Secure Outsourced *k*NN Data Classification Over Encrypted Data Using Secure Chain Distance Matrices

Nawal Almutairi[1,2], Frans Coenen[1], and Keith Dures[1]

[1] Department of Computer Science, University of Liverpool, Liverpool, UK
{n.m.almutairi,coenen,dures}@liverpool.ac.uk
[2] Information Technology Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia
nawalmutairi@ksu.edu.sa

Abstract. The paper introduces the Secure kNN (SkNN) approach to data classification and querying. The approach is founded on the concept of Secure Chain Distance Matrices (SCDMs) whereby the classification and querying is entirely delegated to a third party data miner without sharing either the original dataset or individual queries. Privacy is maintained using two property preserving encryption schemes, a homomorphic encryption scheme and bespoke order preserving encryption scheme. The proposed solution provides advantages of: (i) preserving the data privacy of the parties involved, (ii) preserving the confidentiality of the data owner encryption key, (iii) hiding the query resolution process and (iv) providing for scalability with respect to alternative data mining algorithms and alternative collaborative data mining scenarios. The results indicate that the proposed solution is both efficient and effective whilst at the same time being secure against potential attack.

Keywords: Secure kNN query, Homomorphic encryption, Secure Chain Distance Matrices, Order preserving encryption.

1 Introduction

Recent years have witnessed an increase in the adoption of cloud services to store and manage data. There has been an increasing tendency for Data Owners (DOs), enterprises of all kinds, to outsource their data storage to Cloud Service Providers (CSPs) according to some contractual agreement. However, there are increasing concerns that sensitive data, belonging to the DOs, may be inadvertently exposed or misused [30]. These concerns are compounded by legislative requirements for data privacy preservation [6, 11]. This has motivated DOs to encrypt their data prior to outsourcing to CSPs so that the privacy of sensitive information is guaranteed [24].

Although encryption addresses the above data confidentiality issue it imposes limitations on the functionality of the operations that can be applied to the data in that the data can only be processed (queried) by the DOs who are in possession

of the encryption keys. There is also an increasing desire, on behalf of DOs, for the benefits of data mining and machine learning to be leveraged from their data. Many CSPs provide a Data Mining as a Service (DMaaS) [5] facility. However, the standard encryption techniques used to preserve data confidentiality means that the application of any data mining task will necessitate some form of data decryption. The research domain of Privacy Preserving Data Mining (PPDM) seeks to address this issue [1, 15].

A variety of PPDM methods have been proposed, including: data anonymisation [26], perturbation [19, 33] and the utilisation of Secure Multi-Party Computation (SMPC) protocols [10]. Using data anonymisation, DOs will remove "personal" attributes that are deemed confidential from the data and then irreversibly generalised the remaining dataset according to some "syntactic" condition. However, examples of breaches data confidentiality, reported in [22, 28, 29], have shown that anonymised data can be "de-anonymised" using quasi-identifier attributes and "linkage attacks" [22]. Data perturbation (or transformation) operates by distorting or randomising the entire dataset by adding noise while maintaining the statistical makeup of the data. However, perturbing the data cannot entirely assure data privacy since most of the methods used allow "reverse engineering" of the original data distribution [13]. Perturbation methods and data anonymisation have also been shown to be unsuitable for many instances of DMaaS; it has been demonstrated that they adversely affect the accuracy of the data analysis [19, 27]. The SMPC-based approach is directed at analysis tasks where the data is distributed, not encrypted, across a number of participating parties; such as a number of DOs, or a single DO and several Query Users (QUs). The SMPC-based approach requires many intermediate computations, using a dedicated SMPC protocol, performed over non-encrypted data and using DO and/or QU local resources, the statistical results of which are then shared. The significant computational and communication overhead that is a feature of the SMPC-based approach has rendered the approach to be infeasible for large datasets and complex data mining activities. Moreover, when using a SMPC-based approach, the involvement of many DOs and/or QUs poses a security risk given the presence of a non-honest party who may launch attacks such as "overlapping attacks" [18] and Chosen-Plaintext Attacks (CPAs) [34]. These PPDM methods do not therefore provide a solution to the desire of DOs to take advantage of the benefits offered by CSPs in a manner whereby data confidentiality can be guaranteed while at the same time allowing the techniques of data analytics to be applied to their data.

The emergence of Property Preserving Encryption (PPE) schemes, such as Homomorphic Encryption (HE) [17], Asymmetric Scalar Product Preserving Encryption (ASPE) [31] and Order Preserving Encryption (OPE) [16, 20], has provided a potential solution to the disadvantages associated with PPDM by permitting cyphertext manipulation without decryption. HE schemes allow simple mathematical operations, such as addition and multiplication, to be applied over encrypted data. ASPE schemes preserve scalar distances across cyphertexts. OPE schemes permit cyphertext comparison. However, although PPE schemes

go someway to providing a solution to secure DMaaS they do not provide a complete solution in that, given a particular data mining application, the mathematical operations that are required are currently not all provided by single PPE scheme. This limitation has been addressed in the literature by either: (i) recourse to data owners whenever unsupported operations are required or (ii) confiding the secret key to non-colluding parties using either a secret sharing techniques, as the case of [23], or using two-distinct CSPs as in the case of [25]. The former solution clearly introduces a computation and communication overhead which renders the approach unsuitable for many instances of DMaaS. In the case of the latter, the existence of two non-colluding parties is not always applicable while at the same time raising security concerns for many DOs as the secret key cannot be revoked even when a party is found to be untrustworthy. The solution presented in this paper is to use two complementary PPE schemes which collectively provide the necessary operations without compromising data confidentiality. More specifically, the proposed solution uses two PPEs: Liu's HE scheme [17] and bespoken Frequency and Distribution Hiding Order Preserving Encryption (FDH-OPE) scheme.

In the context of previous work directed at the use of PPE schemes, a popular DMaaS application, because of its simplicity and because it is used with respect to many application domains [25], is k Nearest Neighbour (kNN) classification/querying [7]. Given a query record q and a prelabeled dataset D held by a CSP, the standard kNN approach, where $k = 1$, operates by finding the class label for the most similar record in D to q, and assigning this label to q. Where $k > 1$, kNN operates by finding the "major" class label amongst k nearest records and assigning this to q. The challenges here is not just efficient data privacy preservation in the context the dataset D belonging to the DO, but also the efficient data privacy preservation associated with the query set Q (or sets $\{q_1, q_2, \dots\}$). The general view is that the query process should be controllable by the DO to whom the prelabeled dataset D belongs. This means that any QU cannot encrypt the records in their query set without first being "approved" by the DO. In many proposed solutions [12, 31, 32, 35, 38] the DO is required to either: disclose the encryption key (or at least part of it) to the QUs so as to allow them to encrypt Q, or disclose the key to a Third Party Data Miner (TPDM) which in turn means QUs have to disclose Q to the TPDM (the CSP). Both approaches entail a potential security risk, either because of the wide distribution of the encryption key across QUs or because of the requirement to treat the TPDM as a trusted party. Another challenge is in how to determine securely the data similarity between the records in Q and the records in D. To address the data similarity challenge various techniques have been proposed which rely either on HE schemes that provide only a partial solution and consequently entail recourse to data owners, or make use of SMPC primitives that required DO and QU participation and thus entail an undesired computation and communication overhead.

The work presented in this paper proposes the Secure kNN classification/ querying (SkNN) system. The idea is to encrypt the dataset D using Liu's HE

scheme [17] while at the same time recasting the dataset into a proxy format. More specifically as a Chain Distance Matrix (CDM), of the form first introduced in [3], which is then encrypted using a proposed FDH-OPE scheme to give a Secure CDM (SCDM). By allowing the two encryption schemes to work in tandem the disadvantages associated with earlier approaches reliant on a single encryption scheme are avoided, and hence SkNN can process queries without requiring data owner participation or recourse to SMPC protocols as in the case of earlier solutions. To ensure data confidentiality the encryption keys are held by the DO and never confided with the QUs or the TPDM. The QUs encrypt their query set Q using a proposed Secure Query Cyphering (SQC) protocol that preserves the privacy of the query record and the confidentiality of the DO's private key. The query process is controllable by the DO, although undertaken by the TPDM without involving the QUs or DO. The proposed SkNN system is fully described and evaluated in the remainder of this paper.

2 Previous Work

This section presents a review of previous work directed at secure kNN data classification and kNN querying. The existing work is directed at different kNN querying scenarios and different level of party involvement, however it can be categorised according to the data confidentiality preserving technique adopted: (i) cryptography [12, 31, 34, 36, 38, 39], (ii) data perturbation [32, 35] and (iii) SMPC protocols [9]. In most cases three categories of party are considered: (i) a Third Party Data Miner (TPDM); (ii) a Data Owner (DO) and (iii) one or more authorised Query Users (QUs) who are permitted to query the outsourced data so as to label their own query records (the set Q). In the remainder of this previous work section a number of previously proposed exemplar secure kNN data classification/querying techniques are discussed, each representing a particular approach in the context of the above categorisation.

In [12] the HE scheme presented in [8] was used to encrypt the DO's data. The encrypted dataset was then outsourced to authorised QUs along with the encryption key whilst the decryption key was sent to the TPDM. The secure kNN data classification was collaboratively conducted by the QUs and the TPDM, thus the query process was not controlled by the DO, therefore raising security concerns. Also the approach featured a considerable communication overhead as a result of interactions between the QUs and the TPDM while qeuries were being process; most of the computation was conducted using the QUs' local resources. A general principle of DMaaS is that the QU and/or DO should not need to be involved in the processing of a query once the query is launched, the mechanism presented in [12] does not support this principle. Wong et al. [31] proposed an Asymmetric Scalar Product Preserving Encryption (ASPE) scheme which used a random invertible matrix to encrypt the outsourced data. The APSE scheme supported scalar product operations over cyphertext which were used to calculate Euclidean distances between encrypted data records and encrypted query records. However, in this approach the QUs have access to

the DO's encryption and decryption keys, hence the DO's data privacy may not be preserved. A similar approach was presented in [38], but providing some limitation on the information concerning encryption keys provided to QUs.

The work presented in [39] addresses the risk of encryption key leakage from QUs; however, the QUs can still learn the partial sum of the numbers in the encryption key belonging to the DO using a legal query, the QUs can also launch uncontrolled queries (queries that are processed without DO approval). Yuan et al. [36] present a secure kNN ($k = 1$) query scheme to address the threat of untrusted QUs and/or TPDMs; however, the QUs directly submit private plain query records to the DO which means that query privacy is not preserved. More recently, Zhu et al. [37] demonstrated that the scheme presented in [36] cannot achieve their declared security, and that the encrypted dataset in [36] can be quickly compromised by untrusted QUs and/or TPDMs.

In [35] a transformation method is used to encode the DO data outsourced to the TPDM. However, as in the case in [31, 38], the QUs have access to the encryption and decryption keys, therefore they are assumed to be fully trusted. The trusted QUs encrypt their data records and send queries to the TPDM who conducts an approximate similarity search on the transformed data. The search results are then sent back to the QUs who decrypt the results and determine the label of their query records. The work in [32, 34] presents various schemes to securely support approximate kNN for a given query record. In [34], the secure kNN is executed by retrieving the approximated nearest records instead of finding the encrypted exact k-nearest neighbours that requires the QUs to be involved in a substantial amount of computation during the query processing step. The method presented in [34] considers the TPDM as a provider of storage space, no significant work is done by the TPDM. In [32], Random Space (RASP) data perturbation combined with order preserving features are used to preserve data privacy and allow secure kNN querying. Confiding the encryption and decryption key to QUs, or to the TPDM as in the case of [31, 32, 34, 35], significantly increase the risk of key leakage (it is also difficult to revoke a key distributed to QUs should they be deemed untrustworthy). Thus raising a significant security concern, as detailed in [34], whereby QUs can launch Chosen-Plaintext Attacks (CPAs). The QUs are assumed to be completely trusted QUs; this not only in limits the application scope of this approach, but also raises several practical problems. In general, the existing secure kNN query schemes where QUs can access the DO's encryption key are still far from being practical in many situations.

3 System Model

This section introduces the system model and design goals for the proposed Secure kNN classification/querying (SkNN) system. As in the case of earlier work on secure kNN the proposed system features three types of participant: a DO, a TPDM and several QUs as shown in Figure 1. The TPDM is assumed to have a large but bounded storage and computation capability, and provides outsourcing

storage and computation services, for example the TPDM might be a CSP. The DO has a large privet dataset D which consists of r records, $D = \{d_1, \ldots, d_r\}$. Each record d_i has $a+1$ attribute values; $d_i = \{d_{i,1}, \ldots, d_{i,a}, d_{i,a+1}\}$ where $d_{i,a+1}$ is the class label for data record d_i. The QUs are a set of authorised parties who want to classify their data records $Q = \{q_1, q_2, \ldots\}$. The DO encrypts D using Liu's HE scheme (presented later in Section 4) to arrive at D' and sends it to the TPDM so as to take advantage of storage resources and computational ability provided by TPDM as a service. Note that the class label (attribute value $a+1$) for each record in D is not encrypted. The DO also generates a Secure Chain Distance Matrix (SCDM) encrypted using the proposed FDH-OPE scheme that facilitates secure data similarity determination, this is presented in further detail in Sections 4 and 5.

The DO delegates the generation of a kNN classification model, using its encrypted outsourced data, to the TPDM, and allows QUs to take advantage of the developed model. To maintain privacy any query $q_i \in Q$ needs to be encrypted by the QU who owns the query, before it is submitted to the TPDM for processing. Clearly to allow q_i to be processed using the kNN model generated using the DO's encrypted data, q_i needs to be encrypted using the same encryption key (held by the DO). Query encryption is thus achieved using a proposed Secure Query Cyphering (SQC) protocol that preserves the privacy of the query record and the confidentiality of DO's private key. To determine the similarity between an encrypted query record q_i' and the encrypted kNN model requires q_i' to be processed in such a way that it is integrated with the SCDM, a process referred to as "binding". The secure binding process is presented in Section 6. To make ensure that the querying is controlled by the DO, the binding process requires two records, one generated by the QU ($BindRec_1$) and the other generated by the DO that handles query approval ($BindRec_2$). Once approved query processing is delegated entirely to the TPDM. At the end of which the QU will receive predicted class label for q_i (see Figure 1).

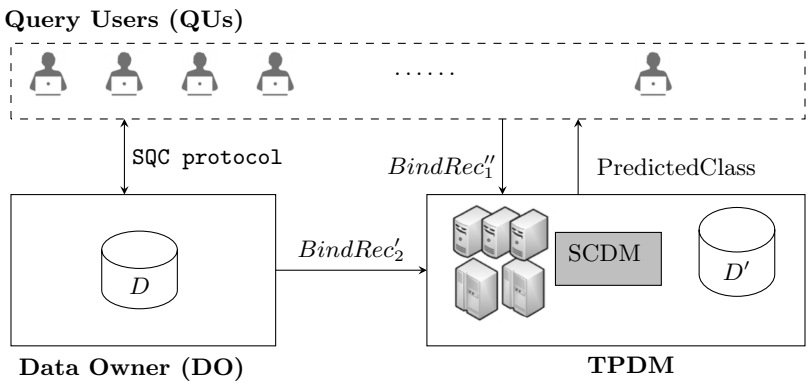


Fig. 1. The SkNN system architecture

4 Cryptographic Preliminaries

As noted above the proposed SkNN data classification and query process operates using two encryption schemes: (i) the FDH-OPE scheme used to encrypt SCDMs and (ii) Liu's HE scheme used to encrypt the DO's outsourced data and securely exchange the FDH-OPE keys using a dedicated SQC protocol. Both are discussed in further detail in the following two sub-sections, Sub-sections 4.1 and 4.2 respectively.

4.1 Frequency and Distribution Hiding Order Preserving Encryption (FDH-OPE)

This sub-section presents the FDH-OPE scheme used to encrypt CDMs, an order preserving scheme. The proposed scheme is an amalgamation of two existing Order Preserving Encryption (OPE) schemes, that of [20] and [16]. The former used to hide the data distribution in generated cyphertexts, the latter used to hide the data frequency. Encrypting data so that the data distribution is hidden requires knowledge of the distribution within the plaintext data, the plaintext intervals where the data density is high, and then generating the cyphertexts in such a way that high density plaintext intervals are dispersed along large cyphertext intervals. The frequency of data is simply hidden by generating different cyphers for the same plaintext value (even when using the same encryption key). The first step in FDH-OPE, is to determine the "interval" of the message space $M = [l, h)$ and the expanded "interval" of the cypher space $C = [l', h')$ in such a way that $M \ll C$ and the l, l', and h, h', are the minimum and maximum interval boundaries for the message and cypher spaces respectively (see Figure 2). Data distribution hiding comprises two steps, *message space splitting* and *non-linear cypher space expansion* which operate as follows:

Message space splitting: The DO randomly splits the message space interval M into t consecutive intervals; $M = \{m_1, \ldots, m_t\}$, where t is a random number. The length of intervals are determined randomly by deciding the minimum and maximum interval boundaries (Figure 2). The data density for each interval is then calculated as $Dens = \{dens_1, \ldots, dens_t\}$ where $dens_i$ is density of data in message space m_i.

Non-linear cypher space expansion: The DO then splits the cypher space C into t intervals; $C = \{c_1, \ldots, c_t\}$. So that the data distribution is hidden, the length of each cypher space interval c_i is determined according to the density of the data in the corresponding message space interval, $dens_i$, so that message space intervals with high data density will have large corresponding cypher space intervals. For example, if $dens_i > dens_j$ then $|c_i| > |c_j|$. The message space and cypher space interval boundaries are the FDH-OPE encryption keys.

The data frequency is hidden using a "one-to-many" encryption function that maps $x \in m_i$ to an OPE equivalent value $x' \in c_i$. Algorithm 1 gives the pseudo

Algorithm 1 FDH-OPE encryption algorithm

1: **procedure** $\text{ENC}_i(x, Sens)$
2: $i \leftarrow \text{IntervalID}(x)$
3: $[l_i, h_i] \leftarrow \text{Range}(i)$
4: $[l'_i, h'_i] \leftarrow \text{Range}'(i)$
5: $scale_i = \frac{(l'_i - h'_i)}{(l_i - h_i)}$
6: $\delta_i = \text{Random}(0, Sens \times scale_i)$
7: $x' = l'_i + scale_i \times (x - l_i) + \delta_i$
8: **Exit** with $\mathbf{x}'$
9: **end procedure**

code for the encrypting function. The algorithm commences by determining the message space interval ID, i, within which x is contained (line 2). The interval boundaries (keys) of the ith message and cypher space are then retrieved in lines 3 and 4. These values are used to calculate interval $scale_i$ and sample random value δ_i as per lines 5 and 6, where $Sens$ is a data sensitivity value representing the minimum distance between plaintext values in the dataset to be encrypted (calculated as specified in [16]). The value of δ_i is sampled for each interval so that longer intervals with a larger $scale_i$ value will consequently have a larger δ_i value than in the case of shorter intervals which contribute toward the hiding of the data distribution. The algorithm will exit (line 8) with cyphertext x' calculated as in line 7. The random value δ_i is added so that identical attribute values will not have the same encryption.

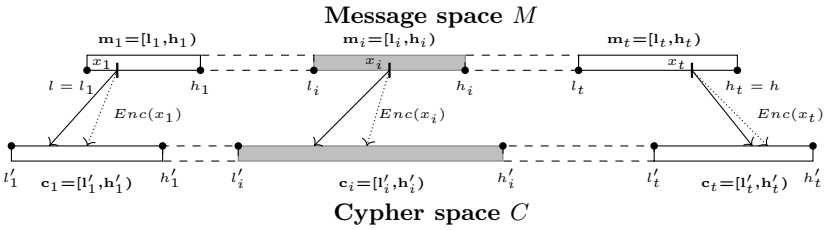


Fig. 2. Message and cypher space splitting

4.2 Liu's Homomorphic Encryption

The Liu's scheme is a symmetric HE scheme that supports cypher addition $\oplus$, cypher multiplication $\otimes$ and the multiplication of cyphertexts by plaintext values $*$. Given a data attribute value v, this is encrypted to m sub-cyphers; E=$\{e_1, \ldots, e_m\}$ where $m \geqslant 3$. The same key (Key) is used for the encryption and decryption processes; $Key(m) = [(k_1, s_1, t_1), \ldots, (k_m, s_m, t_m)]$. The key generation process is as presented in [17]. Algorithm 2 shows the pseudo code for the encryption process, $Encrypt(v, Key(m))$. The pseudo code for the data decryption process, $Decrypt(C, Key(m))$, is given in Algorithm 3.

Algorithm 2 Liu's HE encryption algorithm

procedure ENCRYPT($v, Key(m)$)
  $R = [r_{[1]}, ..., r_{[m-1]}]$, `list of real random numbers`
  $E =$ `Real value array of` m `elements`
  $e_1 = k_1 \times t_1 \times v + s_1 \times r_m + k_1 \times (r_1 - r_{m-1})$
  for $i = 2$ to $m - 1$ **do**
    $e_i = k_i \times t_i \times v + s_i \times r_m + k_i \times (r_i - r_{i-1})$
  end for
  $e_m = (k_m + s_m + t_m) \times r_m$
  Exit with **E**
end procedure

Algorithm 3 Liu's HE decryption algorithm

procedure DECRYPT($E, Key(m)$)
  $t = \sum_{i=1}^{m-1} t_i$
  $s = \frac{e_m}{(k_m + s_m + t_m)}$
  $v = \frac{(\sum_{i=1}^{m-1}(e_i - s * s_i)/k_i)}{t}$
  Exit with **v**
end procedure

Liu's scheme has both security and homomorphic properties. The scheme is semantically secure in that it produces different cyphertexts for the same plaintext on each occasion, even when the same secret key is used. Further detail regarding the security of Liu's scheme is given in Section 7. In terms of its homomorphic properties, as noted above, the scheme support $\oplus$, $\otimes$ and $*$ as shown in Equation 1 (where c is a plaintext value), and thus, by extension, supports cypher subtraction $\ominus$ and division $\oslash$ as shown in Equation 2.

$$\begin{aligned} E \oplus E' &= \{e_1 \oplus e'_1, \ldots, e_m \oplus e'_m\} &&= v + v' \\ E \otimes E' &= \{e_1 \otimes e'_1, \ldots, e_1 \otimes e'_m, \ldots e_m \otimes e'_1, \ldots, e_m \otimes e'_m\} &&= v \times v' \\ c * E &= \{c * e_1, \ldots, c * e_m\} &&= c \times v \end{aligned} \tag{1}$$

$$\begin{aligned} E \ominus E' &= E \oplus (-1 * E') \\ c \oslash E &= \frac{1}{c} * E \end{aligned} \tag{2}$$

5 Secure Chain Distance Matrices (SCDMs)

Liu's scheme described above, does not preserve the data ordering in the generated cyphers. Therefore record comparison, an operation frequently required by many data mining algorithms, cannot be directly applied. To facilitate cyphertext comparison the idea of SCDM, presented recently in [3], was adopted. For the purposed of completeness the SCDM concept is presented in this section.

A SCDM is a 2D matrix that holds the encrypted distances between the attribute values in every *consecutive* data records in a dataset D in whatever ordering the records appear in the dataset. Therefore, the first dimension is $r-1$,

where r is the number of records in D, and the second is the size of the attribute set a. A SCDM has a *linear chain feature* that allows secure derivation of the distances between any pair of data records held in the SCDM without decryption, while at the same time requiring less storage space than that required by alternative distance matrix formalisms, such as the Updatable Distance Matrices (UDMs) proposed in [2]. Given a SCDM a TPDM can determine the similarity between two records, r_x and r_y, where $x \neq y$ as per Equation 3. In the case of $x = y$ the distance will clearly be 0. The SCDM is generated in two steps: (i) `CDM calculation` and (ii) `CDM encryption`:

CDM Calculation: Algorithm 4 gives the `CDM Calculation` process. The algorithm starts by dimensioning the desired CDM (line 2) according to the dimensions of D received as an input. As noted above, the first dimension is the number of records in dataset minus one $(r-1)$ and the second is the size of attributes set (a). The CDM elements are then populated (lines 3 to 7); element $CDM_{i,j}$ will hold the distance between the jth attribute value in record i and the same attribute value in record $i+1$ (this can be a negative value).

CDM Encryption: The CDM, as the case of the UDM presented in [2], is essentially a set of linear equation that may support reverse engineering. To preclude the potential of reverse engineering, the CDM needs to be encrypted in such a way that the data distribution and frequency are hidden, while at the same time preserving the ordering in the generated cyphertexts. To this end, the FDH-OPE scheme described in Sub-section 4.1 above was used. The key feature of the encrypted CDM, the SCDM, is that a TPDM now has access to the "distances value ordering" facilitated by the FDH-OPE scheme, but not the original distance values, between the data records. This means that the TPDM can calculate the order of difference between records.

$$Sim(SCDM, r_x, r_y) = \sum_{j=1}^{j=a} \left| \sum_{i=x}^{i=(y-1)} SCDM_{i,j} \right| \tag{3}$$

Algorithm 4 CDM calculation

```
1: procedure CDMCalculation(D)
2:     CDM = ∅ array of r − 1 rows and a column
3:     for i = 1 to i = r − 1 do
4:         for j = 1 to j = a do
5:             CDM_{i,j} = d_{i,j} − d_{i+1,j}
6:         end for
7:     end for
8:     Exit with CDM
9: end procedure
```

6 Secure Query Processing over Encrypted Data with Query Controllability and Key Confidentiality

This section presents the proposed SkNN data classification and SkNN data querying process designed to achieve the key security requirements of: (i) *key confidentiality* from QUs, (ii) *query controllability*, (iii) *data privacy* and (iv) *query privacy*; without involving the DO and/or QUs while a query is processed and at the same time maintaining the efficiency and accuracy of the data classification. The solution is founded on the concept of SCDMs as described in Section 5. The proposed SkNN algorithm consists of three main steps as follows:

1. **Query encryption:** The secure encryption of the QU's query record to preserve privacy, while maintaining DO encryption key confidentiality. To this end the Secure Query Cyphering (SQC) protocol is used, described in further detail in Sub-section 6.1.
2. **Binding process:** The "binding" of the encrypted query q' with the SCDM to allow the data similarity between the contents of D' and q' to be determined. The binding process is detailed in Sub-section 6.2 below.
3. **SkNN data classification:** Query resolution (classification) conducted in two further steps: (i) nearest neighbour records retrieval and (ii) major class label determination. Both are discussed further in Sub-section 6.3.

6.1 Secure Query Cyphering (SQC) Protocol

The SQC protocol operates between the DO and QUs and is designed to allow the QUs to encrypt a query record, $q_i = \{q_{i,1}, q_{i,2}, \ldots, q_{i,a}\}$, using FDH-OPE, so that a "binding" record can be generated which in turn is utilised by the TPDM to update its SCDM. The binding process and the updating of the SCDM is discussed in the following sub-section, this sub-section presents the SQC protocol. To encrypt the query record q_i, using the FDH-OPE scheme, QU requires the FDH-OPE key. As FDH-OPE is a symmetric scheme, that uses the same key for encryption and decryption, sharing the key with the QU presents a security risk. The idea, instead of providing the FDH-OPE key, is therefore to provide the QU with the parameters to allow FDH-OPE encryption. However, provision of these parameters still presents a security threat. Therefore the parameters are encrypted using Liu's Scheme; recall that this is an HE scheme whose functionality will allow FDH-OPE encryption of q_i without decryption of the parameters. In effect q_i will be double encrypted, firstly using FDH-OPE to give q'_i, and secondly using Liu's scheme to give q''_i. Note that the Liu HE scheme keys used with respect to the SQC protocol is different to the Liu HE scheme keys used to encrypt D (see Section 3). To distinguish between the two, the former will be referred to as the *Shared Liu* scheme (shared because later in the SkNN process it is shared with the TPDM).

Recall that Using FDH-OPE a value x is encrypted as follows (line 7 of Algorithm 1):

$$x' = l'_j + scale_j \times (x - l_j) + \delta_j \tag{4}$$

where l'_j is the minimum bound for the cypher space interval in question, $scale_j$ is the required scaling between the message space interval and the corresponding cypher space interval, and δ_j is a noise value included to prevent identical values being encrypted in the same way on repeated encryptions. The above can be rewritten as follows (with noise δ_j removed):

$$x' = scale_j \times (x) + (l'_j - (scale_j \times (l_j)) \tag{5}$$

which can be further simplified to

$$x' = scale_j \times (x) + e_j \tag{6}$$

where $e_j = l'_j - (scale_j \times (l_j))$. The parameters $scale_j$ and e_j are calculated by the DO, encrypted using the Shared Liu scheme to give $scale'_j$ and e'_j, and sent to the relevant QU. Of course the values of $scale_j$ and e_j are dependent on the interval in which x falls; thus this also needs to be established within the context of the SQC protocol. The SQC protocol to achieve the above can be summarised as follows:

SQC Protocol: Secure Query Cyphering

1: **DO** generates the Shared Liu key.
2: Using binary questioning with the **QU**, **DO** identifies the FDH-OPE interval ID within which each query attribute value in $q_{i,j} \in q_i$ is contained.
3: **DO** calculates the FDH-OPE values for $scale_j$ and e_j for each attribute value $q_{i,j}$.
4: **DO** encrypts the $scale_j$ and e_j values using the Shared Liu scheme to arrive at $scale'_j$ and e'_j.
5: **DO** sends $scale'_j$ and e'_j to **QU**.
6: Using $scale'_j$ and e'_j, **QU** double encrypts the query attribute values in $q_{i,j} \in q_i$ using the HE properties of Liu's scheme as per Equation 7, the result is q''_i.

$$q''_{i,j} = (q_{i,j} * scale') \oplus e' \tag{7}$$

6.2 QU Authorisation and Binding

The binding process is the process whereby a query record is incorporated into the SCDM held by the TPDM. Recall that the SCDM contains distances (differences) between corresponding attribute values in a pairs of records. What we wish to do is add the difference between the first record in D held by the DO and the query record q held by the QU without sending either to the TPDM. The binding process is a collaborative process between the DO and a QU, and is required not only to allow a response to QU's query, but also so that the query can be authorised by the DO.

The process starts with the DO generating a random record p of length a, $p = \{p_1, \ldots, p_a\}$. This is then encrypted twice, firstly using the FDH-OPE scheme to give p', and secondly using the Shared Liu scheme to give p'', which is then sent to the relevant QU. The double encryption is required because, to retain the confidentiality of the FDH-OPE key held by the DO, q_i is also double encrypted. QU will then generate a binding record $BindRec_1$ representing the difference between their double encrypted query record $q''$and the p''. This is achieved using the Shared Liu scheme properties, thus $BindRec''_1 = q'' \ominus p''$ (as described in Sub-section 4.2). The binding record $BindRec_1$ is then sent to the TPDM (see Figure 1). At the same time the DO will calculate the binding record $BindRec_2$, representing the distances between p' (single encryption using FDH-OPE) and the first record in their dataset D, also encrypted using FDH-OPE. The binding record, $BindRec_2$, encrypted using FDH-OPE to give $BindRec'_2$, is then sent to the TPDM. The receipt of $BindRec'_2$ by the TPDM from the DO signals "approval" for the query, without this the TPDM will not process the query. The role of DO and QU is now finished.

Once the TPDM has received $BindRec''_1$ and $BindRec'_2$, the TPDM decrypts the double encrypted $BindRec''_1$, using the Shared Liu scheme, to give $BindRec'_1$. Both binding records remain encrypted using FDH-OPE. The TPDM then creates a $Pivot$ record by adding $BindRec'_1$ to $BindRec'_2$. The $Pivot$ record will now hold the distance between the query record q and the first record in $d_1 \in D$ without either being confided to the TPDM, or each other. The pivot record is then added to the SCDMs. The similarity between the query record q_i (at index 1 in the updated SCDM) and the xth record in dataset is calculated using Equation 8.

$$Sim(SCDM, Q, r_x) = \sum_{j=1}^{j=a} \left| \sum_{i=1}^{i=(x+1)} SCDM_{i,j} \right| \tag{8}$$

6.3 Third Party Data Classification

The processing (classification) of queries from an authorised QUs (note that the DO may also be a QU) is entirely delegated to the TPDM (CSP). The main purpose of using a TPDM is because: (i) the limited computing resource and technical expertise that DOs are anticipated to have, the assumption is that the DO's core business is not data analytics, but some other form of commerce where data is generated which the DO is prepared to share for commercial gain; and (ii) that DOs and QUs are likely to want avail themselves of the analytical capabilities offered using a mobile device of some kind. Using a TPDM for query resolution also provides the additional benefit that query outcomes are not shared with the DO. Algorithm 6 shows the pseudo code for SkNN data classification. The inputs are: (i) the SCDM on completion of the binding process whereby the distance between the query record and the first record in D has been inserted at index 1 ($SCDM_1$), (ii) the encrypted dataset D' and (iii) the desired value for k. The SkNN process comprises two stages: (i) secure NN

retrieval (lines 2 to 5) and (ii) determination of the major class label (line 7 which call procedure given in lines 10 to 17). The first stage starts with the calculation of the similarity between query record q' and each other record $d'_j \in D'$ as per Equation 8. The calculated distance, together with the associated class label held at $d'_{j,a+1}$, is added to the neighbour list N (line 5). The second stage, determining the major class label, is commenced by ordering the neighbour list according to the *dist* values (line 11). Recall that the FDH-OPE scheme used to encrypt the SCDM is an order preserving encryption scheme, thus facilitating secure data ordering. The first k elements in the neighbour list are then used to create list C that holds counts of the number of records in the first k elements in N that correspond to each label featured in the first k elements in N. The maximum class label is returned as the query label (line 13).

Algorithm 6 Secure kNN classification algorithm

```
1: procedure SkNN(SCDM, D',k)
2:     N = ∅
3:     for j = 1 to j = |D'| do
4:         dist = Sim(SCDM, 1, j)
5:         N = N ∪ < dist, d'_[j,a+1] >
6:     end for
7:     predictedClass = majorClassLabel(N, k)
8:     Exit with predictedClass
9: end procedure
10: procedure majorClassLabel(N, k)
11:     Order N using N < dist >
12:     C = {c_1, ..., c_l}
13:     for i = 1 to i = k do
14:         c_[N_i<label>] = c_[N_i<label>] + 1
15:     end for
16:     Exit with Max(C)
17: end procedure
```

7 Experimental Evaluation

The evaluation of the SkNN system, including the SCDM, the binding process and the SQC protocol, is presented in this section. For the evaluation both synthetic data and fifteen datasets from the UCI data repository [14] were used, the latter listed in Table 2. The objectives were to consider the proposed solution in terms of: (i) computation and communication costs on behalf of the DO, (ii) computation and communication costs on behalf of QUs, (iii) performance in terms of runtime, (iv) classifcation accuracy, (v) the security of the proposed approach and (vi) scalability; each discussed in detail in Sub-sections 7.1 to 7.6.

7.1 DO Cost Analysis

The DO will participate in preparing data for the TPDM, running the SQC protocol and authorising QU queries. As noted earlier, there is no DO involvement in the processing of QU queries once authorisation has taken place. The data preparation encompasses: (i) the generation of secret keys, (ii) data encryption, (iii) CDM calculation and (iv) CDM encryption to produce a SCDM.

Key generation is a one time process that does not add any overhead on behalf of the DO. Experiments demonstrated that the average time required to generate the FDH-OPE encryption keys was 80.32ms, whilst the Liu's HE scheme keys were generated in 1.39ms. The magnitude of the remaining DO participation is dependant on the size of the DO's dataset. Therefore, twenty synthetic dataset of differing size where used; ten synthetic datasets were directed at evaluating the effect of the number of data records (r) and the remaining ten were directed at evaluating the effect of the number of data attributes (a). The size of the targeted dimension (r or a) was increasing from $1K$ to $10K$ in steps of $1K$, while the other dimension was kept constant at 100. The results are shown in Figure 3. As expected, the average runtime required to encrypt D, generate the CDM and encrypt the CDM increases linearly as the size of r and a increases. For example, when $r = 1K$ the data was encrypted in 6.88ms; the CDM was generated in 63.73ms and encrypted in 168.04ms, when $r = 10K$ the corresponding runtimes are 19.00ms, 468.31ms and 1101.37ms. The recorded runtimes when $a = 1K$ were 3.81ms, 60.7ms and 158.57ms, compared to 18.24ms, 569.99ms and 1225.79ms when $a = 10K$. These results shown that regardless of dataset size, at least in the context of the conducted experiments, the runtime associated with DO participation was not significant and therefore does not introduce any limiting overhead with respect to the DO.

The SQC protocol requires DO participation in determining and encrypting the scale *scale* and e values required by FDH-OPE scheme so as to allow QUs to encrypt their queries. The runtimes for calculating *scale* and encrypting e were 0.16ms and 0.11ms respectively, which means that no significant computational overhead is encountered by the DO. The DO also participates in the generation and encryption of the binding record $BindRec_2$, this also does not introduces any significant overhead. Table 1 shows the recorded runtimes (ms) for different dimension of $BindRec_2$ records.

7.2 QU Cost Analysis

The QU participates in the SQC protocol to encrypt their query records and compute the binding record, $BindRec_1$, that is compared to the DO's binding record, $BindRec_2$, to produce the *Pivot* record to be included in the SCDM held by the TPDM. This novel approach allows the TPDM to securely resolve the QU's query without involving the DO or QU. Table 1 shows the time required to encrypt a range of query records of increasing length (number of attributes) and the time required by a QU to calculate a binding record $BindRec_1$. Inspection of the table indicates that the runtimes are negligible.

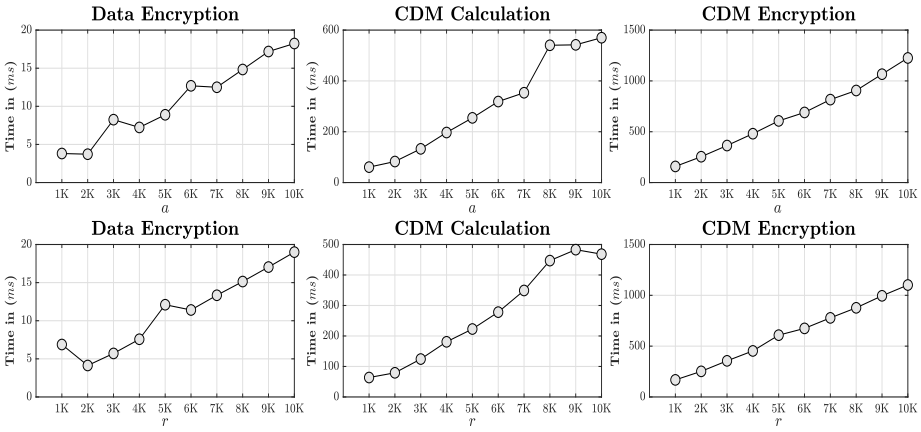


Fig. 3. Average runtimes (ms) for data encryption, CDM generation and CDM encryption using a range of values for r (number of records) and a (number of attributes)

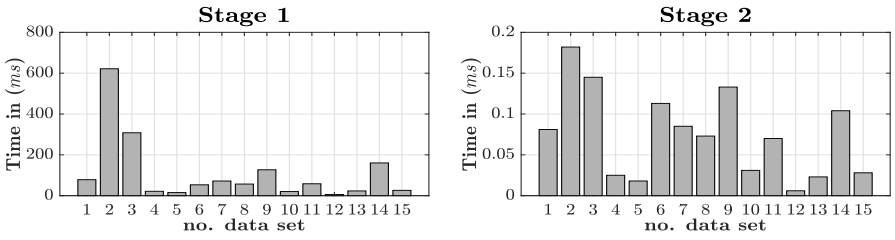
7.3 Performance of SkNN

The runtime required to classify data using the proposed SkNN approach was compared with the runtime required for the standard kNN algorithm operating over un-encrypted data. Figure 4 shows the average recorded runtimes required to classify the datasets for the two stages of the kNN algorithm: secure NN retrieval (Stage 1) and determination of the major class label (Stage 2). The x-axis gives the evaluation dataset ID number from Table 2. The reported runtime were measured in terms of average runtime obtained using Ten-fold Cross Validation (TCV). As expected, the overall time required for SkNN Stage 1 was longer than in the case of standard approach. Note that runtimes for (standard) kNN Stage 1 are reported in millisecond (ms), while runtimes for SkNN Stage 1 are reported in second (sec). The experiment shows that, the bigger the dataset the larger the SCDM, and consequently the greater the time required to interact with the SCDM to classify a record. However, inspection of the recorded results

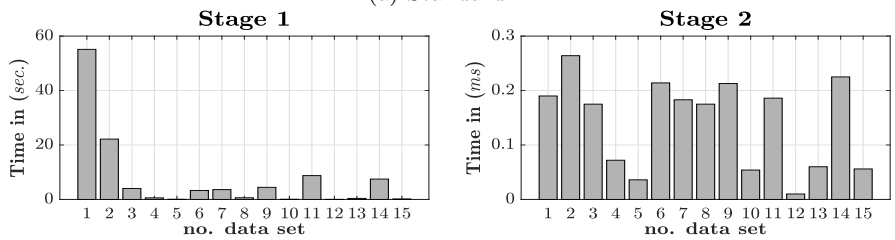
Table 1. Average runtimes (ms) for DO and QU participation when generating binding records and encrypting the query in the context of different values of a (number of attribute values)

	a									
	1K	**2**K	**3**K	**4**K	**5**K	**6**K	**7**K	**8**K	**9**K	**10**K
Encrypt query record (DO and QU)	4.42	6.11	10.77	11.28	11.58	13.41	14.24	15.5	17.76	18.89
Generate and encrypt the $BindRec_1(QU)$	2.32	5.02	6.33	6.94	8.75	9.27	9.37	11.4	11.61	13.77
Generate and encrypt the $BindRec_2(DO)$	2.38	4.23	9.47	7.03	8.9	9.85	12.04	13.94	15.62	16.38

indicates that this did not present a significant overhead. The Stage 2 runtimes were almost the same since the major class was determined over non-encrypted class labels in both cases.



(a) Standard kNN



(b) SkNN

Fig. 4. Comparison of runtimes using standard kNN and SkNN classification

The effect of the size of a query record, measured in terms of a (number of attribute values) and the selected value for k was also evaluated. A range of values for a was considered from 1K to 10K increasing in steps of 1K, coupled with $k = 1$, $k = 5$ and $k = 9$. The required classification runtime in each case is plotted in Figure 5. As expected, the runtime increases as the size of the query record increases, whilst the value of k does not introduce any significant overhead.

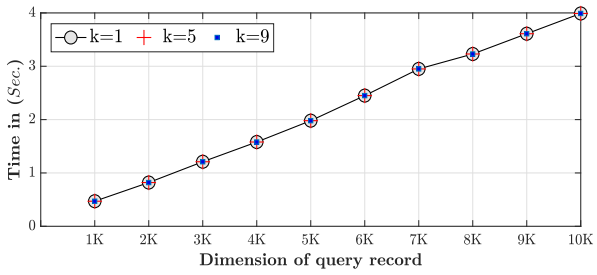


Fig. 5. Average computation costs of SkNN for varying number of k and number of attributes in query record

7.4 Classification Accuracy

The classification accuracy obtained using the proposed SkNN was compared with the accuracy obtained using standard kNN. The aim was to evidence that SkNN operated correctly; the accuracy values obtained should be comparable. The UCI evaluation datasets were split into training (the outsourced dataset D) and testing (the query set Q). Average Precision, Recall and F1 measure [21] were used as the evaluation metrics obtained using TCV. So as to conduct a fair comparison the same value for k was used in all cases. The results are presented in Table 2. From the table it can be seen that from the fifteen datasets considered, in six cases the results obtained were different (highlighted in bold font); interestingly in five of the cases SkNN produced a better performance. In the remaining cases the performance was not as good (lower F1 value recorded in the context of Arrhythmia). The difference, it was conjectured, was because the FDH-OPE scheme does not support equality matching in that two identical plain text values will have different encrypted equivalents because of the δ random noise added. Sometimes this operated in favour of SkNN by preventing overfitting. The overall average Precision, Recall and F1 values were 0.71, 0.72 and 0.71 for Standard kNN and 0.72, 0.73 and 0.72 for SkNN, indicating that both approaches produced similar results and therefor the proposed SkNN operated correctly.

Table 2. Comparison of prediction accuracies using Standard kNN and SkNN (differing results highlighted in bold font)

no. UCI DataSet	**Standard kNN**			**SkNN**		
	Precision	Recall	F1	Precision	Recall	F1
1. Arrhythmia	0.25	0.22	**0.24**	0.25	0.22	**0.23**
2. Banknote Authent.	1.00	1.00	1.00	1.00	1.00	1.00
3. Blood Transfusion	**0.60**	**0.59**	**0.60**	**0.61**	**0.61**	**0.61**
4. Breast Cancer	0.64	0.63	0.63	0.64	0.63	0.63
5. Breast Tissue	0.57	0.57	0.57	0.57	0.57	0.57
6. Chronic Kidney	0.82	**0.84**	0.82	0.82	**0.85**	0.82
7. Dermatology	**0.90**	**0.90**	**0.90**	**0.92**	**0.92**	**0.92**
8. Ecoli	**0.58**	**0.61**	**0.59**	**0.65**	**0.69**	**0.67**
9. Indian Liver Patient	0.58	0.58	0.58	0.58	0.58	0.58
10. Iris	0.96	0.96	0.96	0.96	0.96	0.96
11. Libras Movement	0.88	0.87	0.87	0.88	0.87	0.87
12. Lung Cancer	**0.45**	**0.51**	**0.47**	**0.50**	**0.58**	**0.52**
13. Parkinsons	0.81	0.82	0.81	0.81	0.82	0.81
14. Pima Disease	0.67	0.66	0.66	0.67	0.66	0.66
15. Seeds	0.90	0.90	0.90	0.90	0.90	0.90
Average	**0.71**	**0.72**	**0.71**	**0.72**	**0.73**	**0.72**

7.5 Security Under The Semi-Honest Model

Using the SkNN approach, the TPDM and QUs are assumed to be non-colluding parties and the TPDM is considered to be a "passive adversary" who follows the semi-honest model where the proposed solution (algorithms and protocols) are honestly executed. This assumption is reasonable since the primary objective of CSPs, acting as TPDMs offering DMaaS, is to deliver a high quality services to clients (DOs). The privet data of a DO and the privet queries of a QU are not shared with any other parties in the proposed system. The TPDM is the only party who gains access to the encrypted dataset D', SCDM and the query binding records. No decryption takes place at the TPDM side which implies even more security.

To better evaluate the strength of the proposed scheme, potential attacks were divided into two categories according to the knowledge H that the attacker possess:

Low-Level: The attacker only has access to cyphertexts; the encrypted dataset (D'), the encrypted CDM (SCDM) and the encrypted binding records; thus $H =< D', SCDM, BindRec_1'', BindRec_2' >$. In terms of cryptography a Low-Level attack therefore corresponds to a Cyphertext Only Attack (COA) [24].

High-Level: Apart from cyphertexts, the attacker also has access to at least one plaintext record $d \in D$ (but not the corresponding cyphertext for d in D'); thus $H =< D', d >$. The attacker may then be able to obtain knowledge concerning the distribution and/or frequency of records in D. In terms of cryptography a High-Level attack corresponds to a Known Plaintext Attack (KPA).

High-Level attacks present a greater threat than Low-Level attacks.

Liu's HE scheme, used to encrypt D (and the second level encryption for binding record $BindRec_1$), has been shown to be semantically secure [17], which in turn means that the SkNN approach is secure against Low-Level attacks (COAs). Deriving any information from accessing cyphertexts generated using Liu's HE scheme will be computationally expensive due to the semantically secure features incorporated into the scheme, the likely success of a Low-Level attacks is therefore negligible. In the context of the proposed FDH-OPE scheme, used to encrypt CDMs and binding records (the first level of encryption in the case of binding record $BindRec_1$), a feature of the scheme is that different cyphers are generated given plaintext values (by adding noise). The likelihood of an adversary being able to determine any information given an encrypted record d' is therefore negligible, hence the threat of a successful Low-Level attack is minimal. High-Level attacks directed at the FDH-OPE scheme, where the attacker attempts to obtain knowledge of the statistical make-up of the dataset (the data distribution and/or data frequency), are of greater concern. However, the proposed FDH-OPE scheme utilises the concept of "message space splitting" and "non-linear cypher space expansion" to obscure the data distribution in the generated cyphertexts, and a one-to-many encryption function to obscure the data frequency, thus protecting against the threat of High-Level attacks.

7.6 Scalability

The scalability of the proposed SkNN approach was measured in terms of: (i) the resource required to generate SCDMs compared to other comparable approaches from the literature, namely the Updateable Distance Matrices (UDMs) mechanism presented in [2]; (ii) the potential for extending the SkNN approach to support different data mining algorithms; and (iii) the potential of extending the approach in the context of collaborative data mining involving a number of DOs. In terms of the required memory resources the linear chain feature of SCDMs reduces the number of elements in a SCDM compared to a UDM. This is illustrated in Figure 6 which shows the number of SCDM and UDM elements with respect to a sequence of datasets increasing in size from $r = 1$K to $r = 10$K in steps of 1K (a kept constant throughout at $a = 100$). As shown in the figure, the number of UDM elements grows exponentially with the dataset size. More formally the number of elements in a UDM equates to $\frac{r(r+1)\times a}{2}$, while the number of elements in a SCDM equates to $(r-1) \times a$. The reduced memory requirement associated with SCDMs, compared to UDMs, facilitates the scalability of the proposed SkNN approach. The small number of elements in a SCDM also means that the time required to calculate the SCDM is less than that required for the UDM. In terms of extending the proposed SkNN approach to address alternative data mining algorithms, the SCDM concept can support any data mining algorithm that involve distance comparison. For example three different clustering algorithms, founded on the idea of SCDMs, were presented in [3]: Secure k-Means (Sk-Means), Secure DBSCAN (SDBSCAN) and Secure Nearest Neighbour clustering (SNNC). With respect to the concept of collaborative data mining, where a number of DOs pool their data for analysis so as to gain some mutual advantage, the proposed SkNN approach can be adapted so that the idea of Super SCDMs (SSCDMs), as presented in [4], is supported. Note that in [4] a mechanism was presented whereby SCDMs belonging to a number of DOs could be "bind" to produce a Super SCDM (SSCDM) which could then be used in the context of collaborative data clustering.

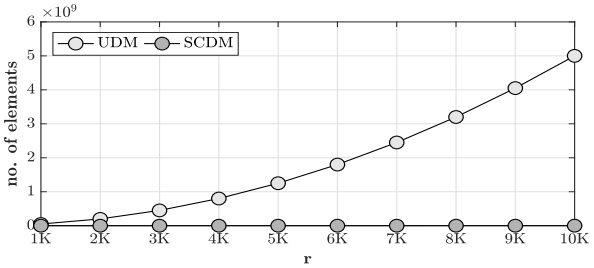


Fig. 6. Number of elements in UDM and SCDM for different number of records in dataset ($a = 100$)

8 Conclusion and Future Work

In this paper the SkNN approach to secure kNN querying (classification) has been presented that features a novel cryptographic approach. The approach delegates the required data analysis to a Third Party Data Miner (TPDM), the assumption is that this will typically be a Cloud Service Provider. SkNN operates in such a way that the data confidentiality of the Data Owner's (DO's) dataset D and the Query User's (QU's) query set Q is maintained; the dataset D belonging to the DO and the query set Q belonging to a QU are never shared. The mechanism operates using the concept of Secure Chain Distance Matrices (SCDMs), encrypted using a proposed Frequency and Distribution Hiding Order Preserving Encryption (FDH-OPE) scheme, which are generated by the DO and sent to the TPDM. For a query $q \in Q$ to be resolved by the TPDM using the SCDM received from the DO the distance information concerning q needs to be incorporated into the SCDM. To do this q first needs to be encrypted using the same FDH-OPE encryption as used by the DO to encrypt the SCDM. However, given that the FDH-OPE scheme is a symmetric scheme, it is not appropriate for the DO to share the FDH-OPE key with the QU. Instead the relevant FDH-OPE encryption parameters, encrypted using Liu's Scheme, are sent to the QU who can then encrypt q without decrypting the received parameters. The effect is that q is double encrypted (using Liu's scheme and the FDH-OPE scheme) to give q''. This is facilitated through a proposed Secure Query Cyphering (SQC) protocol. However, q'' is never shared with the TPDM. What the TPDM needs to resolve the query is to include the difference between the query record and the first record in D into the SCDM, essentially adding an additional row at the start of the SCDM. This is achieved by both the DO and the QU each generating an encrypted "binding" record, the DO with respect to the first record in D and the QU with respect to q'', and sending them to the TPDM who creates a "pivot" record to add to the SCDM. The process of the DO generating a binding record and sending it to the TPDM indicates authorisation for the resolution of the query. The TPDM then resolves the query, using a Nearest Neighbour (NN) search facilitated by the contents of the SCDM and returns the major class label to the QU. The proposed SkNN approach was evaluated by: comparing its operation with standard kNN, considering the security level provided by the approach and analysing the potential for scalability. The evaluation indicated that: (i) the SkNN approach operated in a manner comparable to Standard kNN (sometimes better) without entailing a significant runtime overhead; (ii) was robust against Low-Level (Cyphertext Only) and High-Level (Known Plaintext) attacks; (iii) had the potential to operate using "Big Data" datasets; and (iv) be applicable to other data mining activities that entail distance comparison and alternative forms of collaborative data mining.

References

1. Agrawal, R., Srikant, R.: Privacy-preserving data mining. In: Proceedings of the 2000 SIGMOD International Conference on Management of Data. pp. 439–450.

ACM (2000)
2. Almutairi, N., Coenen, F., Dures, K.: K-Means Clustering Using Homomorphic Encryption and an Updatable Distance Matrix: Secure Third Party Data Clustering with Limited Data Owner Interaction., pp. 274–285. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-64283-3_20, https://doi.org/10.1007/978-3-319-64283-3_20
3. Almutairi, N., Coenen, F., Dures, K.: Data Clustering using Homomorphic Encryption and Secure Chain Distance Matrices. SciTePress (2018), https://liverpool.idm.oclc.org/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=ir00019a&AN=uol.3023624&site=eds-live&scope=site
4. Almutairi, N., Coenen, F., Dures, K.: Secure third party data clustering using ϕ data: Multi-user order preserving encryption and super secure chain distance matrices. In: Artificial Intelligence XXXV. pp. 3–17. Springer, Cham (2018)
5. Chen, T., Chen, J., Zhou, B.: A system for parallel data mining service on cloud. In: Second International Conference on Cloud and Green Computing. pp. 329–330 (2012)
6. Das, A.K.: European Union's general data protection regulation, 2018: A brief overview. Annals of Library and Information Studies (ALIS) **65**(2), 139–140 (2018)
7. Dasarathy, B.V.: Nearest neighbor (NN) norms: NN pattern classification techniques. IEEE Computer society press (1991)
8. Domingo-Ferrer, J.: A provably secure additive and multiplicative privacy homomorphism. In: Proceedings of the 5th International Conference on Information Security. pp. 471–483. ISC '02, Springer-Verlag, London, UK (2002)
9. Elmehdwi, Y., Samanthula, B.K., Jiang, W.: Secure k-nearest neighbor query over encrypted data in outsourced environments. In: 2014 IEEE 30th International Conference on Data Engineering. pp. 664–675 (March 2014)
10. Goldreich, O.: Secure multi-party computation. Manuscript. Preliminary version **78** (1998)
11. Gostin, L.O.: National health information privacy: regulations under the Health Insurance Portability and Accountability Act. Journal of the American Medical Association (JAMA) **285**(23), 3015–3021 (2001)
12. Hu, H., Xu, J., Ren, C., Choi, B.: Processing private queries over untrusted data cloud through privacy homomorphism. 27th International Conference on Data Engineering (ICDE) pp. 601–612 (2011), https://liverpool.idm.oclc.org/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edseee&AN=edseee.5767862&site=eds-live&scope=site
13. Huang, Z., Du, W., Chen, B.: Deriving private information from randomized data. In: Proceedings of the 2005 SIGMOD International Conference on Management of Data. pp. 37–48. ACM (2005)
14. Lichman, M.: UCI machine learning repository. (2013), http://archive.ics.uci.edu/ml
15. Lindell, Y., Pinkas, B.: Privacy preserving data mining. Journal of Cryptology **15**(3), 177–206 (2002)
16. Liu, D., Wang, S.: Nonlinear order preserving index for encrypted database query in service cloud environments. Concurrency Computation: Practice and Experience **25**(13), 1967–1984 (2013)
17. Liu, D.: Homomorphic encryption for database querying. Patent **27**(PCT/AU2013/000674) (12 2013), iPC_class = H04L 9/00 (2006.01), H04L 9/28 (2006.01), H04L 9/30 (2006.01)
18. Liu, J., Xiong, L., Luo, J., Huang, J.Z.: Privacy preserving distributed DBSCAN clustering. Transactions on Data Privacy **6**(1), 69–85 (2013)

19. Liu, L., Kantarcioglu, M., Thuraisingham, B.: The applicability of the perturbation based privacy preserving data mining for real-world data. Data and Knowledge Engineering **65**(1), 5–21 (2008)
20. Liu, Z., Chen, X., Yang, J., Jia, C., You, I.: New order preserving encryption model for outsourced databases in cloud environments. Journal of Network and Computer Applications **59**, 198–207 (2016)
21. Makhoul, J., Kubala, F., Schwartz, R., Weischedel, R.: Performance measures for information extraction. In: Proceedings of DARPA broadcast news workshop. pp. 249–252. Herndon, VA, Morgan Kaufmann (1999)
22. Narayanan, A., Shmatikov, V.: Robust De-anonymization of large sparse datasets. In: Proceedings of the 2008 Symposium on Security and Privacy. pp. 111–125. IEEE (2008)
23. Rahman, M.S., Basu, A., Kiyomoto, S.: Towards outsourced privacy-preserving multiparty DBSCAN. In: 22nd Pacific Rim International Symposium on Dependable Computing. pp. 225–226. IEEE (2017)
24. Robling Denning, D.E.: Cryptography and data security. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1982)
25. Samanthula, B.K., Elmehdwi, Y., Jiang, W.: k-Nearest Neighbor classification over semantically secure encrypted relational data. IEEE Transactions on Knowledge and Data Engineering **27**(5), 1261–1273 (2015)
26. Samarati, P.: Protecting respondents identities in microdata release. IEEE Transactions on Knowledge and Data Engineering **13**(6), 1010–1027 (2001)
27. Sun, X., Wang, H., Li, J., Pei, J.: Publishing anonymous survey rating data. Data Mining and Knowledge Discovery **23**(3), 379–406 (2011)
28. Sweeney, L.: Matching known patients to health records in washington state data. (06/01/2013 2013), `http://thedatamap.org/risks.html`, available at `http://thedatamap.org/risks.html`, [Online; accessed 3-May-2019]
29. Sweeney, L., Abu, A., Winn, J.: Identifying participants in the personal genome project by name. (04/24/2013 2013), available at `http://dataprivacylab.org/projects/pgp/`, [Online; accessed 3-May-2019]
30. Takabi, H., Joshi, J.B., Ahn, G.J.: Security and privacy challenges in cloud computing environments. IEEE Security and Privacy **8**(6), 24–31 (2010)
31. Wong, W.K., Cheung, D.W.l., Kao, B., Mamoulis, N.: Secure knn computation on encrypted databases. In: Proceedings of the 2009 ACM SIGMOD International Conference on Management of data. pp. 139–152. SIGMOD '09, ACM, New York, NY, USA (2009)
32. Xu, H., Guo, S., Chen, K.: Building confidential and efficient query services in the cloud with RASP data perturbation. IEEE Transactions on Knowledge and Data Engineering **26**(2), 322–335 (Feb 2014). https://doi.org/10.1109/TKDE.2012.251
33. Xu, S., Cheng, X., Su, S., Xiao, K., Xiong, L.: Differentially private frequent sequence mining. IEEE Transactions on Knowledge and Data Engineering **28**(11), 2910–2926 (2016)
34. Yao, B., Li, F., Xiao, X.: Secure nearest neighbor revisited. In: 2013 IEEE 29th International Conference on Data Engineering (ICDE). pp. 733–744 (April 2013). https://doi.org/10.1109/ICDE.2013.6544870
35. Yiu, M.L., Assent, I., Jensen, C.S., Kalnis, P.: Outsourced similarity search on metric data assets. IEEE Transactions on Knowledge and Data Engineering **24**(2), 338–352 (2012)
36. Yuan, J., Yu, S.: Efficient privacy-preserving biometric identification in cloud computing. In: 2013 Proceedings IEEE INFOCOM. pp. 2652–2660. IEEE (2013)

37. Zhu, Y., Takagi, T., Hu, R.: Security analysis of collusion-resistant nearest neighbor query scheme on encrypted cloud data. IEICE Transactions on Information and Systems **97**(2), 326–330 (2014)
38. Zhu, Y., Wang, Z., Zhang, Y.: Secure k-NN query on encrypted cloud data with limited key-disclosure and offline data owner. In: Advances in Knowledge Discovery and Data Mining. pp. 401–414. PAKDD 2016, Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-31750-2_32
39. Zhu, Y., Xu, R., Takagi, T.: Secure k-NN query on encrypted cloud database without key-sharing. International Journal of Electronic Security and Digital Forensics **5**(3-4), 201–217 (2013)