Secure Mining of Classification in Horizontally Distributed Databases

Mohamed A. Ouda, Sameh A. Salem, Ihab A. Ali, and El-Sayed M. Saad
Department of Communication and Computer, Faculty of Engineering
Helwan University, Cairo - Egypt

Abstract
Data mining can extract interesting knowledge from large amount of data, but in distributed databases these large amounts of data is split among different parties. Privacy concerns may prevent parities from directly sharing the data which could lead to mutual benefit. This paper addresses a proposed secure mining protocol using K-Nearest Neighbor (KNN) classification method over horizontally partitioned data. The proposed protocol depends mainly on the integration of Elliptic Curve Cryptography (ECC) public key cryptosystem and Diffie-Hellman Key Exchange. Comparisons with RSA public key cryptosystem show that the proposed protocol has relatively achieved the best performance. A security analysis of the presented protocols along with the experimental results show their applicability with different parameters such as number of parties involved, and the size of the underlying datasets.

Keywords: privacy-preserving; secure multi-party computation; K-Nearest Neighbor algorithm.

I. Introduction
Data mining techniques are used increasingly by private and public sectors to identify patterns, trends and exceptions. Data mining functionalities include establishing correlations through association rules, supervised and unsupervised machine learning through classification and clustering analysis, rare events analysis through outlier detection and trend analysis for sequential and time series.

Data mining techniques in general are performed on two types of data environment: centralized databases or data warehouses and decentralized data sets. Here we use the distributed classification discovery as a scenario. The issue of violating the privacy of individuals when data to be shared is still one of the main drawbacks of distributed data mining. We study the problem of secured classification mining in horizontally partitioned databases. In that model there are several sites that hold homogeneous databases, which mean databases share the same set of attributes but hold information on different entities.

This paper is organized as follows: Literature survey and related work in section 2. Section 3 introduces the cryptographic privacy preserving mechanisms that are used. Section 4 shows the proposed protocol. Section 5 presents the implemented system. Analysis and evaluation of proposed protocol is shown in section 6 and finally the conclusion.

II. Related work
A research on privacy-preserving data mining has become of great interest since few years ago [1]-[3]. There have been proposed privacy preserving algorithms for different data mining applications, including clustering [4]-[7], association rules mining across multiple databases [8]-[10], Bayes classification [11]-[13], and decision trees [1][2]. The Secure Multiparty Computation paradigm provides cryptographic solutions for protecting privacy in any distributed computation [14][15]. Works most related to ours are [16]-[20]. Herein, we study applying Elliptic Curve Cryptography (ECC) integrated with Diffie-Hellman Key Exchange for
secured KNN classification mining. The work done on privacy preserving data mining using ECC can be also shown in [21][22]. The contributions of this paper contain the following:

- A solution for KNN classification with horizontal collaboration.
- Privacy and efficiency analysis to show the performance scaling up with various factors.

II.1 Homogeneous collaborative classifying mining with privacy preserving

There have been two approaches for privacy-preserving in multiparty classification mining. One is a randomization approach [23][24] in which altering or perturbing the data before submitting to the data miner such that the actual individual data cannot be recovered, while certain computations can still be applied to the data. The other is a cryptographic approach mostly using SMC (Secure Multiparty Computation) [25][26].

Most efficient privacy preserving solutions can be often designed for specific distributed computations. We focus on SMC-based classification mining algorithms on horizontally partitioned data in the cryptographic approach. The SMC literature defines two basic adversarial models:

- **Semi-Honest Model:** Semi-honest adversaries follow the protocol faithfully, but can try to infer the secret information of the other parties from the data they see during the execution of the protocol.

- **Malicious Model:** Malicious adversaries may do anything to infer secret information. They can abort the protocol at any time, send spurious messages, spoof messages, collude with other (malicious) parties, etc.

III. Cryptographic privacy preserving mechanisms:

A- RSA Public-Key Cryptographic Algorithm

RSA is the most widely used in public-key cryptosystem [27]. Its security depends on the fact of number theory in which the factorization of big integer is very difficult.

In RSA algorithm as in Fig.(1), key-pair \((e, d)\) is generated by the receiver, who posts the encryption-key \(e\) on a public media, while keeping the decryption-key \(d\) secret.

![ENCRYPTION SCHEMES](image)

Fig. 1: Public-key encryption schemes: an illustration

B- Diffe-Hellman Protocol

The Diffie-Hellman protocol [28] is the basic public-key cryptosystem proposed for secret key sharing. If party \(A\) and party \(B\) first agree to use a specific curve, field size, and type of mathematics, they then share the secret key by process as follows. We can see that we just need scalar multiplication \(P\) in order to implement the Diffie-Hellman protocol.

1. \(A\) and \(B\) each chose random private key \(k_a\) and \(k_b\)
2. \(A\) and \(B\) each calculate \((k_a P)\) and \((k_b P)\), and send them to opposite side.
3. \(A\) and \(B\) both compute the shared secret key \(S_k = k_a (k_b P) = k_b (k_a P)\).

C- Elliptic Curve Cryptography

The elliptic curves we are using are restricted to elements of a finite field. In our work we use ECC defined over odd prime field \(F_p\). Introduction to elliptic curves and their usage in cryptography can be found in [29]. Elliptic curve cryptosystems over finite field have some advantages like the key size can be much smaller.
compared to other cryptosystems like RSA, e.g. a 160-bit key in ECC is considered to be as secure as 1024-bit key in RSA \[30][31]\). This leads to faster encryption/decryption process. 

The security of ECC depends on the difficulty of Elliptic Curve Discrete Logarithm Problem (ECDLP), which states that, “Given an elliptic curve \(E\) defined over a finite field \(F_p\) of order \(n\), a point \(P \in E(F_p)\), and a point \(Q \in E(F_p)\), find the integer \(k \in [0, n - 1]\) such that \(Q = kP\). The integer \(k\) is called the discrete logarithm of \(Q\) to the base \(P\). It is computationally infeasible to obtain \(k\), if \(k\) is sufficiently large. Hence the main operation involved in ECC is point multiplication. i.e. multiplication of a scalar \(k\) with any point \(P\) on the curve to obtain another point \(Q\) on the curve.

IV. Proposed privacy-preserving classification in homogeneous collaborative environment
IV.1 proposed model for privacy preserving:

-Key Generation for Privacy Preserving Model:

In our proposed protocol we extend Elliptic Curve \[30\] Diffie-Hellman Key Exchange \[28\] to be multi-parties’ cryptosystem for distributed environment dataset. In the proposed model, master site and distributed slave sites on both ends of communication send a public key which can be seen by anyone. The public key is then combined with the private key to create a shared secret key which, due to the underlying mathematics, is the same on both sides. This shared secret key is then used to hash a new key that can be used by either site (master and slave site) for encrypting and decrypting messages.

![Fig.2: Generation of shared secret keys \(S_k\) at the master and slave sites](image)

Where:
\(p\): is a prime number on which the finite field \(F_p\) is defined.
\(g\): is a base point taken from the elliptic group.
\(pk_s, pk_m\): are private keys of slave site \(S_s\) and the master site \(S_m\) respectively selected from the interval \([1, p-1]\), \(1 \leq s \leq n\).
\(B_s, A\): are public keys of slave site \(S_s\) and the master site \(S_m\) respectively.

Shared secret key \(S_k\):
\[
S_{k_s} = \text{mod}(A^{pk_s}, p) = \text{mod}((g^{pk_m}, p)^{pk_s}, p) = \text{mod}(\text{mod}(g^{pk_m}, p)^{pk_s}, p) = \text{mod}(\text{mod}(g^{pk_s}, p)^{pk_m}, p)
\]

-Proposed Encryption Scheme:
1) Let \((G, E, D, M)\) be a ECC cryptography scheme, where \(G\) is an algorithm generating keys, \(E\) and \(D\) are the encryption and decryption algorithms, and \(M\) is the message space.

R S. Publication (rspublication.com), rspublicationhouse@gmail.com
2) Let \( s, 1 \leq s \leq n \), \( (n = \text{number of sites/parties}) \), and \( k \) is number of nearest neighbors beforehand.

3) At the master site and each slave site the public and private key pair of ECC algorithm is generated. Key pair for a slave site \( s \) is: \( (B_s, p_{ks}) \), \( 1 \leq s \leq n \), and for the master site is: \( (A, p_{km}) \)

4) Each slave party/site \( s \) exchanges public encryption key with the master site.

5) The shared secret key \( S_s \) is generated for each slave site \( s, 1 \leq s \leq n \) and the master site. This shared secret key is then used by both master and slave site for encrypting and decrypting of messages. Each slave site \( s \) calculate its shared secret key as follows:

\[
S_{k_s} = \text{mod}(A^{p_{ks}}, p) = \text{mod}(g^{p_{km} p_{ks}}, p)
\]

Where \( A \) is the public key of the master site, \( A = \text{mod}(g^{p_{km}}, p) \). While the shared secret key at the master site corresponding each slave site is calculated as follows:

\[
S_{k_s} = \text{mod}(B_s^{p_{km}}, p), B_s = \text{mod}(g^{p_{ks}}, p), \text{where } B_s \text{ is the public key of site } s.
\]

\[
S_{k_s} = \text{mod}(g^{p_{ks}})^{p_{km}}, p) = \text{mod}(g^{p_{km} p_{ks}}, p).
\]

6) Given a minimum distance \( d_{\text{min}} \in M \) and corresponding class label \( c_l \in M \), at site \( s_i \) as plaintext messages where,

\[
m_s = \{d_{\text{min}}, c_l\}, \quad 1 \leq s \leq n
\]

7) the encrypted values are computed as:

\[
E_{S_{k_s}}(m_s) = \text{mod}(m_s g^{p_{km} p_{ks}}, p), \quad 1 \leq s \leq n
\]

- **Proposed Decryption Scheme:**

To decrypt \( E_{S_{k_s}}(m_s) = \text{mod}(m_s g^{p_{km} p_{ks}}, p) \) at the master site, the decryption key is calculated which represents the inverse value of the shared key \( S_{k_s} \), then

\[
D_{S_{k_s}} = \text{mod}((g^{p_{km} p_{ks}})^{-1}, p) = (S_{k_s})^{-1}, \text{such that:}
\]

\[
(S_{k_s}^{-1}(E_{S_{k_s}}(m_s))) = \text{mod}((g^{p_{km} p_{ks}})^{-1}, p)(\text{mod}(m_s g^{p_{km} p_{ks}}, p)) \]

\[
= \text{mod}((g^{p_{km} p_{ks}})^{-1} g^{p_{km} p_{ks}} m_s, p) = \text{mod}(m_s, p)
\]

then,

\[
D_{S_{k_s}}(E_{S_{k_s}}(m_s)) = \text{mod}(m_s, p) = m_s, \quad 1 \leq s \leq n
\]

- **Correctness of the proposed model:**

As \( |F_p| \), is the order of the finite group \( F_p \), then, \( x^{s|F_p|} = 1 \) for all \( x \) in \( F_p \), as established from Lagrange's theorem in group theory [32]. The order \( |F_p| \), of the group \( F_p \) is known for all sites.

a- The value \( D_{S_{k_s}} = \text{mod}(g^{p_{km} p_{ks}})^{-1}, p \) at the master client will be calculated as follows:

It is known its private key \( p_{km} \), and the public key \( B_s \) of slave site \( s, B_s = \text{mod}(g^{p_{ks}}, p) \) then:

\[
\text{mod}(g^{p_{ks}}, p)|F|^{p_{km} p_{ks}} = \text{mod}(g^{p_{km} p_{ks}} g^{p_{km} p_{ks}}, p)
\]

\[
= \text{mod}(g^{p_{km} p_{ks}} g^{p_{km} p_{ks}}, p)
\]

\[
= \text{mod}(g^{p_{km} p_{ks}}, p)
\]

\[
= \text{mod}(g^{p_{km} p_{ks}})^{-1}, p
\]

b- The value \( D_{S_{k_s}} = \text{mod}(g^{p_{km} p_{ks}})^{-1}, p \) at the slave site will be computed as follows:

It is known its private key \( p_{ks} \), and the public key of master site \( A = \text{mod}(g^{p_{ks}}, p) \) then,

\[
\text{mod}(g^{p_{km}}, p)|F|^{p_{km} p_{ks}} = \text{mod}(g^{p_{km} p_{ks}} g^{p_{km} p_{ks}}, p)
\]

\[
= \text{mod}(g^{p_{km} p_{ks}} g^{p_{km} p_{ks}}, p)
\]
The result of decryption from one site $s$ is $m_s = \{d_{sm}, cl_s\}$ which represent the minimum distance of $k$ neighbors and corresponding dominant class label to produce the global predicted class of the classifier.

In homogeneous collaborative classification mining it is required to get data mining results over the whole database $DB = DB_1 \cup DB_2 \cup ... \cup DB_n$, where each $DB_i (1 \leq i \leq n)$ represents a partial database located at one of the distributed sites $P_1, P_2, ..., P_n$ (i.e. $n$-divisions). Each partial database $DB_i$ has $r$ attributes and different number of entities.

**IV.2 The K-Nearest Neighbor classifier:**

Standard data mining algorithm K-Nearest Neighbor classification \[33\][34] is an instance based learning algorithm that has been shown to be very effective for a variety of problem domains. The objective of k-nearest neighbor classification is to discover k nearest neighbors for a given instance, then assign a class label to the given instance according to the majority class of the k nearest neighbors as shown in Fig. 3. The standard KNN algorithm can be stated as follows:

a) Consider learning discrete-valued target functions of the form $f : R^r \rightarrow C$, where $C$ is the finite set $c_1, c_2, ..., c_s$, where $c_1, c_2, ..., c_s$ are class values.

b) Building training set:

For each training example $(x, f(x))$, add the example to the list training set.

c) Classification algorithm: Given a query instance $X_p$ to be classified, let $x_1, x_2, ..., x_k$ denote the k instances from training set that are nearest to $X_p$.

Then return $\hat{f}(X_p) \leftrightarrow \arg \max_{c \in C} \sum_{i=1}^k \delta(c, f(x_i))$, where $\delta(a, b) = 1$ if $a = b$ and $\delta(a, b) = 0$ otherwise.

The value $\hat{f}(X_p)$ represents the dominant class value of query instance $X_p$.

To build a k-nearest neighbor classifier, the key point is to privately obtain k-nearest instances for a given point which will be achieved in the proposed protocol.

- The distance function used in this work is the standard Euclidean distance which is defined as:

$$D(X, Y) = \sqrt{\sum_{i=1}^r (X[i] - Y[i])^2} \quad (3)$$

where $r$ is the dimension space of an instance $X$, $X[i]$ denote the $i^{th}$ component value of data object $X$, and $D(X, Y)$ is the distance between two data objects $X, Y$.

![Fig.3: The classification of a test point at K=3, based on the classes of its nearest neighbors.](index.htm)

**IV.3 Proposed protocol**

The proposed protocol presents a method for privately computing data mining process from distributed sources without disclosing any information about the sources or their data except that revealed by final classification result. The proposed protocol develops a solution for privacy-preserving k-nearest neighbor classification. In this paper, a semi-honest model for an adversary is used.
The Integrated PPDM Algorithm of K Nearest Classifier is as follows:

**Input:** \( DB_1, DB_2, ..., DB_n \) at \( n \) sites, each of \( d_j \) data objects, \( j \in \{1, ..., n\} \), each data object \( X = x_1, x_2, ..., x_r \) of \( r \) dimension space, \( r > 1 \). \( K \) = number of nearest neighbors beforehand, \( k > 1 \) integer odd value, \( y_i \) class values \( i \in \{1, ..., m\} \), \( l \) attribute values, \( X_p \) query instance

**Output:** Global class value of query instance \( X_p \).

1. \( d_{i_{\text{min}}} \) Represents minimum neighbor distance with majority class relative to query instance \( X_p \), and class label \( c_l \) is the corresponding class of \( d_{i_{\text{min}}} \).
2. \textbf{For} \( i = 1 \) \ldots \( n \) \textbf{do} // generating shared secret keys
3. Master and slave sites generate public and private key pair using ECC Algorithm;
4. The master and slave sites exchange their public keys and shared secret keys \( S_{kl} \) are constructed.
5. \textbf{End For} // generating shared secret keys.
6. \textbf{For} \( i = 1 \) \ldots \( n \) \textbf{do} // computing \( d_{i_{\text{min}}} \) (local KNN) at \( n \) sites and performing the encryption process.
7. Each site \( S_i \) locally computes \( d_{i_{\text{min}}} \) and corresponding class value \( c_l \) according to \( K \) nearest algorithm relative to query instance \( X_p \).
8. Using ECC encrypts \( (d_{i_{\text{min}}}, c_l) \) to \( C_i = E_{S_{kl}}(d_{i_{\text{min}}}, c_l) \) as per Eq. (1).
9. The encrypted value \( C_i \) is sent to the master site.
10. \textbf{End For} // computing \( d_{i_{\text{min}}} \) and performing the encryption process.
11. \textbf{For} \( i = 1 \) \ldots \( n \) \textbf{do} // Decryption process at master site
12. Decrypt \( C_i \) as per Eq. (2) to get \( d_{i_{\text{min}}} \) and \( c_l \)
13. \textbf{End For} // decryption process
14. Construct the mapping table that maps the relative difference between \( d_{i_{\text{min}}} \) with all \( d_{j_{\text{min}}} \) \( \{i \neq j \& i, j \in \{1, m \text{ to } +1, -1\} \)
15. Calculate the weight for each row in the mapping table by adding the row elements and get the sum.
16. Determine the global min distance which corresponds to min weight in the mapping table.
17. Get the predicted class that match global min distance (min weight in the mapping table).

Global computation in KNN Classifier:
Every client/site send its encrypted local \( d_{i_{\text{min}}} \), and corresponding class value \( c_l \) to master client (as a third trusted party TTP). Master client after decryption each \( d_{i_{\text{min}}} \) and its class label \( c_l \), has a sequence of \( d_{i_{\text{min}}} \), \( 1 \leq i \leq n \) (for \( n \) parties/sites) which uses to construct the permutation mapping table. To construct a mapping table we compare every value in the sequence with other values and if the result is equal or greater than zero the result inserted in the permutation mapping table will be +1 otherwise will be -1, e.g if \( d_{1_{\text{min}}} - d_{2_{\text{min}}} \geq 0 \) the value in the mapping table is +1 otherwise is -1. As an example, let us have the sequence \( d_{1_{\text{min}}}, d_{2_{\text{min}}}, d_{3_{\text{min}}}, d_{4_{\text{min}}} \) of four parties/sites and \( d_{2_{\text{min}}} < d_{1_{\text{min}}} < d_{4_{\text{min}}} < d_{3_{\text{min}}} \) then \( (d_{1_{\text{min}}} - d_{2_{\text{min}}}), (d_{1_{\text{min}}} - d_{3_{\text{min}}}), (d_{1_{\text{min}}} - d_{4_{\text{min}}}), (d_{2_{\text{min}}} - d_{3_{\text{min}}}), (d_{2_{\text{min}}} - d_{4_{\text{min}}}), (d_{3_{\text{min}}} - d_{4_{\text{min}}}) \) will be \{+1, -1, -1, -1, -1, +1\} as presented in Table 1. The weight for any element in the sequence relative to others is the algebraic sum of the row corresponding to that element. Since \( d_{2_{\text{min}}} \) has the smallest weight -2, then its corresponding class label will be the predicted value of query instance.

<table>
<thead>
<tr>
<th>( d_{1_{\text{min}}} )</th>
<th>( d_{2_{\text{min}}} )</th>
<th>( d_{3_{\text{min}}} )</th>
<th>( d_{4_{\text{min}}} )</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>+1</td>
<td>+1</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>-1</td>
<td>+1</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
</tr>
<tr>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+4</td>
</tr>
<tr>
<td>+1</td>
<td>+1</td>
<td>-1</td>
<td>+1</td>
<td>+2</td>
</tr>
</tbody>
</table>
V. Implemented System

To show the applicability of the proposed protocol and its performance when we are dealing with different number of sites/parties, the protocol has been tested on various real-world datasets as per Table 2 [35]. In these experiments, datasets are securely and horizontally distributed among 2, 3, 4, 5, and 6 sites/parties. The data objects were generated on each local site/party independently. For the central reference classification we used the union of the local data object sets. As we suppose that the central classification is optimal, we measure the performance time of our proposed approach w.r.t. the central encrypted classification. We varied both the number of data objects and the number of client sites. We compared proposed protocol to a single run of KNN classification on all data objects. In order to evaluate the proposed protocol, we carried out the local classification sequentially. We collected all encrypted representatives of all local runs, and then applied a global classification on these representatives after decryption process. For all these steps we always used the same computer. All experiments were developed using C# standard Edition 2010 on Intel® Core2 Duo, 2.0 GHz, 4 GB RAM machine.
Table 2: Data Sets

<table>
<thead>
<tr>
<th>Dataset Name</th>
<th>Attribute Characteristics</th>
<th>Number of Instances</th>
<th>Number of Attributes</th>
<th>Number of classes</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Landsat satellite images</td>
<td>integer</td>
<td>4435</td>
<td>36</td>
<td>7</td>
<td>Physical</td>
</tr>
<tr>
<td>Radar return</td>
<td>Integer, Real</td>
<td>210</td>
<td>34</td>
<td>2</td>
<td>Physical</td>
</tr>
<tr>
<td>Adult</td>
<td>Categorical, Integer</td>
<td>6000</td>
<td>13</td>
<td>2</td>
<td>Social</td>
</tr>
</tbody>
</table>

V.2 Performance Tests

The experiments that are performed are as follows:

The first experiment is done on real dataset called Statlog (Landsat Satellite) that can be available by NASA. This database was generated from Landsat Multi-Spectral Scanner image data. The database consists of the multi-spectral values of pixels in 3x3 neighborhoods in a satellite image, and the classification associated with the central pixel in each neighborhood. In the sample database, the class of a pixel is coded as a number. This sample dataset contains 36 attributes with 6,435 instances and its aim is to predict the class of the soil of an image land instance. Table 3 shows the performance and accuracy of this experiment. Table 4 shows a comparative performance and accuracy of RSA encryption method with the proposed one.

The second experiment is done on radar data set which is collected by a system in Goose Bay, Labrador. This system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kilowatts. The targets were free electrons in the ionosphere. “Good” radar returns are those showing evidence of some type of structure in the ionosphere. “Bad” returns are those that do not; their signals pass through the ionosphere. This sample dataset contains 34 attributes with 351 instances. Table 5 shows the performance and accuracy of this experiment. Table 6 shows a comparative performance and accuracy of RSA encryption method with the proposed one.

The last experiment is done on Adult model, which is known as Census Income dataset, which predict whether income exceeds $50k/year base on census data. This model contains 14 attributes with 48842 instance. Table 7 shows the performance and accuracy of this experiment. Table 8 shows a comparative performance and accuracy of RSA encryption method with the proposed one.

For comparison study we use the following notations:

- Relative time overhead = Execution time with proposed encryption / Execution time without proposed encryption.
- Speedup = Execution time of RSA/Execution time of proposed encryption
- Relative accuracy = Accuracy of RSA/Accuracy of Proposed encryption

Table 3: Execution Time and accuracy of distributed/centralized Land Satellite Images dataset

<table>
<thead>
<tr>
<th>No. of sites</th>
<th>Dataset size (No. of records)</th>
<th>Distributed data set</th>
<th>Encrypted Centralized data set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Accuracy %</td>
<td>Execution Time (ms)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Without proposed encryption</td>
<td>With proposed encryption</td>
</tr>
<tr>
<td>2</td>
<td>1490</td>
<td>82.1</td>
<td>9435</td>
</tr>
<tr>
<td>3</td>
<td>2235</td>
<td>83.9</td>
<td>10530</td>
</tr>
<tr>
<td>4</td>
<td>2980</td>
<td>84.1</td>
<td>11467</td>
</tr>
<tr>
<td>5</td>
<td>3725</td>
<td>83.0</td>
<td>12690</td>
</tr>
<tr>
<td>6</td>
<td>4470</td>
<td>84.3</td>
<td>13991</td>
</tr>
</tbody>
</table>
Fig. 5: The overhead of proposed encryption algorithm on execution time of distributed Land Satellite Images dataset.

Fig. 6: The execution time of proposed encryption algorithm on (distributed/centralized) Land Satellite Images dataset.

Table 4: Execution time and accuracy of proposed and RSA encryption protocols for distributed Land Satellite Images dataset.

<table>
<thead>
<tr>
<th>No. of sites</th>
<th>Dataset size (No. of records)</th>
<th>Execution Time (ms)</th>
<th>Speed up</th>
<th>Accuracy %</th>
<th>Relative Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Proposed encryption</td>
<td>RSA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Execution Time</td>
<td>Encryption</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1490</td>
<td>9866</td>
<td>13392</td>
<td>1.36</td>
<td>82.1</td>
</tr>
<tr>
<td>3</td>
<td>2235</td>
<td>11650</td>
<td>17491</td>
<td>1.50</td>
<td>83.9</td>
</tr>
<tr>
<td>4</td>
<td>2980</td>
<td>12884</td>
<td>22783</td>
<td>1.77</td>
<td>84.1</td>
</tr>
<tr>
<td>5</td>
<td>3725</td>
<td>14883</td>
<td>25714</td>
<td>1.73</td>
<td>83</td>
</tr>
<tr>
<td>6</td>
<td>4470</td>
<td>16482</td>
<td>29975</td>
<td>1.78</td>
<td>84.3</td>
</tr>
</tbody>
</table>
Fig 7: Execution time of proposed and RSA encryption algorithm for distributed Land Satellite Images dataset.

Table 5: Execution Time and accuracy of distributed/centralized Radar Return dataset

<table>
<thead>
<tr>
<th>No. of sites</th>
<th>Dataset size (No. of records)</th>
<th>Distributed data set</th>
<th>Encrypted Centralized data set</th>
<th>Relative time Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Accuracy (%)</td>
<td>Execution Time (ms)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Without proposed encryption</td>
<td>With proposed encryption</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Execution Time (ms)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Without encryption</td>
<td>With encryption</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Accuracy (%)</td>
<td>Execution Time (ms)</td>
</tr>
<tr>
<td>2</td>
<td>70</td>
<td>87.23</td>
<td>429</td>
<td>749</td>
</tr>
<tr>
<td>3</td>
<td>105</td>
<td>89.36</td>
<td>568</td>
<td>801</td>
</tr>
<tr>
<td>4</td>
<td>140</td>
<td>92.2</td>
<td>704</td>
<td>956</td>
</tr>
<tr>
<td>5</td>
<td>175</td>
<td>92.91</td>
<td>838</td>
<td>1185</td>
</tr>
<tr>
<td>6</td>
<td>210</td>
<td>92.1</td>
<td>987</td>
<td>1375</td>
</tr>
</tbody>
</table>

Fig 8: The overhead of proposed encryption algorithm on execution time of distributed Radar Return dataset.
Fig. 9: The execution time of proposed encryption algorithm on (distributed/centralized) Radar Return dataset.

Table 6: Execution time and accuracy of proposed and RSA encryption protocols for distributed Radar Return dataset.

<table>
<thead>
<tr>
<th>No. of sites</th>
<th>Dataset size (No. of records)</th>
<th>Execution Time (ms)</th>
<th>Speed up</th>
<th>Accuracy %</th>
<th>Relative Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Proposed encryption</td>
<td>RSA encryption</td>
<td>Proposed encryption</td>
<td>RSA encryption</td>
</tr>
<tr>
<td>2</td>
<td>70</td>
<td>3672</td>
<td>4.90</td>
<td>87.23</td>
<td>86.52</td>
</tr>
<tr>
<td>3</td>
<td>105</td>
<td>5006</td>
<td>6.24</td>
<td>89.36</td>
<td>87.23</td>
</tr>
<tr>
<td>4</td>
<td>140</td>
<td>6843</td>
<td>7.15</td>
<td>92.2</td>
<td>91.41</td>
</tr>
<tr>
<td>5</td>
<td>175</td>
<td>8188</td>
<td>6.90</td>
<td>92.91</td>
<td>92.91</td>
</tr>
<tr>
<td>6</td>
<td>210</td>
<td>9888</td>
<td>7.19</td>
<td>92.91</td>
<td>92.91</td>
</tr>
</tbody>
</table>

Fig. 10: Execution time of proposed and RSA encryption algorithm for distributed Radar Return dataset.

Table 7: Execution time and accuracy of distributed/centralized Adult dataset.

<table>
<thead>
<tr>
<th>No. of sites</th>
<th>Dataset size (No. of records)</th>
<th>Distributed data set</th>
<th>Encrypted Centralized data set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Accuracy %</td>
<td>Execution Time (ms)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Without</td>
<td>With</td>
</tr>
<tr>
<td>2</td>
<td>2000</td>
<td>78.56</td>
<td>4939</td>
</tr>
<tr>
<td>3</td>
<td>3000</td>
<td>79.38</td>
<td>5985</td>
</tr>
<tr>
<td>4</td>
<td>4000</td>
<td>79.62</td>
<td>6930</td>
</tr>
<tr>
<td>5</td>
<td>5000</td>
<td>80.04</td>
<td>8610</td>
</tr>
<tr>
<td>6</td>
<td>6000</td>
<td>80.62</td>
<td>9520</td>
</tr>
</tbody>
</table>
Fig. 11: The overhead of proposed encryption algorithm on execution time of distributed Adult dataset

Fig. 12: The execution time of proposed encryption algorithm on (distributed/centralized) Adult dataset.

Table 8: Execution time and accuracy of proposed and RSA encryption algorithm for distributed Adult dataset.

<table>
<thead>
<tr>
<th>No. of sites</th>
<th>Dataset size (No. of records)</th>
<th>Execution Time (ms)</th>
<th>Speed up</th>
<th>Accuracy %</th>
<th>Relative Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Proposed encryption</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RSA encryption</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Speed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Proposed encryption</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RSA encryption</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2000</td>
<td>5960</td>
<td>4.26</td>
<td>78.56</td>
<td>1.003</td>
</tr>
<tr>
<td>3</td>
<td>3000</td>
<td>7223</td>
<td>4.81</td>
<td>79.38</td>
<td>1.003</td>
</tr>
<tr>
<td>4</td>
<td>4000</td>
<td>8319</td>
<td>5.34</td>
<td>79.62</td>
<td>1.002</td>
</tr>
<tr>
<td>5</td>
<td>5000</td>
<td>10100</td>
<td>5.43</td>
<td>80.04</td>
<td>1.002</td>
</tr>
<tr>
<td>6</td>
<td>6000</td>
<td>11404</td>
<td>4.83</td>
<td>80.62</td>
<td>1.003</td>
</tr>
</tbody>
</table>
Our proposed protocol is performed with \(K\) number equal 3 for KNN classification algorithm. We did distributed classification with and without proposed encryption to know how the proposed encryption algorithm can affect the system performance.

Fig 5, 8, and 11 show the execution time for distributed datasets Land Satellite Images, Radar return, and Adult with and without proposed encryption scheme. The experimental results show that the overhead in performance time does not exceed 20% relative to nonencrypted value for large size dataset e.g. Adult and Land Satellite Images, which means that the overhead due to encryption for the proposed protocol is within acceptable range.

Fig. 6, 9, and 12 show the execution time of encrypted centralized system compared to the proposed encrypted distributed one. Comparing the execution time for encrypted distributed datasets of 6 sites with centralized one, the reduction of execution time in the proposed distributed system of 6 sites of total size 4470 record of Land Satellite Images dataset is not less than 41% the centralized one with the same dataset size as in table 3. But for the case of Radar Return dataset in table 5, the proposed distributed system of 6 sites of total size 210 record reduce the execution time to more than 48% of centralized one. Moreover in Fig.12 the reduction in performance time relative to centralized one for proposed distributed system of 6 sites of total size 6000 record of Adult dataset is above 45%. This is shows how encrypted centralized clustering can affect the performance of the system comparing it with the proposed distributed encrypted scheme.

- In Fig. 7, 10, and 13 show that the proposed protocol has better performance than RSA encryption algorithm with comparable/equal accuracy to. The speed up parameter ranges from 1.36 to 7.19 for the tested data sets.

VI. Analysis and evaluation of proposed protocol

To evaluate privacy preserving algorithm, we have to discuss three main issues which are: privacy (security), accuracy and efficiency.

- Privacy preserving analysis

Since all the model parameters are completely present with all parties then evaluation can be performed easily. The party that wants to classify an instance using KNN evaluation procedure can do that locally, so no interactions between parties. Thus there is no question of privacy being revealed or compromised. The result of local classification is encrypted and transmitted to the master site using local shared secret key which is semantically secured. According to the composition theorem [21], if the main protocol is partitioned into sub-protocols such that the output of one sub-protocol is the input to the next and all intermediate results are kept private, then the whole protocol would be privacy preserving.
So each party $P_i$ encrypts its output pair $d_{i\text{,}min}$ and corresponding class value $cl_i$ as per equation (1).

\[ m_s = \{d_{s\text{,}min}, cl_s\}, \ 1 \leq s \leq n \]

$m_s$ is plaintext and $E_{sk_i}(m_s) = \text{mod}(m_s g^{pk_mpk_s}, p)$, is a cipher encryption of output pair. These encrypted values are transmitted to the master client for global classification. So the output of each party is securely transmitted to the master client to compute the global classification without leaking any information about the private data of a party except its output.

- **ECC** is semantically secured due to the difficulty of the Elliptic Curve Discrete Logarithm Problem. Also using ECC in combination with Diffie-Hellman protocol is believed to make public key encryption more secure.
- The participants and the master site learn nothing but the results as in secure computation.
- No realized attack is possible, since the master site does not hold any part of the database.

### Accuracy of proposed protocol

Master client, which decrypts, $d_{i\text{,}min}$ and corresponding class value $cl_i$ produce accurate results with ECC cryptosystem. As shown in Tables 3, 5, and 7 the accuracy of the classifier for parties between 2 to 6 is 78 – 92%, and is varied according to data set size and number of parties but accuracy range is still accepted and as long as the number of parties increases the accuracy gets better.

### Efficiency of proposed protocol

Raising efficiency of the algorithm is mainly shown the decreases in time complexity. Proposed-KNN classifier algorithm reduces the time complexity mainly in two aspects.

- **First**, global $d_{g\text{,}min}$ and corresponding class value $cl_g$ are quickly generated, since the KNN classifier algorithm executed locally for every party $P_i$, this enables solutions where the communication cost is independent of the size of the database and greatly cut down communication costs comparing with centralized data mining which needs to transfer all data into data warehouse to perform data mining algorithm.
- **Second**, the length of encryption – decryption key size is shorter than other public key encryption methods (e.g. RSA) with the same level of security.

### The complexity analysis of the protocol

a- The communication cost

Let us use $\beta$ and $a$ to denote the number of bits of public key size and ciphertext sent to the master site respectively and $n$ is the total number of sites/parties. The total communication cost is the cost of $2n\beta+n\alpha$ from steps 4 and 9 in the proposed protocol.

b- The computational cost is affected by:

- The generation of $n$ cryptographic key pair and generation of $n$ shared secret keys.
- The total number of $n$ encryptions and $n$ decryptions.
- Complexity for local KNN algorithm is $O(kqr)$, where $r$ is the dimension of training sample , $q$ is number of training samples and $k$ is the parameter of KNN algorithm. Additional computations as $n^2$ additions, $n(n-1)$ subtractions and $n \log(n)$ sorting $n$ numbers.

Therefore, the complexity of KNN classifier for $n$ parties is dominant for not only the other computational costs but also for communication costs too. Consequently, the overall complexity of the proposed model = $O(nkqr)$.

### Conclusion:

In this paper we have provided a solution for K Nearest Neighbor classification with horizontal collaboration using the integration of Elliptic Curve Cryptography (ECC) public key cryptosystem and Diffie-Hellman Key Exchange. To analyze the overall efficiency not only the computational cost is considered but also the communication cost for examining the performance of proposed protocol. The computational time of the proposed protocol is compared with RSA public key encryption algorithm as well as the centralized data mining approach using real world datasets. Experimental results show that the overhead in the execution time due to proposed protocol is not exceeding more than 20% compared with non-encrypted distributed scheme in large datasets as in Land Satellite Images and adult dataset. While, it gives not less than 40% reduction in execution.
time relative to the centralized scheme with the same size of dataset. Experimental results show that the proposed protocol with ECC has better performance compared to RSA encryption algorithm with the same level of security. As demonstrated through the theoretical analysis and the experimental results, the proposed protocol achieves significant improvements in terms of privacy, accuracy and efficiency.

References:


