Optimization of k -Nearest Neighbour classifier for dealing with secure encrypted rational data

Ali Abbas Younis Al-Arbo[1], Dr. Shadi Alshehabi[2]

[1]Postgraduate Student, Department of Information Technology; The University of Turkish Aeronautical Association,
Institute Of Science And Technology, Turkey,
Ministry of higher education, Iraq

[2]Assistance Professor The University of Turkish Aeronautical Association, Institute Of Science And Technology, Turkey

ABSTRACT

Data could be facts, numbers, statics or figures accumulated to be followed by several operations to give a certain result. To do that kind of operations data meant to be organized in a way which enables the developer inject several operations to get the desired result. So here it comes the term "Database " a database is a place meant to organize and sort the data to get easily for it the right data . Several database models has been created to do the work like hierarchical Database model, Network data model, relational database, etc. But one of the most important objectives of IT- Administrators, software Developers, Networks Developers is to maintain these Data secured during the time being transferred, which shouldn't be revealed or hacked easily. One of the ways was invented to maintain a secured is to encrypt these data before being sent to the server in the database itself [9]. So the Cloud is considered as a computation server which is built, held and supplied by the platform of cloud over the net, what does cloud servers do is that they process and display the same type of competencies plus activity to a normal server but are reached remotely from a cloud service supplier. the main issue behind using cloud is that when the sender uploads the encrypted data, [1, 5, 6], the receiver has decrypt the data before downloading them in order to conduct few operations on the data itself , this kind of decryption may expose the data to be hacked easily in the server itself by other intruders. So this paper proposes to benefit the data mining strategies, by using the k-nearest classification method to help the receiver get the desired data without the need of decrypting data and exposing them to be hacked easily.

1. INTRODUCTION

In recent years, cloud figuring model is to revolutionize the way N organizations "dealing out with their own data, particularly in how they are stored, accessing and processing data. As cloud computing emerging computing paradigm attracts the attention of many enterprises to seriously take into consideration the possibility using cloud from the side of its cost-effectiveness, flexibility, and relieve executive issues. In most cases, organizations keep their own math operations plus data in cloud. In spite of the huge benefits offered by cloud maintaining the privacy of the data prevent companies from using these benefits. When the data is extremely sensitive, the contents must be encrypted contents of data before sending them to the cloud. However, when the data is encrypted, regardless of the primary encryption schemes, carry out any analysis of information becomes a very difficult task, not always decrypt the data [7, 8]. There are other privacy issues as evidenced by the coming example. Example 1: Let us assume that a company sent its encrypted client database and associated data mining tasks to the cloud. When the agent of the company wants to define the risky level for a possible new client, the agent can use a classification method for determining the level of customer risk. First, the agent should generate a record Q data for the client, containing some personal information about the customer, such as credit rating, age, marital status and so on, this record may be sent to the cloud, and the cloud will calculate the class labels to d. However, since Q contains confidential data to maintain the privacy of the client's life, etc. must be encrypted before sending it to the cloud. The above example shows that an intelligent analysis of data on the encrypted data (denoted DMED) on the cloud also need to protect the user record when the record is part of the process of data mining. Additionally, the cloud may also remove useful or confidential information on actual observations data element by a data access pattern, even with having the data encrypted. Thus, the requirements for privacy / security problems DMED a cloud of three parts: (1) The secrecy of the data got encrypted, (2) user registration request privacy, and (3) to hide data access model. Work undertaken by Privacy-preserving Data Mining (any disturbances or protocol confidential calculation based approach) cannot solve the problem DMED. Perturbed lack semantic data safe, so the data perturbation methods cannot be used to encrypt the high sensitivity data. Also perturbed data do not give so exact results of data mining. Minutes confidential computation (SMC) basic way involves data is distributed and are not encrypted in each of the participating parties. In addition, a large number of intermediate calculations performed on the basis of unencrypted data. As a result, in this paper we have proposed new ways to effectively solve the problem DMED if we assume that the encrypted data is transferred to the cloud. In particular, focusing on the issue of classification, because it is one of the most common tasks of data mining. Because each method of classification has its own advantages, to be specific, this article focuses on the performance of K-nearest neighbor for encrypted data in a cloud computational environment. Cloud deals with transferring, manipulating data between clients and storing them. You can instead of storing data on your personal device or updating apps to the newest versions, you use no matter app online, to upload your own data or use its app. doing so may expose your privacy to be revealed by intruders. lets say that's you want to upload your data over the internet to be shared with others but in a private way , one of the possibilities allows you to do that is to use cloud server , instead of uploading bare data which is easily to be hacked you may encrypt them for the first instance then doing them uploaded after that user who want to receive may inject several data mining on the data label to get the desired data , that's what is called data mining over encrypted data(DMED), this kind of procedure ensures that the privacy of the data is kept save without the need of decrypt the data over the sever.

2. RESEARCH OBJECTIVE / PROPOSED SYSTEM

In the proposed system, the system will offer new methods to effectually crack the problem DMED if we assume that the contents of sent files which are encrypted are transferred to the cloud. In particular, the system focuses on the problem of classification, since it is a primary task of data mining. Each classifying way has its own advantages, to be specific, this article focuses on the performance of K-nearest neighbor for encrypted data in a cloud computational environment.

2.1 Main Contributions

Roman SkNN protocol to facilitate the search of semantically Knn encrypted relational database offered. Under this protocol, Alice is not involved in any computation, as encrypted data are transmitted in the cloud. Because of that, no data are going to be unveiled to Alice. This protocol also meets the desired requirements. Neither T content, nor any midway outcomes should not be detected in the cloud.

•request to Bob (d) should not be detected in the cloud.
•Exit { $t'_1,...,T'_k$ } must be calculated precisely and reveals only to Bob.

In addition, no Data but{ $t'_1,...,T'_k$ } should not be disclosed to Bob. Entails a low processing load on Bob after encrypted report request is sent to the cloud. • data access patterns (e.g., records corresponding K-nearest neighbors Q) should not be disclosed to either Alice or clouds to prevent any attacks inference. Interim results observed clouds in the protocol or the newly generated, randomized encryption and random facts. Accordingly, the records of info match to the k-nearest neighbors d is unknown in the cloud. In addition, after the encrypted Bob sends his request to the cloud record, it stops with the calculations. Thus, the data access pattern further protected from Bob. For more information.

Gender	Masculine=1, feminine = 0
Chest	The chest pain kind: typical angina =1, uncharacteristic angina =2, non-anginal pain = 3, asymptomatic = 4.
trestbps	remaining blood pressure (mm Hg)
Cholestrol	serum cholesterol in mg/dl
Sugar	fasting blood sugar is> 120 mg/dl (1=true; 0=false)
Slope	rise of the ultimate workout ST segment (1=up sloping, 2=flat, 3=down sloping)
Ca	number of main vessels (0-3) colored by fluoroscopy
Thal	3=normal, 6=standard defect, 7=reversible defect
Num	identifications of the disease of heart counting from 0-4 (no occurrence)

2.2 Algorithm I

```
$SkNN_b T', q \rightarrow \{ t'_1, ..., t'_k \}$
Require: $C_1$ has $T'$; $C_2$ has sk; Bob has q
1: Bob
        (a). Computes $E_{pk}$ ($q_i$), for $1 \leq j \leq m$
        (b). Send $E_{pk}$ (q) = $E_{pk}$ ($q_1$)... $E_{pk}$ ($q_m$) to $C_1$
2: $C_1$ and $C_2$:
        (a). $C_1$ receives $E_{pk}(q)$ from Bob
        (b). for i = 1 to n do: $E_{pk}$ ($d_i$) $\leftarrow$SSEDE$_{pk}$(q),$E_{pk}(t_i)$
        (c). Send {(1,$E_{pk}(d_1)$ ,..., ( n, $E_{pk}(d_n)$) to $C_2$
3: $C_2$:
        (a). Receive { [ 1,$E_{pk}(d_1)$ ,..., n, $E_{pk}(d_n)$ } from $C_1$
        (b). $d_i \leftarrow D_{sk}E_{pk}(d_i)$, for $1 \leq i \leq n$
(c). Generate $\delta \leftarrow \{ i_1 ,..., i_k \}$, such that {$d_{i1}$ ,..., $d_{ik}$ } are the top k smallest distances among { $d_1$,...,$d_n$}
        (d). Send $\delta$ to $C_1$
4: $C_1$:
        (a). Receive $\delta$ from $C_2$
        (b). for $1 \leq j \leq k$ and $1 \leq h \leq m$ do: $\gamma_{j,h} \leftarrow E_{pk}(t_{ij,h}) * E_{pk}(r_{j,h})$, where $r_{j,h} \in_R Z_N$
        Send $\gamma_{j,h}$ to $C_2$ and $r_{j,h}$ to Bob
5: $C_2$:
        (a). for $1 \leq j \leq k$ and $1 \leq h \leq m$ do: Receive $\gamma_{j,h}$ from $C_1$ $\gamma'_{j,h} \leftarrow D_{sk}(\gamma_{j,h})$;
        send $\gamma'_{j,h}$ to Bob
6: Bob:
        (a). for $1 \leq j \leq k$ and $1 \leq h \leq m$ do: Receive $r_{j,h}$ from $C_1$ and $\gamma'_{j,h}$ from $C_2$
         $t'_{j,h} \leftarrow \gamma'_{j,h} - r_{j,h}$ mod N
```

2.3 Algorithm II

```
$SkNN_m(T', q) \rightarrow t'_1, ..., t'_k$
Algorithm 2 $SkNN_m$
Require : $C_1 hasT^1 and\ \pi; C_2\ has$ Sk; Bob has q
1: Bob Sends $E_{pk}(q) =< E_{pk}(q_1), \ldots, E_{pk}(q_m) > to\ C_1$
2: $C_1\ and\ C_2:$
(a) . $C_1 recieves E_{pk}(q) from\ Bob$
(b) . for i= 1 to n do:
•    $E_{pk}(d_i) \leftarrow SSED\ (E_{pk}(q). E_{pk}(t_i))$
•    [$d_\downarrow i$] $\leftarrow SBD\ (E_\downarrow pk\ (d_\downarrow i\ ))$
     3: for s=1 to k do:
(a) . : $C_1\ and\ C_2:$
•    [$d_{\min}$ ⬚ ] , nil $\leftarrow SMIN_n(\Theta_1, \ldots, \Theta_n), where \Theta_i$ = ([$d_\downarrow i$]$, nil)$
(b) .$C_1$ :
•    $E_{pk}(d_{min}) \leftarrow \prod_{y=0}^{l-1}$ ⬚ $E_{pk}(d_{min,y+1})^{2l-y-1}$
•    If $s \neq 1$ then , for $1 \leq i \leq n$
```

$-E_{pk}(d_i) \leftarrow \prod_{y=0}^{l-1} \square\, E_{pk}\left(d_{i,y+1}\right)^{2l-y-1}$

- for i=1 to n do:

$-T_{i\leftarrow} E_{pk}(d_{\min \square}) * E_{pk}(d_i)^{N-1}$

$T_i^1 \leftarrow T_i^{Ti}, where\ Ti \in_R Z_n$

- β$\leftarrow \pi(T^1); send\ \beta\ to\ C_2$

(c). $C_2:$

- receives $\beta\ from\ C_1$
- $\beta\ \square_i^1 \leftarrow D_{sk}(\beta\ _i), for\ 1 \le i \le n$
- Compute U , for 1 ≤ i ≤ n:

– If $\beta\ \square_i^1 = 0\ then\ U_i = E_{pk}(1)$

– Else $U_i = E_{pk}(0)$

- Send U to C_1

(d). $C_1:$

- Receive U from $C_2\ and\ compute\ V \leftarrow \pi^{-1}(U)$
- $V_{i,j}^1 \leftarrow SM(V_i . E_{pk}(t_{i,j})), for\ 1 \le i \le n\ and\ 1 \le j \le m$
- $E_{pk}(t_{s,j}^1) \leftarrow \prod_{i=1}^{n} \square\, V_{i,j}^1 . for\ 1 \le j \le m$
- $E_{pk}(t_s^1) = < E_{pk}(t_s^1 . 1), \ldots, E_{pk}(t_{s,m}^1) >$

(e). $C_1\ and\ C_2\ for\ 1 \le i \le n:$

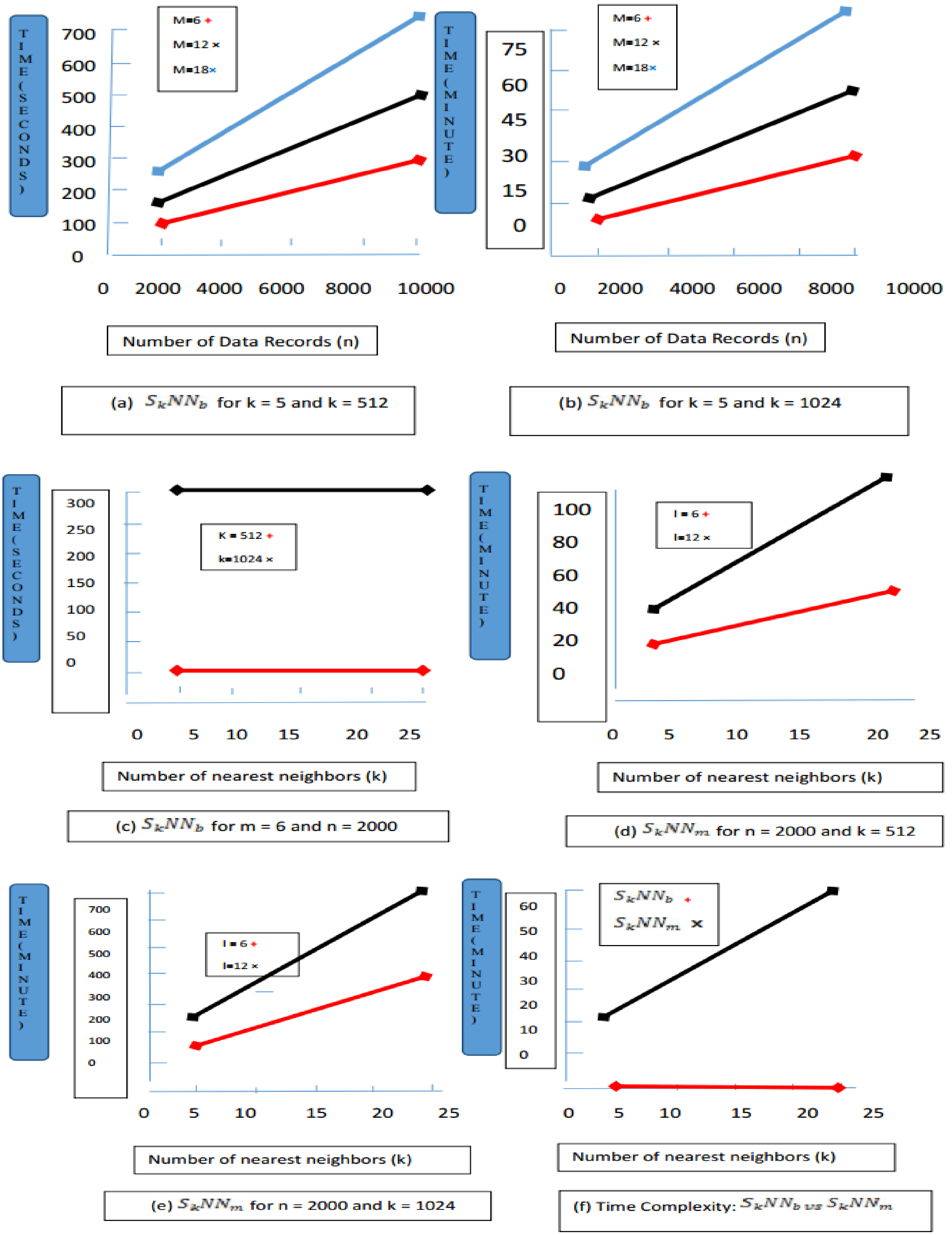
- $E_{pk}(d_{i,y}) \leftarrow SBOR\ (V_{i,} E_{pk}(d_{i,y})), for\ 1 \le y \le l$

The rest of the steps are pretty much similar to the steps 4-6 of $SKNN_b$

Performance of the Maximally Secure k-Nearest Neighbor Protocol.

3. BASIC THEORETICAL RESULTS

The computation costs of the SkNN$_m$ protocol were analyzed by the following range of values of k, l, and K. Throughout this evaluation, the values of m and n were fixed to 6 and 2000, respectively. However, the running time of the SkNN$_m$ protocol grew a bit linearly by n, m considering K is equal to 512 bits, the process costs of the SkNN$_m$ protocol for varying k and l are given in Figure 1(d). For l = 6, the running time of the SkNN$_m$ protocol varied from 11.93 to 55.65 minutes when k was changed from 5 to 25, respectively. Also, for l = 12, the running time of the SkNN$_m$ protocol varied from 20.68 to 97.8 minutes when k varied from 5 to 25, correspondingly. In both situations, the cost of the SkNN$_m$ protocol grew almost linearly with k and l. the same drift can be noticed by taking K = 1024 (see Figure 1(e)). Precisely, for any given fixed parameters, the computation cost of the SkNN$_m$ protocol increased by almost a factor of 7 when K is doubled. For example, when k = 10, the SkNN$_m$ protocol took 22.85 and 157.17 minutes are taken to cause the ten nearest neighbors of q under K = 512 and K=1024 bits, respectively. Besides, k = five, a percent of the cost equal to 69.7% in SkNN$_m$ is accounted due to the SMIN$_n$ protocol which is initiated k times in SkNNm .Also, the experienced cost due to the SMINn protocol increased from 69.7% to, at least, 75% when k was increased from 5 to 25. Additionally, the running times of both protocols were compared by fixing n = 2000, m = 6, l = 6, and K = 512 and varying values of k. The running time of the SkNNb protocol remained to be constant at 0.73 minutes because it was almost independent of k. The running time of the SkNNm protocol , however, changed from 11.93 to 55.65 minutes when k increased from 5 to 25



(a) S_kNN_b for k = 5 and k = 512

(b) S_kNN_b for k = 5 and k = 1024

(c) S_kNN_b for m = 6 and n = 2000

(d) S_kNN_m for n = 2000 and k = 512

(e) S_kNN_m for n = 2000 and k = 1024

(f) Time Complexity: S_kNN_b vs S_kNN_m

Figure 1: Time complexities of both SkNNb and SkNNm for varying values of n, m, l, k, and encryption key size K

Table 1: Sample of the data applied by algorithms

Buying	Maint	Doors	Persons	Lug_Boot	Safety
vhigh	vhigh	2	2	small	low
vhigh	vhigh	2	2	small	med
vhigh	vhigh	2	2	small	high
vhigh	vhigh	2	2	med	low
vhigh	vhigh	2	2	med	med
vhigh	vhigh	2	2	med	high
vhigh	vhigh	2	2	big	low
vhigh	vhigh	2	2	big	med
vhigh	vhigh	2	2	big	high
vhigh	vhigh	2	4	small	low
vhigh	vhigh	2	4	small	med
vhigh	vhigh	2	4	small	high
vhigh	vhigh	2	4	med	low
vhigh	vhigh	2	4	med	med
vhigh	vhigh	2	4	med	high
vhigh	vhigh	2	4	big	low
vhigh	vhigh	2	4	big	med
vhigh	vhigh	2	4	big	high
vhigh	vhigh	2	more	small	low
vhigh	vhigh	2	more	small	med
vhigh	vhigh	2	more	small	high
vhigh	vhigh	2	more	med	low
vhigh	vhigh	2	more	med	med
vhigh	vhigh	2	more	med	high
vhigh	vhigh	2	more	big	low
vhigh	vhigh	2	more	big	med
vhigh	vhigh	2	more	big	high
vhigh	vhigh	3	2	small	low
vhigh	vhigh	3	2	small	med

4. PERFORMANCE OF IMPROVING RESULTS:

Initially, the previously proposed (on-line) protocols were inapplicable, as well as expensive and not scaled for huge sets of data. However, in both cases, the processes that are entangled in each data record are free from others. Thus, operations with data records can be paralleled in order to improve efficiency. To empirically evaluate this statement, a parallel version of the SkNNb protocol was implemented using Java Framework programming and compared its computational costs with its sequential version. Recall that the machine used in the experiments had 6 cores, which could be used to perform parallel operations on 6 streams. For m = 6, k = 5 and K = 512 bits, the comparison results are shown in Figure 2. The parallel version of the SkNNb protocol was about 6 times more efficient than its serial version, because the parallel version could simultaneously perform operations on 6 data records (i.e., 6 threads in parallel). For example, the running time of parallel and serial versions of the SkNNb protocol for n = 10000 was 40 and 215.59 seconds, respectively. A similar increase in efficiency can be achieved by parallelizing transactions in the SkNNm protocol. Based on the discussion above, it was concluded that the scalability of the proposed protocols can be eliminated or mitigated, especially in a cloud computing environment where high parallel processing performance can easily be achieved. In addition, using existing methods to reduce the number of cards, you can significantly improve performance by performing parallel operations on multiple nodes. From the previous empirical analysis, it can be seen that the SMINn protocol is the most expensive subroutine used in the SkNNm protocol. Thus, increasing the efficiency of the SMINn protocol, which can improve the overall computational cost of the SkNNm protocol.

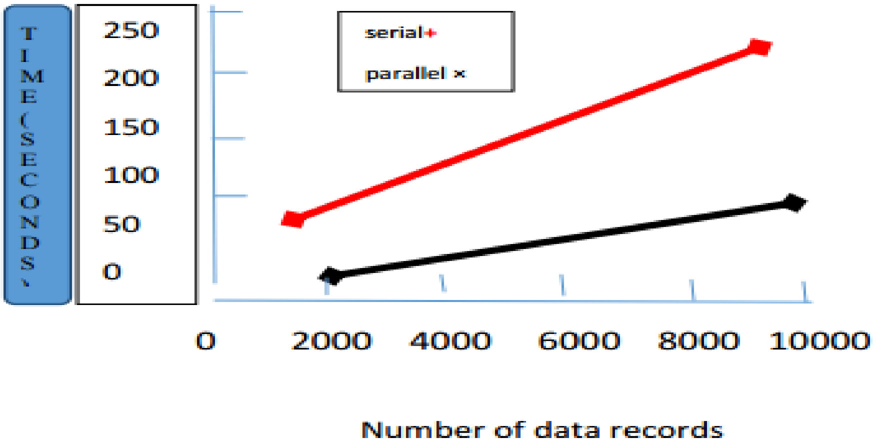
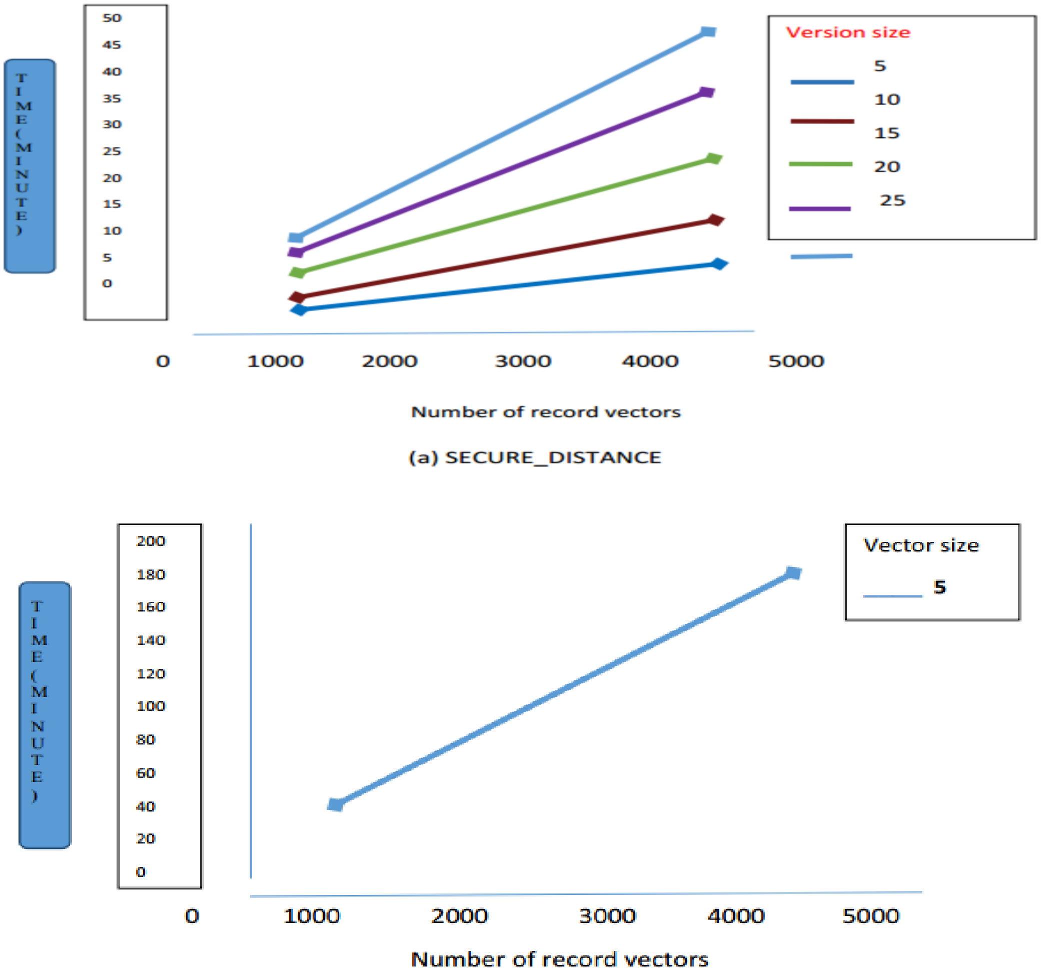


Figure 2: The Result of Parallel vs. serial versions of the SkNNb protocol for m = 6, k = 5, and K = 512

The proposed protocols are not practical if biometric authentication is to be completed in real time, although this is the best bipartisan protocol that is known. One of the main advantages of the proposed protocol is that calculations of subcomponents can be paralleled. For example, in 1 and 2 steps of the KNNm algorithm, the calculation of the safe distance and SCT are independent and can be performed simultaneously. The protocol can take benefit of (elevated) parallel computing, since it is assumed that C1 and C2 are cloud services:



(a) SECURE_DISTANCE

(b) SECURE_COMPARISON for m=5

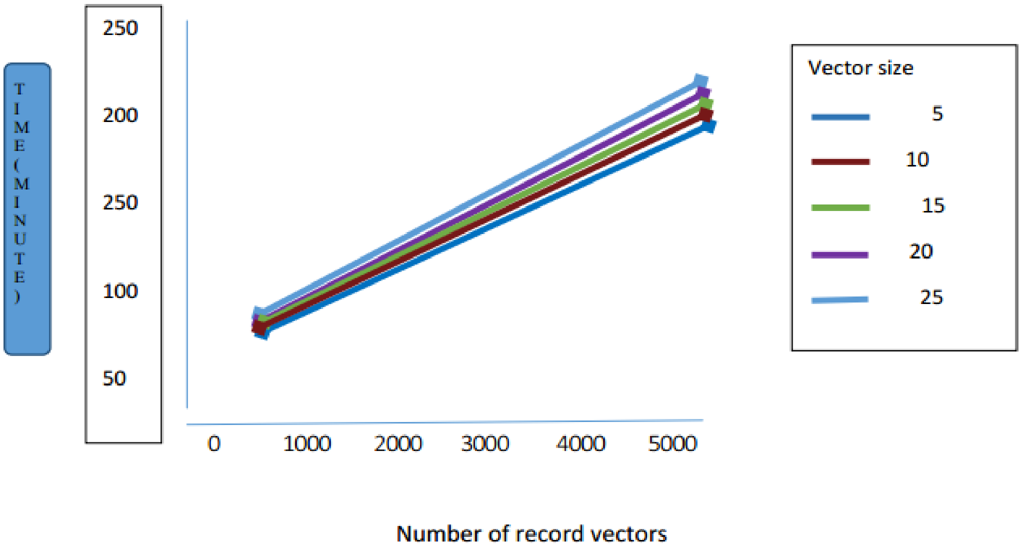
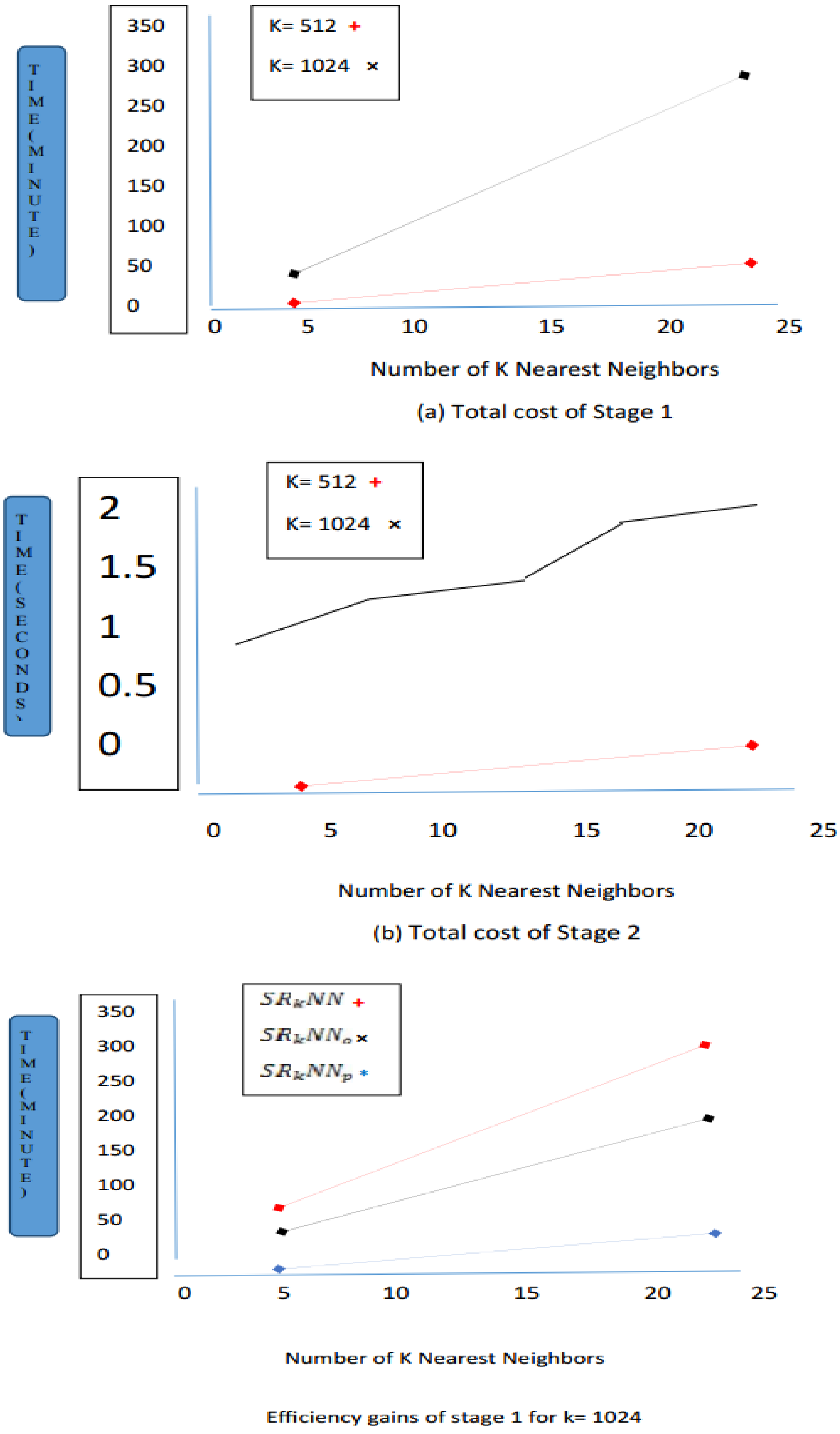


Figure 3: Computation costs of the PPkNN protocol for varying number of Knn's and different encryption key sizes in bits (K)

providers. If C1 and C2 have (n)number of nodes to do the protocol, Step 1 can be performed within a second, and Step 2 can be performed in slightly more than 2 seconds. Thus, the total running time would be approximately 5 seconds. In general, SMC-basedprivacy-preserving protocols are very expensive. Utilizing the cloud is the righttrack to make real-time applications practical.

Two diffierent approaches are proposed to boost the efficiency of Stage 1 (as the performance of the PPkNN protocol depended primarily on Stage 1). The first approach to improving the quality of Stage 1 is by pushing some computation offline. More specifically, some calculations done in Stage 1 could be precomputed (pushed offline). For example, encryptions of random numbers, 0s and 1s could be pre-computed (by the corresponding parties) in the offline phase. As a result, the online computation cost of Stage 1 (denoted by kNNm) is expected to be improved. To see the actual efficiency gains of such a strategy, the costs of kNNm were computed and compared to the costs of Stage 1 without an offline phase (simply denotedby kNN) and the results for K = 1024 bits are shown in Figure 4(c). regardless of the values of k, one could observe that SRkNNo was around 33% faster than kNN. For example, the computation costs of SRkNNo and kNN for k = 10 were 84.47 and 127.72 minutes, respectively (boosting the online running time of Stage 1 by 33.86%). The second approach to improving the performance of Stage1 is by using parallelism. Since operations on data records are independent of one another, the most computations in Stage1 could be parallelized. to empirically evaluate this claim, aparallel version of Stage1 (denoted by SRkNNp) was implemented using Java Framework programming and compared its cost with the costs of SRkNN (i.e., the serial version of Stage 1).

The computation cost of SRkNNp for K = 1024 varied from 12.02 to 55.5 minutes when k changed from 5 to 25 (see Figure 4(c)). SRkNNp was almost 6 times more efficient than SRkNN. This was because the machine used for this experiments had 6 cores. Thus, the computations could be run in parallel on 6 separate threads. Based on the above discussions, it was clear that efficiency of Stage 1 could indeed be improved significantly using parallelism. Moreover, one could also use the existing map-reduce techniques to execute parallel operations on multiple nodes to drastically improve the performance further. Hence, the level of achievable performance in the PPkNN protocol actually depended on the implementation. In contrast, Bob’s computation cost in the PPkNN protocol was mainly due to the encryption of his input query. In the dataset, Bob’s computation cost was 4 and 17 millisec- ondswhen K was 512 and 1024 bits, respectively. It was apparent that PPkNN protocol was very efficient from Bob’s computational perspective which was especially beneficial when he issued queries from a resource-constrained device (such as mobile phone and PDA).



(a) Total cost of Stage 1

(b) Total cost of Stage 2

Efficiency gains of stage 1 for k= 1024

Figure 4: Time complexities of a) SSEDR, b) SCT, and c) PPBAO by varying n and m

CONCLUSIONS

To guard the concealment of data, various privacy-preserving methods of classification have been came from the last period. The current ways are not applicable to outsourcing data from the environments of databases, where data is stored in encrypted form on the server of the manufacturer. In this paper, in this document a new privacy-preserving k- NN classification protocol is proposed for encoded data in the cloud. Our protocol protects the confidentiality of data, a user request, and hides the model data access. We also evaluated the effectiveness of the proposed protocol within altered situations. Since SMINn efficiency is an important first step towards improving the efficiency of our protocol PPkNN, we plan to explore alternative and more effective ways to solve the problem SMINn in our future work. In addition, we will explore and spread out our research to other sorting algorithms.

Cloud computing paradigm [3, 4] have recently revolutionized the way their organization operating data, in particular, in the way they store and access data. As cloud computing emerging computing paradigm attracts the attention of numerous societies consider the possibility of a cloud in terms of economic efficiency, rigidity, as well as discharge of administrative costs. Organizations often send their computing operations, in addition to their data in the cloud; otherwise, there would be no point in data outsourcing in the first place. Isolation issues and the problems of security in the cloud prevented the company from the use of these advantages, in spite of the huge benefits that the cloud offers. Therefore, due to the growth of various issues of confidentiality, confidential data got to be encoded as a first step before sending the data. Encryption as far as to reach the confidentiality of data may cause another problem in the cloud during query evaluation. In general, it is very difficult to cope with the encrypted data privacy-preserving manner without the need to decrypt. The problem here is as a cloud server is able to perform calculations on encrypted data, also the contents in cloud has to remain encrypted. so Encryption is not the only followed method to protect data , plus the diversity of multiple procedures such as randomization and secret sharing exist. You can plan to explore whether a more efficient and scalable than-based encryption solutions to these methods. In addition, the current operation can be extended to other models of the enemy, such as malicious patterns. PPQP protocols can be developed, in which a security model and evaluate malicious compromise between safety and efficacy.

REFERENCES

[1]. E. Shi, J. Bethencourt, T.-H. Chan, D. Song, and A. Perrig. Multi-dimensional range query over encrypted data. In IEEE Symposium on Security and Privacy (SP'07), pages 350–364. IEEE, 2007.

[2]. S. Bugiel, S. Nürnberger, A.-R. Sadeghi, and T. Schneider. Twin clouds: An architec- ture for secure cloud computing (extended abstract). In Workshop on Cryptography and Security in Clouds, March 2011.

[3]. R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation computer systems, Vol. 25(No.6), pages 599–616, 2009.

[4]. P. Mell and T. Grance. The dentition of cloud computing (draft). NIST special publication, 800:145, 2011.

[5]. Y. Huang, D. Evans, J. Katz, and L. Malka. Faster secure two-party computation using garbled circuits. In Proceedings of the 20th USENIX conference on Security (SEC '11), pages 35–35, 2011.

[6]. A. C. Yao. How to generate and exchange secrets. In Proceedings of the 27th Symposium on Foundations of Computer Science, pages 162–167, Washington, DC, USA, 1986. IEEE Computer Society.

[7]. J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Advances in Cryptology–EUROCRYPT 2008, pages 146–162. Springer, 2008.

[8]. E. Shen, E. Shi, and B. Waters. Predicate privacy in encryption systems. In Theory of Cryptography, pages 457–473. Springer, 2009.

[9]. M. Naor and B. Pinkas. Oblivious transfer and polynomial evaluation. In Proceedings of the Thirty-first Annual ACM Symposium on Theory of Computing, pages 245–254, Atlanta, Georgia, United States, 1999. ACM Press.