking Instances by Maximizing Area under the ROC curve) algorithm is used as a solution in this study. As a part of this model, the frequencies (i.e., count queries) are required to weight each feature by analyzing the area under the ROC curve. According to the study, each categorical value of each feature is encrypted by Paillier's cryptosystem and the frequencies are computed over this encrypted data.

Hasan et al. [10] have proposed a secure and efficient method for outsourcing genomic data. The proposed method builds an index tree from the different sources of genomic data and then outsources it to the third party cloud server. By using a secure protocol, the cloud server can traverse the nodes of this index tree and answer the count queries. Besides, Bloom filter [11] is added to each node of this index tree. The underlying idea is to utilize a structure similar to Bloomfilter search tree [12], which eases searching process over the index tree.

2.2. Classification with differential privacy

Data mining is the process of discovering the useful information from the data. Privacy preserving data mining is an important research area in data mining. The goal of the privacy preserving data mining is to ensure the privacy of individuals while enabling to perform data mining techniques. Many privacy preserving techniques such as privacy preserving association rule mining, privacy preserving clustering [13, 14, 15, 16], privacy preserving classification relying on a number of data mining algorithms such as SVM, k-NN etc. [17, 18] have been studied. However, differential privacy has recently been proposed method to guarantee strong privacy and it has been used for privacy preserving classification. Therefore, differential privacy has been implemented with some data mining algorithms in the literature. A differentially private logistic regression algorithm has been proposed by [19]. In 2010, a differentially private version of ID3 tree, where the information gain is estimated with the utilization of noisy counts obtained by adding noise drawn from Laplace distribution, has been proposed [20]. After that, Jagannathan et al. has demonstrated that construction of such a differentially private ID3 tree with the usage of low-level queries cannot ensure both good privacy and accuracy meanwhile [21]. Hence, they have presented a private ensemble method attributed to random decision trees. They observed that this algorithm performs better than the differentially private ID3 tree in terms of accuracy values even for small datasets. In 2013, they have proposed a variant of the differentially private random tree ensemble in [22]. In this study, a semi-supervised method which modifies the random decision tree approach to be used with the unlabelled data has been performed. This hybrid technique increases the accuracy values of the previous study [21] without decreasing the privacy [22].

Vaidya et al. [23] performed differential privacy on Naïve Bayes classification algorithm. Fletcher and Islam [24] have developed a differentially private decision forest approach which employs Gini index to construct a decision tree. The proposed approach has been compared with differentially private ID3 of [20] and non-private random forest algorithms. It has been demonstrated that the proposed method has very close accuracy values to those of classical random forest algorithm [24]. At the same time, Bojarski et al. [25] have presented three variants of differentially private random decision trees with majority voting, threshold averaging, and probabilistic averaging mechanism to classify instances. Su et al. [26] has developed a differentially private k-means clustering algorithm. Gursoy et al. [27] has conducted a differentially private nearest neighbor classification method by using k-NN.

According to the literature, any private version of 1R classification algorithm does not exist and there are only a few studies to operate count queries (i.e., frequencies) over a database securely. However, the security protocol proposed by Kantarcioglu et al. [7] insures strong confidentiality thanks to two-party computation such that even if one of the parties disrupts, obtaining of original data will not be possible unless another party disrupts simultaneously as well. Therefore, in our proposed model, we adopted a security protocol which is similar to

the secure count protocol used for the computation of frequencies of DNA sequences in the study of Kantarcioglu et al. [7].

In our proposed method, squared Euclidean distances of the records to a given query are used in the data storage site before the aggregation of the decrypted outcomes unlike [7], and our data encryption scheme utilizes the categorical index values of the attributes while a mapping is used for the encryption of nucleotides in [7]. Moreover, differential privacy is applied to the actual result of count queries (i.e., frequencies) in private key holder site as well. The other difference of our proposed security protocol from the method used in [7] is that, in our study the count queries are received as batch from the researcher, who builds 1R classifier, and then these queries are permuted in data storage site, and count queries are run in this changed order. Thus, a malicious user will not distinguish which result belongs to which query, in other words, actual result of a query cannot be obtained even if he/she disrupts any of the parties. We use Paillier's cryptosystem similarly to other existing studies in the literature, which operates count queries securely.

3. Background

3.1. Paillier's homomorphic encryption

Paillier's Encryption [3] is a partially homomorphic encryption layout where a single type of computation is possible unlike fully homomorphic encryption where different types of computation are allowed. It is simple and applicable to the real world applications (e.g. secure e-voting). Paillier's cryptosystem provides the addition as a computation in ciphertext domain as defined in Equation (1).

$$E(m_1 + m_2) = E(m_1).E(m_2) \quad (1)$$

where $E(m_1)$ and $E(m_2)$ are encrypted plaintexts (i.e., ciphertexts) using the same public key p_k. Additionally, a ciphertext $E(c.m)$ can be computed as $E(m)^c$ with Paillier's additive homomorphic property. Furthermore, the encryption of a plaintext and the decryption of a ciphertext in Paillier's cryptosystem can be described as follows:

Let p and q are large prime numbers with the same bit length, public key p_k is set to $n = p.q$, and private key p_r is (γ, n), and γ be the lowest common multiplier of $(p-1)$ and $(q-1)$. Given n, the plaintext m, and a random number r between 1 to $n-1$; the encryption of a plaintext m is equal to $E(m) = (n+1)^m r^n \bmod n^2$. On the other hand, given n and the ciphertext $c = E(m)$, the decryption of c can be computed as follows:

$$D(c) = m = [((c^{\gamma} mod\ n^2) - 1)/n].\gamma^{-1}\ mod\ n \quad (2)$$

where γ^{-1} is the inverse of modulo n.

3.2. Differential privacy

Differential privacy [4], which is a strong privacy guarantee, has been proposed to perform data mining algorithms over databases which contain sensitive information. It determines privacy leakage ratio by an ϵ parameter, and enables individuals' data to be taken safely in a database [4, 5, 6]. Differential privacy asserts that the output of a function does not entirely depend on any instance in the database. It claims that yielding of the same output is highly probable even if an instance is or not in the database.

Definition 1 (*Neighbor databases*) D and D' are two neighbor databases which differ from each other with a single instance, $|D' \Delta D|=1$.

Definition 2 (*ϵ-differential privacy*) A randomized mechanism A (such as Laplace mechanism) is ϵ-differentially private if all subsets S of the outputs of the algorithm A for all neighbor databases D' and D satisfy the following condition: And

$$S \subseteq \text{Range(A)}\ \ AND\ \ \Pr[A(D) \in S] \leq e^{\epsilon} \times \Pr[A(D') \in S] \quad (3)$$

where $\Pr[A(D) \in S]$ is the probability of $A(D)$ of being an element of S, $A(D)$ and $A(D')$ are the outputs of the randomness algorithm A for the databases D and D' respectively, and ϵ is used to check out how much a malicious client can recognize the difference between the databases D' and D, and $Range(A)$ represents the range of the outputs which can be generated by randomized mechanism A.

Definition 3 (*Sensitivity*) Let $f(D): D \to \mathbb{R}$ be a function mapping a database D into real numbers. The sensitivity for $f(D)$ is determined by

$$\Delta f := max_{D,D'} || f(D) - f(D') || \quad (4)$$

where $\Delta f := 1$, $||.||$ is the L_1 norm and the sensitivity is equal to 1 for all neighbor databases D and D'. The sensitivity of a function f represents the maximum magnitude in which the record of only one individual can alter the value of f for the worst case. In other words, the sensitivity for a function grants a maximum bound on how much its output must be perturbed to provide differential privacy [4, 5, 6].

Definition 4 (*Laplace mechanism*) Let $Lap(\gamma)$ be the Laplace distribution by mean 0 and standard deviation γ. For the function $f(D): D \to \mathbb{R}$, the randomized algorithm A represents Laplace mechanism and responds $f(D)$ as follows:

$$A(f(D)) = f(D) + V \quad (5)$$

where V is an independent and identically distributed

random variable drawn from $Lap(\gamma)$, and provided that $\gamma \geq \Delta f/\epsilon$, then the algorithm A is ϵ-differentially private. Therefore, Laplace mechanism is ϵ-differentially private [4, 5].

In this study, we need only frequency queries (i.e., count queries) of each predictor for each class to build 1R classifier. Considering the definitions of differential privacy given above, a count query can be represented as a function. If its actual query result is δ, the differentially private result (i.e., noisy result) is $\delta + b$, where b is drawn from Laplace distribution with mean 0 and standard deviation $\frac{\Delta f}{\epsilon}$, such that Δf is the sensitivity of count query and is equal to 1, and ϵ is the privacy parameter of which smaller values mean much more privacy.

3.3. 1R classification algorithm

1R [2] is a simple and efficient rule-based classifier, which finds the most informative attribute in the data and classifies instances with the values of this single attribute. As it uses only one attribute for classification task, it is called "One Rule". It generates very few rules that are very easy to interpret for humans. The pseudo-code of building 1R classifier is given in Algorithm 1.

Algorithm 1. 1R classification algorithm

Input: Database *D*
Output: The IF-THEN rules of 1R classifier
Begin
  for each attribute A_j in *D* **do**
    for each attribute value v_i **in** A_j
      Count how often v_i appears in each class, and set this value to n_{ji}
    end for
    Detect the most frequent class of v_i by using n_{ji} values
    Make an IF-THEN rule with consequent as the most frequent class label and the antecedent as $A_j = v_i$
    Calculate the total classification error of the rules of A_j
  end for
  Choose the best attribute A_{best} of which IF-THEN rules that have the smallest total error among all A_j
  return The IF-THEN rules of A_{best};
end

4. The proposed privacy preserving classifier

In this paper, we propose a privacy preserving classification approach that employs homomorphic encryption and differential privacy while building the 1R classifier. Our proposed privacy preserving classification model operates as the scheme given in Fig. 1. This approach considers two parties for security requirements such that it is impossible to see the data and to have the actual query result unless both of these two parties do not cooperate. Consequently, there exists more than one point for data security thanks to the proposed privacy model and in this way, if a malicious user seizes one of the parties, the user cannot get any information about the individuals in the encrypted database and the actual frequency value (i.e., count query) requested by the researcher.

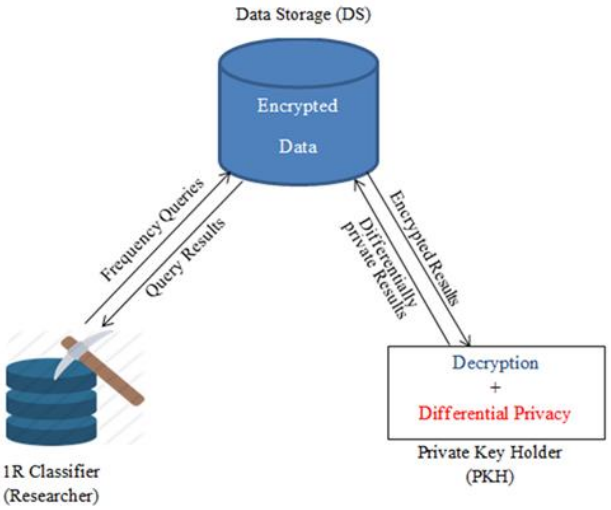

Fig. 1. Scheme of the proposed privacy model to build 1R classifier.

According to Fig. 1, encrypted versions of the database records obtained by Paillier's cryptosystem are held in a party which is called Data Storage (DS). The storage fulfills the frequency queries requested by the researcher, who builds 1R classifier, without decrypting any records in the data. Query results are dispatched to another party (i.e., Private Key Holder (PKH)) that decrypts and accumulates the results (i.e., How many records meet the condition given by the requested query). After that, this party applies differential privacy to the actual query result and returns the noisy result to DS again. Then, DS returns the differentially private frequency query result to the researcher (1R classifier).

4.1. Security structure

In this section, we describe how cryptographic properties are used to query the encrypted training data for the classification task. According to, Fig.1, the following steps form the security infrastructure of our proposed method.

Step 1 (Key generation): For the first step of the proposed protocol, PKH sends a public key to data owner.

Step 2 (Data encryption): When data owner demands to locate the training data to the DS, then DS sends the public key to the data owner. After that, the data owner encrypts the training data with the public key and dispatches this encrypted training data to DS. It is DS in which the encrypted training data is kept securely, and the queries requested by the researcher who builds 1R classifier are answered. Note that, we suppose only authorized data owner can locate her/his data to DS for the construction of 1R classifier on top of available

authentication and access mechanisms.

Step 3 (Query submission and processing): Queries are submitted by the researcher in a batch form, and they are shuffled before processed by the PKH. The order of the queries are changed by the DS. For a query, DS computes the squared Euclidean distances between the query and each encrypted record from the training data and dispatches these intermediate encrypted results to PKH. Then, PKH decrypts these intermediate results and obtains the aggregated result of the query. After obtaining the final result, PKH injects some Laplace noise with respect to an ϵ parameter to provide differential privacy guarantee, and then sends this perturbed count query result to DS. Finally, after all queries are operated by DS, the perturbed query results are sent by DS to the researcher in the order that the researcher requests.

Since training data for classification task kept in DS is semantically secure, the DS will know the content of the encrypted training data only if it possesses the private key. However, the DS does not get the private key since PKH keeps it private. The PKH only publishes DS the public key and the training data kept in DS is naturally private against DS and the researcher. However, the count queries are taken from the researcher, who builds the 1R classifier, as batch form; and these queries are permuted among themselves in DS, and the queries are run in this permuted order. Thus, a malevolent user cannot distinguish which result belongs to which query even if a hacker breaks one of the security parties. At the same time, the query results are perturbed by adding some Laplace noise, and its' amount is determined by the differential privacy mechanism. Thus, the actual query results of the queries are not disclosed as long as DS and PKH collaborate.

4.2. Data encryption

In our approach, the training dataset for classification is settled in a database. Our classification datasets consist of only categorical attributes. Table 1 shows how the categorical attribute values of a feature are encrypted.

Table 1. Encryption of the categorical values of attribute "age"

Categorical attribute value of feature "age"		
Original attribute values	Encrypted attribute values	Encrypted square of attribute values
young	E(0)	$E(0^2)$
middle	E(1)	$E(1^2)$
old	E(2)	$E(2^2)$

According to Table 1, we assume that we have a feature "age" in the training dataset and this feature has 3 values that are "young", "middle", and "old". We keep the categorical index values of this feature that are 0, 1, and 2 in the database and encrypt these values as given in the second column of Table 1. All feature values including class labels in the training dataset are encrypted with the public key of Paillier's cryptosystem in the same way. In our encrypted database, we also keep the square of the attribute values in the encrypted form in the data storage, since squared Euclidean distance is used for the computation of how many records are provided by a given query.

4.3. Computation of distance between a query and an encrypted database record

In our proposed method, square of the Euclidean distance between the attributes values in the query and the encrypted attribute values in a database record is utilized to calculate the frequency (i.e., count query) by taking advantage of additive homomorphic property of Paillier's cryptosystem. In this study, we measure the total distance between the categorical attribute values of related features in the query and an encrypted database record by using square of the Euclidean distance. The square Euclidean distance is given in Equation (6).

$$d^2 = \sum_{i=1}^{m}(r_i - q_i)^2 \tag{6}$$

where d is the Euclidean distance, r_i is the categorical attribute value of feature i in a training instance (i.e., record), q_i is the categorical attribute value of feature i in the query, and m is the number of features in the query. On the other hand, the encrypted version of d^2 is computed as in Equation (7).

$$E(d^2) = \prod_{i=1}^{m} E(r_i^2).E(q_i^2).E(-2r_i.q_i) \tag{7}$$

where $E(d^2)$, $E(r_i^2)$, $E(q_i^2)$, and $E(-2r_i.q_i)$ are the encrypted versions of d^2, r_i, q_i, and $-2r_i.q_i$ respectively. In Equation (7), $E(-2r_i.q_i) = E(r_i)^{-2q_i}$ since $E(m.a \bmod n)$ is equal to $E(m)^a \bmod n^2$. To compute Equation (7), DS encrypts the squares of the categorical attribute values in the query with public key of Paillier's cryptosystem. This encryption is performed as described in Data Encryption Section above. For instance, considering the "age" example in Table1 we assume that a count query q_1 sent by the researcher to DS is *Select Count (*) from Table where age="old".* In this case, the encrypted versions of the q_1 and q_1^2 are equal to $E(2)$ and $E(2^2)$ respectively. For the simplicity and limited space in this paper, we illustrate the encryption of a query for a single attribute. However, the encryption of all categorical attribute values including class attribute values in the query are discharged in similar way.

4.4. The proposed securely count query

Frequencies (i.e., count queries) of each categorical attribute values for each class are required to build 1R classifier. However, count queries were not performed over homomorphically encrypted data. Therefore, we propose a protocol which securely computes these

frequencies without decrypting the encrypted data. Our security protocol is based on the secure count protocol used for the computation of frequencies of DNA sequences in the studies of Kantarcioglu et al. [7]. However Euclidean distance is used in DS side and our data encryption scheme utilizes the categorical index values of the attributes while a mapping is used for the encryption of nucleotides in [7]. Moreover, differential privacy is applied to the frequency values in PKH side. In our DS protocol, the squared Euclidean distance of each encrypted record to the count query requested by the researcher is computed. On the other hand, PKH computes the result of the query by decrypting these distances calculated by DS and Laplace noise is added to this actual query result to provide differential privacy in our PKH protocol. At the same time, the protocols for DS and PKH are given in Protocol 1 and 2 respectively.

Protocol 1. DS-Computation of distance between a query and an encrypted database record.

Input: A count query q, encrypted database D
  for each record r in D **do**
    $d_r \leftarrow \prod_{i=1}^{m} E(r_i^2).\ E(q_i^2).E(r_i)^{-2q_i}$
  end
  $d \leftarrow \pi(d_1,\ d_2, \dots, d_n)$, where π is a random permutation, n is the # of records in D
  Send d to PKH.

Protocol 2. PKH-Differentially private query result

Input: d from DS, Differential privacy parameter ϵ
  $\epsilon' := \frac{\epsilon}{class_{number} \times \sum_{j=1}^{n} \sum_i 1}$
  $\gamma := \Delta f / \epsilon'$
  count$\leftarrow 0$;
  for each distance d_i in d **do**
    if $Decrypt(d_i) == 0$ **then**
      count$\leftarrow count + 1$;
    end if
  end for
  count$\leftarrow count + Lap(0, \gamma)$;
  Send $count$ to DS.

In the first protocol, Euclidean distances between each record in the database and a given count query are calculated and these distances are sent to PKH after applying a random permutation π to be processed by the second protocol. In the second protocol, these encrypted distances are decrypted, and checked whether they are equal to 0 or not. If the Euclidean distance is equal to 0, count is incremented by 1 which means the query matches with the encrypted data record. After all distances are decrypted and checked, we obtain the result of the given count query (i.e., frequency). Following this, Laplace noise is added to the result of the count query to provide differential privacy guarantee. Moreover, the count queries dispatched by the researcher to DS are sent to PKH with a random permutation as well. Thus, a malicious user will not distinguish which result belongs to which query even if he/she disrupts the second party (i.e., PKH).

In Protocol 2, the number of count queries required to build an 1R classifier is equal to $class_{number} \times \sum_{j=1}^{n} \sum_i 1$, where n represents the number of attributes in the training dataset, j is the j^{th} attribute (i.e., predictor) of this dataset, and i is the i^{th} value of the attribute j. On the other hand, sensitivity of a count query, Δf, is equal to 1. ϵ is the total budget to guarantee differential privacy. ϵ' is the budget per each count query and is equal to $\frac{\epsilon}{class_{number} \times \sum_{j=1}^{n} \sum_i 1}$. $Lap(0, \gamma)$ represents the noise drawn from Laplace distribution with mean 0 and standard deviation γ where $\gamma = \frac{\Delta f}{\epsilon'}$.

5. Experimental results

In the experiments, we use 4 UCI datasets that are Congressional votes, Mushroom, Nursery, and Spect-h. According to Table 2, number of classes for the datasets changes from 2 to 5, and number of attributes ranges from 8 to 23.

Table 2. Description of the datasets

Dataset	# of Attributes	# of Classes	# of Instances
Cong. votes	16	2	435
Mushroom	22	2	8124
Nursery	8	5	12960
Spect-h	23	2	267

Table 3. Average classification accuracies of private 1R

Dataset	Epsilon Value (ϵ)						
	∞	3	2	1	0.5	0.25	0.1
Cong.	0.956	0.886	0.800	0.739	0.625	0.558	0.570
Mush.	0.950	0.968	0.946	0.862	0.760	0.641	0.560
Nursery	0.709	0.709	0.709	0.707	0.691	0.545	0.347
Spect-h	0.723	0.571	0.564	0.526	0.510	0.504	0.515

Table 4. Average classification accuracies of private Naive Bayes

Dataset	Epsilon Value (ϵ)						
	∞	3	2	1	0.5	0.25	0.1
Cong.	0.901	0.893	0.886	0.866	0.799	0.701	0.603
Mush.	0.957	0.929	0.926	0.911	0.873	0.803	0.688
Nursery	0.902	0.895	0.886	0.854	0.740	0.549	0.386
Spect-h	0.681	0.619	0.605	0.568	0.512	0.515	0.484

In the experimental results, we present the performance of the proposed private 1R classifier for the different values of privacy parameter ϵ which is applied in the second party (i.e., PKH). According to these results, the lower the values of ϵ parameter are, the lower classification accuracies are observed but the more privacy is provided; while the higher the values of ϵ parameter are, the higher classification accuracies are obtained but having less privacy as in other differentially private classifiers in the literature [20, 21, 22, 23, 24, 25, 27].

We analyze the classification performance of the proposed private 1R classification algorithm for the various ϵ parameter values that are 0.1, 0.25, 0.5, 1, 2, and 3. We run the classifier 100 times. For each time, 90% of the whole dataset is determined as training dataset and the encrypted version of this training data is settled in the first party (i.e., DS). 10% of the dataset is used to test the private 1R algorithm. We give the average classification accuracy values for each ϵ parameter value at the end of 100 runs in Table 3. At the same time, we also use our privacy model to perform privacy preserving Naïve Bayes classification algorithm since Naïve Bayes classification algorithm is a baseline classifier in the literature [23, 27] to utilize differential privacy, and it also requires frequencies of attribute values (i.e., count queries) to build the classifier. The average classification accuracies of Naïve Bayes are presented in Table 4 as well.

According to Table 3 and Table 4, $\epsilon = \infty$ means differentially privacy is not applied in PKH. When $\epsilon=\infty$, 1R outperforms Naïve Bayes for the datasets Congressional votes and Spect-heart. On the other hand, Naïve Bayes achieves 0.957 average accuracy values for the dataset Mushroom and 1R reaches slightly lower accuracies with 0.950 for this dataset. When $\epsilon=3$, 1R performs better than Naïve Bayes for the dataset Mushroom. 0.893 accuracy value is obtained by Naïve Bayes while 0.886 is attained by 1R for the dataset Congressional votes. The difference between these accuracy values is only 0.007 which is quite low. When $\epsilon = 2$, 1R outperforms Naïve Bayes for the dataset Mushroom. For the dataset Spect-h, 0.605 accuracy value is observed for Naïve Bayes while 0.564 accuracy value is observed for 1R which is only 0.041 lower than that of Naïve Bayes. On the other hand, 1R achieves 0.800 accuracy while Naïve Bayes reaches 0.886 accuracy for the dataset Congressional votes. When $\epsilon<2$, Naïve Bayes performs better than 1R classifier in general. But, the differences between the accuracy values of Naïve Bayes and 1R are quite close to each other for the datasets Mushroom and Spect-h. However, Naïve Bayes outperforms 1R classifier for the dataset Congressional votes when $\epsilon \leq 1$. When examined the classification performances of 1R and Naïve Bayes over the dataset Nursery, we infer from the tables that Naïve Bayes is more successful than 1R for all values of ϵ. At the same time, Naïve Bayes attains 0.740, 0.549, and 0.386 accuracy values for $\epsilon<1$ while 1R figures out at 0.691, 0.545, and 0.347 accuracy values. The differences between these accuracy values are only 0.049, 0.004, and 0.039.

When a general classification performance assessment is made for privacy preserving 1R and Naïve Bayes algorithms, it can be inferred that 1R and Naïve Bayes algorithms show similar performances to each other when $\epsilon \geq 2$ over the datasets that are Congressional votes, Mushroom, and Spect-heart. However, Naïve Bayes is superior to 1R for these datasets when $\epsilon < 2$. Besides, Naïve Bayes performs better than 1R for all values of ϵ over the dataset Nursery. But, the average classification accuracies of Naïve Bayes are slightly higher than those of 1R when $\epsilon<1$.

5.1. Security analysis

In this section, we investigate the security of data when our proposed method is applied in terms of secure multi-party computation. In our proposed protocol, the result of a query requested by the researcher, which is sent from DS to PKH, includes encryptions of either 0's or other values in ciphertext domain. Consequently, it can be demonstrated that only PKH knows the query result (i.e., # of encrypted 0's). However, something else about encrypted data kept in DS are not learned by PKH.

Considering the proposed privacy protocol, two parties (i.e., DS and PKH) are utilized to compute the result of a count query. In the first party (i.e., DS), Euclidean distance of each record to the given query is computed over the encrypted training data, and then these distances are permuted such that $d \leftarrow \pi(d_1, d_2, \dots, d_n)$ where π is a random permutation, are sent to the second party for the decryption. Let $d \leftarrow \pi(d_1, d_2, \dots, d_n)$ and $d' \leftarrow \pi(d_1, d_2, \dots, d_n)$ be two permutated vectors in ciphertext domain. According to polynomial-time sampling theorem, d and d' are computationally indistinguishable [7, 28]. Therefore, the researcher asks for queries in the batch form and a random permutation, π, is also used to compute the results of these requested queries in our privacy protocol. Thus, we prevent if a malicious user disrupts any of the parties, he/she cannot differentiate which result belongs to which query.

5.2. Computational complexity and storage requirement

The storage requirement of our proposed scheme is 8 bytes for each feature value in the original form of data, and 128 bytes for each feature value in the encrypted form of data, since the bit length is determined as 512 bits for encryption. As for run-time analysis of the proposed privacy approach, it is clear that the time complexity of the computation of Euclidean distances between records

and a given query under encryption depends on the number of records in the training data and the number of features in the query, and it can be given as $O(m.n)$ in Big O notation where m represents the # of features in the query and n is the # of records in the data.

6. Conclusion

In this study, we propose a privacy preserving protocol to build Holte's 1R classifier. In our proposed protocol, we perform the combination of Paillier's cryptosystem and differential privacy that are the two most popular confidentiality techniques. Our protocol consists of two parties which are called DS and PKH to provide more strong privacy guarantee. In DS, we keep the encrypted training data and a Euclidean distance based scheme is proposed to compute the frequencies of attribute values over this homomorphically encrypted data to build 1R classifier, while the actual results of these frequency values are obtained and differential privacy is applied to these results in PKH to minimize the privacy leakage as much as possible. According to this study, count queries are performed without decrypting the training data for the classification and having the actual results of these count queries are quite complex thanks to random permutation applied during the communication of two parties (i.e., DS and PKH) and differential privacy in PKH. To compare the classification performance of our proposed 1R classifier, the proposed privacy preserving model is also applied to Naïve Bayes classification algorithm which is a baseline technique used for the performance comparisons of the differentially private algorithms in the literature. According to the experimental results, the proposed method can be efficiently used for privacy preserving classification.

Author's Note

Abstract version of this paper was presented at 9th International Conference on Advanced Technologies (ICAT'20), 10-12 August 2020, Istanbul, Turkey with the title of "A Hybrid Approach of Homomorphic Encryption and Differential Privacy For Privacy Preserving Classification".

References

[1] Vaghashia H. and Ganatra A., 2015. A survey: Privacy preservation techniques in data mining. International Journal of Computer Applications., vol. 119, no. 4, pp. 20-26.

[2] Holte R. C., 1993. Very simple classification rules perform well on most commonly used datasets. Machine Learning., vol. 11, pp. 63-90.

[3] Paillier P., 1999. Public key cryptosytems based on composite degree residosity classes. In Advances in Cryptology-Proceedings Eurocrypt '99. (Lecture Notes in Computer Science, no. 1592). New York: Springer-Verlag, 1999, pp.223-238.

[4] Dwork C., McSherry F., Nissim K., and Smith A., 2006. Calibrating noise to sensitivity in private data analysis. In Theory of Cryptography., pp. 265-284, Springer.

[5] Dwork C., 2008. Differential privacy: A survey of results. In Proc. 5th International Conference on Theory and Applications of Models of Computation, Xi'an, China.

[6] Dwork C. and Roth A., 2014. The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science, Vol. 9, Nos. 3-4 pp. 211-407.

[7] Kantarcioglu M., Jiang W., and Malin B., 2008. A cryptographic approach to securely share and query genomic sequences. IEEE Transactions on Information Technology in Biomedicine, Vol. 12, No. 5.

[8] Canim, M., Kantarcioglu, M., and Malin, B. (2011). Secure management of biomedical data with cryptographic hardware. *IEEE Transactions on Information Technology in Biomedicine*, *16*(1), 166-175.

[9] Faramarzi, N. S., Ayday, E., & Guvenir, H. A. (2016, December). A Privacy-Preserving Solution for the Bipartite Ranking Problem. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 375-380). IEEE.

[10] Hasan, M. Z., Mahdi, M. S. R., Sadat, M. N., and Mohammed, N. (2018). Secure count query on encrypted genomic data. *Journal of biomedical informatics*, *81*, 41-52.

[11] Bloom, B. H. (1970). Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, *13*(7), 422-426.

[12] Pappas, V., Krell, F., Vo, B., Kolesnikov, V., Malkin, T., Choi, S. G., ... & Bellovin, S. (2014, May). Blind seer: A scalable private dbms. In *2014 IEEE Symposium on Security and Privacy* (pp. 359-374). IEEE.

[13] Vijarayani S., and Prabha M. S., 2011. Association rule hiding using artificial bee colony algorithm. International Journal of Computer Applications, 33(2), pp. 41-47.

[14] Preethi P., K., Kumar P., Ullhaq M. R., Naveen A., and Janapana H., 2018. Privacy preserving data clustering using a heteregeneous data distortion. Smart Intelligent Computing and Applications, pp. 477-486.

[15] Inan A., Kaya S. V., Saygın Y., Savaş E., Hintoğlu A. A., and Levi A., 2007. Privacy preserving clustering on horizontally partioned data. Data and Knowledge Engineering, 63(3), pp.646-666.

[16] Hyma, J., Varma, P. S., Gupta, S. N. K., & Salini, R. (2019). Heterogeneous Data Distortion for Privacy-Preserving SVM Classification. In *Smart Intelligent Computing and Applications* (pp. 459-468). Springer, Singapore.

[17] Kantarcioglu, M., & Clifton, C. (2004, September). Privately computing a distributed k-nn classifier. In *European conference on principles of data mining and knowledge discovery* (pp. 279-290). Springer, Berlin, Heidelberg.

[18] Rubinstein B. I. P., Bartlett P. L., Huang L., and Taft N., 2009. Learning in a large function space: Privacy preserving mechanisms for SVM learning. Computing Research Repository.

[19] Chaudhuri, K., & Monteleoni, C. (2009). Privacy-preserving logistic regression. In *Advances in neural information processing systems* (pp. 289-296).

[20] Friedman A., and Schuster A., 2010. Data mining with differential privacy. In Proc. 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA.

[21] Jagannathan G., Pillaipakkamnatt K., and Wright R. N., 2012. A practical differentially private random decision tree classifier. Transactions on Data Privacy., no. 5, pp. 273-295.

[22] Jagannathan G., Monteleoni C., and Pillaipakkamnatt K., 2013. A semi-supervised learning approach to differential privacy. In Proc. 13th International Conference on Data Mining Workshops, TX, USA.

[23] Vaidya J., Shafiq B., Basu A., and Hong Y., 2013. Differentially private naïve bayes classification. In Proc. IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technologies, pp. 571–576.

[24] Fletcher S., and Islam M. Z., 2015. A differentially private decision forest. In Proc. 13th Australasian Data Mining Conference, Sydney, Australia.

[25] Bojarski M., Choromanska A., and Choromanski K., 2015. Differentially-and non-differentially private random decision trees. arXiv preprint arXiv:1410.6973v2.

[26] Su, D., Cao, J., Li, N., Bertino, E., & Jin, H. (2016, March). Differentially private k-means clustering. In *Proceedings of*

the sixth ACM conference on data and application security and privacy (pp. 26-37).

[27] Gursoy M. E., Inan A., Nergiz M. E., and Saygın Y., 2017. Differentially private nearest neighbor classification. Data Mining and Knowledge Discovery, vol. 31, no. 5, pp. 1544-1575.

[28] Goldreich O., 2004. General cryptographic protocols. The Foundations of Cryptography. Cambridge, U.K.:Cambridge Univ. Press.