Improvising and Securing Encrypted Relational Data using k-Nearest Neighbor Classification

[1]Punam Pratap Jogdand

[1]ME Student,
[1]Computer Science and Engineering,
[1]CSMSS's Chh shahu college of Engineering Aurangabad
[1]Dr. Babasaheb Ambedkar Marathwada University, Aurangabad,Maharashtra

***Abstract*— Data mining is diverse in areas such as banking, medicine, scientific research, and government agencies. Classification is a widely used task in data mining. In the past decade, due to increasing privacy concerns, many theoretical and practical classification solutions have been offered under different security models. However, with the recent popularity of cloud computing, users now have the opportunity to outsource their data in encrypted forms, including data mining to the cloud. Because the data in the cloud is encrypted, it is not possible to use a privacy classification technique to maintain personal information. In this paper, we focus on solving encoded data classification problems. In particular, we offer a safe k-NN classification to encrypt data in the cloud. The proposed model protects confidential information, user privacy, and data access hiding. For our understanding, our work was the first to develop a secure k-NN encoder for encrypted data under semi-accurate format. We also analyze the performance of the model we offer using real data sets in various parameter settings.**

***IndexTerms*— Security, k-NN classifier, cloud databases, encryption**

I. INTRODUCTION

Recently, the cloud computing paradigm [1] is revolutionizing the way organizations conduct their data, particularly in the form of storage, access, and processing information. As the emerging computer paradigm, cloud computing has attracted many organizations to seriously consider the potential of the cloud in terms of performance, cost, flexibility, and value. Management spend more with organization as it assigns its computing operations, in addition to its data, to the cloud. Despite the huge advantage that the cloud offers, privacy and security issues in the cloud make companies unable to take advantage of these features. When data is sensitive, data needs to be encrypted before outsourcing to the cloud. However, once the data is encrypted irrespective of the encoding format used, any data mining becomes very challenging without decrypting the data.

As the emerging computer paradigm, cloud computing attracts many organizations to consider the benefits of cloud computing in terms of performance, cost, flexibility, and cost reduction. manage In cloud computing [1], [2] the owner of the data outsources his / her database and DBMS functions to the cloud with the infrastructure to host the external database and provide an access mechanism for querying and managing the database. host Data owners benefit from reduced data management costs and improved service quality with one hand. On the other hand, hosting and processing data queries of data out of control over information owners increases security challenges such as maintaining confidential information and querying privacy. The easiest way to protect the confidentiality of external data from the cloud and from unauthorized users is to encrypt the data by the owner before the data is hired. [3] In this way, the data owner can protect his or her personal information. In addition, to keep the privacy of the query, authorized users must encrypt their search terms before sending them to the cloud for evaluation.

In addition, during the query processing, the cloud can also find useful and sensitive information about the actual data by observing the data access patterns, even if the data is encrypted and the query [5]. Initially, a secure query processing need to guarantee the confidentiality of the encrypted data. [2]. The objective is to keep the confidentiality of the query records and information and to hide the information. This objective may cause other issues in the search query process in the cloud such as over head processing. Generally, encrypted data processing is very difficult without decrypting.

The question is this is how clouds can retrieve encrypted data while data stored in the cloud is encrypted at all times. In literature, there are various techniques that involve query processing through data encoding, including query intervals [6] - [8] and other aggregate queries [9], [10].

In this article, we discuss the problem of processing encrypted (SkNN) data in the cloud. The purpose of the problem is to identify k-closest neighboring queues using the encrypted T-database in the cloud. By not allowing the cloud to learn anything about the actual content of the T database and Q queries. Especially when the encoded data is outsourced to the cloud, we notice that the SkNN model is functional.

The basic necessities to be kept in design must meet following constraints :

- Keep the T and Q secrets at all times.

- Hide the data from the cloud.
- Calculate the nearest neighbor queue of the query.
- Low cost calculation on end users.
- Low cost calculation on end users.

We proposed a new approach to SkNN that addresses all of the above constraints. The model developed in this document are secured in a semi-straight format. [14] However, it can be extended to secure model under other antagonistic models, such as malicious and covert ones, using cryptographic systems based on criteria and evidence.

Suppose Alice's data owner holds the database T of the record n, which is represented by t1, ..., tn, and m attributes. Let ti, j denote the value of the attribute j th of the set ti. The default encrypts her database, which means she calculated the Epk (ti, j) value for $1 \leq i \leq n$ and $1 \leq j \leq m$, where Epk represents the cryptosystem's primary cryptosystem. Obtained results in [15] gives the database Encryption by Epk (T).

Suppose Alice outsource Epk (T) including a processing in the cloud. Consider authorized users. Bob wants to request a k-neighbor data record of the cloud that matches the input query q = hq1, .qmi from Epk (T). During this process, query Q and The contents of Bob's T-database should not be disclosed in the cloud. In addition, data access patterns should be protected from the cloud. We refer to this process as a secure kNN (SkNN) query for encrypted data in the cloud. Without loss of universality, ht 0 1, ..., t0 ki represents the record. k-to Q to Q. Then we will define the official SkNN model as follows:

SkNN(Epk(T), Q) → ht 0 1 , . . . , t0 k i

We emphasize that at the end of the SkNN model , the results ht 0 1, ..., t0 k should be revealed only to Bob. Example 1: Consider the physician who wants to know the risk factors for heart disease in a particular patient. T refers to the cardiovascular disease specimen that has the id attribute. Age, gender, cp, trestbps, chol, fbs, slope, ca, thal and num. The cardiovascular disease data provided is obtained from the UCI study repository. [16]. (For hospitals) to encrypt T attributewise. Epk (T) encrypted database outsourcing to the cloud for easy management. In addition, the data owner will assign a future search query service to the cloud.

We think that doctors who work at the hospital say Bob, who wants to know the risk factors of heart disease in specific patients. According to T., the medical information of patients is Q = 58, 1, 4, 133, 196, 1. , 2, 1, 6i. In the model , SkNN Bob needs to encrypt Q before (to preserve the privacy of his search term) and send it to the cloud. The cloud then searches in an encrypted database, Epk (T), to locate nearby neighbors k with user requests. For simplicity, let's assume that k = 2. Under this case, the closest neighbors Q is t4 and t5 (using the Euclidean distance as a measure of similarity). After this, the clouds send t4 and t5 (in Encrypted format) to Bob. Here, the cloud should identify the nearest neighbor of Q without paying attention without getting to know any sensitive information. Like all calculations, it must be My secret code Eventually, Bob gets t4 and t5 to help him make a medical decision.

II. Related Work and Background

Due to space constraints, we will review existing work and provide some background definition. See our technical reports [5] for more relevant results and backgrounds. At first it looked like a system. Full cryomphomomorphic (such as [6]) can solve the DMED problem because it allows third parties. (Which hosts the encrypted data) will act as an arbitrary data encryption without decrypting. However, we emphasize that the technique is very expensive and that in-app use in practice has not been explored. For example, it was shown in [7] that even for one weak-security -bootstrapping parameter, the operation of a homomorphic operation would take at least 30 seconds on a high-performance machine. It is possible to use the secret sharing techniques available in SMC, such as Shamir's project [8], to develop PPkNN model . However, our work is different from the secret shared solution in the following areas. Secret-based solutions require at least three parties, while our work requires only two parties. For example, the construction is based on Sharemind, [9] which is a well-known framework of SMC, based on a secret sharing model, assuming that the number of participants is three. So our work follows the Sharemind plan and other secret sharing schemes.

2.1 Privacy-Preserving Data Mining

Agrawal and Srikant [10], Lindell, and Pinkas [11]. Initially, the concept of privacy-preservation under data mining techniques, PPDM techniques can be divided into 2 types (1), Data interference, and (2) data distribution. Agrawal and Srikant proposed the first data harassment technique to generate the classification of decision making and many other methods proposed later (eg [12], [13], [[14] However, The variant technique information cannot be used with semantically encrypted data. Also, it is not possible to generate accurate mining results because of the way noise is generated. Statistics into data. On the other hand, Lindell and Pinkas [11] proposed the first decision classifier under both parties' assumption that the data was distributed among the parties. Since then, a number of publications have been published using SMC techniques (eg, [15], [16], [17] We claim that the PPkNNproblem can not be modified using data distribution techniques. For the same reason, we will not consider a k-NN security method that distributes data between two parties (eg, [18]).

2.2 Query Processing over Encrypted Data

In recent years, researchers have proposed ways [1], [11] - [13] to address SkNN's problem. However, we emphasize that the existing SkNN method violates one or more of the properties. On the other hand, methods in [1], [11] are not secure because of the risk of selected and known plaintext attacks. On the other hand, the latest method in [13] would result in incorrect kNN affecting the end user. More precisely, in [13] the cloud will retrieve an encrypted partition that is associated instead of finding the nearest neighbor. The k-closest encrypted host also in [1], [12], [13] It involves heavy calculations during the query processing process.

However, we note that PPkNN is a more complex problem than simple kNN query processing through encrypted data [22], [23] For one nearest neighbor, the k-nearest classification process should not be disclosed to the cloud or to any user. [23] We emphasize that the most recent method in [23] K the neighbor closest to the user. Second, even though we know that k-neighbors are the closest. It is still difficult to find the most labels in these neighborhoods, as they were encrypted initially to protect the cloud from sensitive information. Third, the existing work does not address access issues, which is a significant privacy requirement from the user's point of view. In our recent work [24], we proposed a new secure k-neighbor neighbor query model for encrypted data to protect confidential information, user privacy, and data access hiding. However, as mentioned above, PPkNN is a more complex problem and cannot be solved directly using the k-nearest neighbor technique already contained in encrypted data. So in this article we will expand on the previous work [24] and provide a new approach to solving the problem of smuggling PPkNN data through encrypted data. In particular, this article differs from our preliminary work [24] in the following 4 points. First of all, in this article, we introduce new security principles, including minimum security (SMIN), minimum security from n numbers (SMINn), secure frequencies (SF), and the introduction of new solutions for them.

Secondly, work in [24] does not have an official security analysis of sub-model. On the other hand, this document contains the official security evidence of the referenced subroutine, including the PPkNN model, under a semi-straight format. We also discussed various techniques that the PPkNN model offers can be extended to secure model under malicious settings. Third, our preliminary work [24] will only identify kNN safe search terms, which are similar to Step 1 of PPkNN. However, Step 2 in PPkNN is entirely new. Finally, our empirical analysis in Section VI is based on the actual dataset, while the results in [24] are based on the simulations. In addition, new experimental results are included in this article.

.

Retrieval using K-nn

Calling the closest neighbors of k to the queue Q The Q is the most fundamental problem in many application domains, such as search, similarity, pattern recognition, and data mining. In literature, there are numerous techniques proposed to address the problem of SkNN, which can be classified into two categories according to whether the data is encrypted or not.

1) Centralized approach: In the centralized approach, we assume that the data owner outsources the database and DBMS functionality (such as kNN query) to an untrusted external provider that handles the data on behalf of the owner. Only trusted users are available. Allowed to query hosted information. Storing data to an untrusted server can cause security issues such as privacy of information. To obtain information that is private, the data owner needs to use a data erasure model (such as k-anonymityity) or cryptographic techniques. (Such as encryption and data interference) over their data before hiring them to the server.

Encryption is a traditional technique used to protect the confidentiality of sensitive information such as medical records. Because encoding the data, the process of evaluating a questionnaire through encrypted data becomes a challenge. In this direction, various techniques have been proposed for processing ranges [6] - [8] and aggregation queries [9], [10] through data encryption. However, in this document we limit our discussion to a safe search result for kNN. In recent years, researchers have proposed different methods [1], [11] - [13] To solve the problem of SkNN, Wong and his team [11] proposed a new cryptographic scheme called asymmetric cryptography (ASPM) that maintained the product level between the vector Q query and any vector of t from the database T. The data and search terms are encoded using a slightly different encoding scheme before they are outsourced to the server and all query users know the decryption key. Renovation Zhu et al. [12] proposed a new approach to SKNN, where the owner of the data was not disclosed to the user. However, their architecture requires the involvement of the owner of the information during the search query encoding. One alternative is Hu et al. [1]. It offers a method based on the stable Morse code encryption model [17] that supports the separation, deletion, and multiplication of data to be encrypted.

They discuss the SkNN problem under the following settings: The client has a ciphertext of all data points in the T database and its encryption function, while the server has a decryption function of T and some extra information about it.

Recently, Yao et al. [13] proposed a new method, SkNN, based on a partition using the Voronoi Safe Diagram (SVD). Instead of calling the cloud to retrieve the kNN, they certainly wanted it from the cloud to restore the partition. Epk (G) for Epk (T) so that G is guaranteed to have the closest k-neighbor of Q. However, in our work, we can solve the problem SKNN properly. Must be by check Let the neighboring neighbors call k-nearest Q's (in encrypted form). In addition, most computations in the message processing step [1], [12], [13] will Internal implementation by the end user conflicts with the primary purpose of outsourcing the DBMS functionality to the cloud. In addition, the model in [13] leaks data access patterns such as partition IDs that match user queries to the cloud.

2) Distribution method: In the data distribution method, data is assumed to be either vertical or horizontal partition and distributed between a set of independent maverick. In the literature, data distribution methods rely on a number of secure calculation techniques (SMCs), which allow multiple parties to safely evaluate their functions using their personal inputs without

revealing one party's information to another. Several attempts have been made to solve the problem of kNN queries in a distributed environment. Shaneck et al. [18] proposes a privacy approach to perform neighboring k-neighbor search model in [18]. There are several safe parts for the kNN calculation point, especially in the horizontal partitioned data series. Qi et al. [19] proposes a single, secure kNN search model that is sufficiently secure. Vaidya et al. [20]. A study of privacy in the storage of top-k search data, in which the information is categorized as vertical. Ghinita et al. [21]. We emphasize that in [21] the data residing on the server is in plaintext format. However, if the data is encrypted to ensure that the information is confidential. It is clear that the user can retrieve the output because he / she does not know the index that matches his / her query.

However, even if the user can retrieve the records using the PIR, the user still needs to perform an internal calculation. However, in our framework, user-based computing is completely outsourced to the cloud. In summary, we emphasize that the above distribution method does not work with kNN query data through encrypted data for two reasons: (1) In our work, we deal with the database encryption model and the query. (2) The database in our case is encrypted and stored in the cloud, while the above method is partitioned (in the form of plaintext) between groups.

III. Proposed Architecture

In this article, we present the new SkNN model to facilitate locating k-neighbor neighbors, close to encrypted data in the cloud that helps maintain data privacy and privacy of the query. In our model , when encrypted data was outsourced to the cloud, Alice did not participate in any computation, so no information would be shared with Alice, especially the proposed model . Meet the following requirements:

• Confidential Information - The contents of any T or any intermediate result should not be disclosed to the cloud.

• Query privacy - Do not disclose Bob's query Q in the queue.

• Correct - output ht 0 1, ..., t0 k I should disclose only with Bob. There is no information other than t 0 1, ..., t0 k should be revealed to Bob.

• Bob's Minimum Cost Calculation - After sending an encrypted search record to our cloud, Bob's model costs less than Bob's [1], [11] - [13]

• Hidden access patterns - Access patterns, such as records that correspond to neighboring neighbors. Q's closest neighbors should not be exposed to Alice and Cloud (to prevent invasive attacks).

.

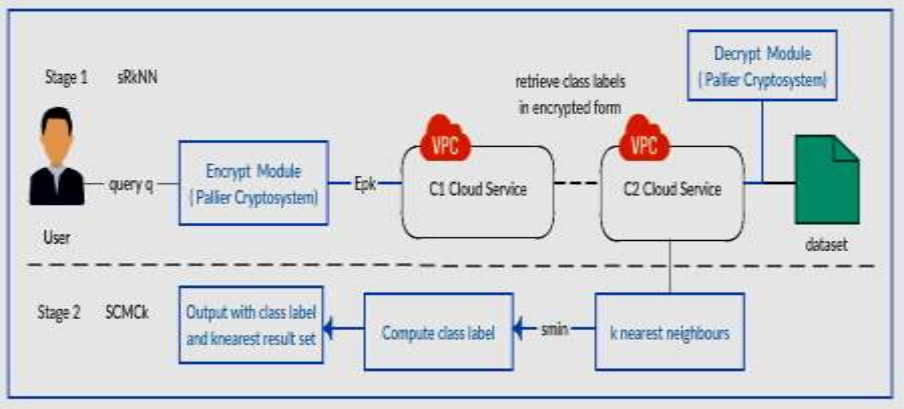

Figure1.0 Architectural Diagram

In the SkNN model , the presence of two non-semi-cloud providers, represented by C1 and C2, is combined into a cloud federation model. The assumption is not new and has been used in domain related problems. [16] For more details, see Section 3.6. Remember that the database of Alice contains n records, which are represented by T = ft1,...., tng and m where ti; j specifies the jth attribute value of the ti record. At first, Alice encrypts her database using the public key here. Under this setting, Alice stores the encrypted database (T0) at C1 and the secret key (sk). The encrypted database is represented by T0, assuming all attribute values and Euclidean distance are in [0; 2L) To C2, the goal of the proposed SkNN model is to retrieve the kth top of the query that closely matches the user's search query, effectively and safely.

Consider Bob looking for k records close to the record. Deep search q = hq1, ..., qmi by T0 in C1 Bob sent his query q (in encrypted form) to C1. After that, both C1 and C2 will be associated with a set of subroutines to retrieve. The k record set corresponds to the neighbor k nearest the query q input. At the end of the model , the proposed Bob only gets the nearest K-neighbors up to q as the output. In the underlying model specified by SkNNb, the desirable properties discussed in Section 1.2 are relaxed to create effective model s. The overall procedure involved with the SkNNb model is presented in Algorithm 8. At first, Bob encrypted his query (q) wisely. He computes Epk (q) = Epk (q1),...., EPK (QM). He then sends Epk (q) to C1.

Upon receiving Epk(q) from Bob, both C1 with private input Epk(q);Epk(ti) and C2 with the secret key sk jointly involve in the Secure Squared Euclidean Distance (SSED) model , where

Epk(ti) = Epk(ti;1),.....,Epk(ti ;m)
for 1 < i < n
*sqrEclDis = denoted by Epk(di), between q and ti, di = (jq * tij)2.*
Epk(di) is known only to C1
for 1< i< n.
compare the squared Euclidean distances as it preserves relative ordering.
C1 sends Epk(d1) till Epk(dn) o to C2,
Upon receiving Epk(d1) till Epk(dn)
C2 decrypts the encrypted distance in each entry to obtain
di = Dsk Epk(di)
C2 then generates an class label list = hi1,...., hin which are the top k smallest distances among hd1,.....,hdn After this, C2 sends list to C1.

PPkNN- algorithm

1. User encrypts his query q attribute wise, sends encrypted query about C1
2. C1 and C2:
   (a) C1 receives encrypted query and applies SSED model
   (b) C2 calculates the encryption applying SM model
3. C1 and C2:
   (a) C1 and C2 iteratively measures encryption compared to k-nearest neighborhood of q
   (b) C1: At the end of the last iteration, only C1 knows the encryption key, sends to C2.
   (c) C2: Decrypts encrypted vectors and send to C1
   (d) C1: C1 Receives from C2 and performs inverse permutations
   (e) Build Squared Difference Map at C1 and update distance vectors with help of C2
4. Applying SCMC model C1 and C2 simultaneously computes KNN of q..

Psuedocode for Proposed Methodology

1) Initiliaze Pallier Cryptosystem Key On C1
2) Obtain User Input Parameters and calculate summation of obtained data values on C1
3) Perform Squaring of obtained summation value on C1
4) Obtain Encypted value of plain summation squaring on C1
5) Generate Random Number Rn on C1
6) Obtain randomized sum by processing record dataset attributes with random number Rn on C2
7) Perform Secure Squaring of Random Number and Encrypt the resultant value from C2
8) Perform Secure Multiplication with RandomSum and Randomized Dataset Attributes from C2
9) Obtain Absolute Value of Secure Product to obtain Euclidean distance between two tuples From C2
10) Min Distance signifies maximum similarity of tuples, add top 5 record's (enc) label values to the list from C2
11) Obtain Squared Difference Map of Data Value and Class Label and return the list to C1.
12) Return Result in decrypted for m to the user.

We emphasize that the intermediate results visible from the clouds in our model are random or newly created encodings. Therefore, the data recorded corresponds to the neighbors closest neighbors of Q are not known to the cloud. Also after sending an encrypted search record to the cloud, Bob did not participate in any calculation (low cost on Bob), so the access pattern was further protected by Bob.

Secure Multiplication (SM) Model :

This model considers P1 with inputs (Epk (a), Epk (b)), and the output of Epk (a * b) to P1, where A and B are not known in P1 and P2. A and B are not present in P1 and P2. The output of Epk (a * b) is known only to P1.

Secure Proximity Euclidean Distance (SSED) Model : P1 with inputs (Epk (X), Epk (Y)) and P2 Calculate the encryption of Euclidean distance in X and Y safely. Here X And Y is the dimension vector m at Epk (X) = hEpk (x1),..., Epk (xm) i and Epk (Y) = hEpk The Epk output (| X - Y | 2) is known only to P1.

• **Secure Bit-Decomposition (SBD) Model :** P1 with the input of Epk (z) and P2 can compute the encoding of each z-bit, where $0 \leq z < 2$ l results [z] = hEpk (z1), ..., Epk (zl) i is only known with P1. Here z1 and z denote the greatest and least significant of the integers z, respectively.

• **Minimal Model (SMIN):** P1 with inputs ([u], [v] and P2 with sk. Calculates the encryption of each bit of the minimum number between u and v. That is, The output is [min (u., V)], which is known only to P1. During this model , no information about u and v is disclosed to P1 and P2.

• **Minimum security of n number (SMINn) model:**

P1 has n encoded vectors ([d1], .., [dn]) and P2 has sk here [di] = hEpk (di, 1),..., Epk , 1 and di, l are the smallest and least significant of the integers di respectively. For $1 \leq i \leq n$, P1 and P2 will calculate the output [min (d1,., Dn)] at the end [min (d1,., dn)] is known only to P1 between SMINn. Information about di is not disclosed in P1 and P2.

IV. Experimental Results

For our tests, we used a car assessment kit from UCI KDD [25]. Record 1728 (eg n = 1728) and 6 attributes (such as m = 6). There are also separate class attributes. Data sets are divided into four different classes (eg, w = 4). We encrypt this data set. By using the Paillier encoding, which is important in our experiments, and the encrypted data is stored locally on our machine from our PPkNN model , we conducted a random query over this encrypted data. For the rest of this topic, we're not talking about Alice's performance because it's a one-time cost. However, we evaluate and analyze the representation of these two steps in PPkNN separately.

First of all, we calculated the cost calculation of Step 1 in PPkNN for different k neighbors.
In addition, the Paillier K encryption keys are 512 or 1024 bits. The results are shown in Figure 2 (a).
For K = 512 bits, the calculation cost of step 1 changes from 5.73 to 46.20 minutes when k changes from 5.
Is 25, respectively. On the other hand, when K = 1024 bits, the calculation cost of Phase 1 will vary from 66.97. It is 309.98 minutes when k turns from 5 to 25, respectively. In both cases, we notice that the cost
Of phase 1 grows near the straight line with k for K. We've identified that the cost of step 1 increases.
By almost 7 factors, when k is twice as eg when k = 10, phase 1 takes 11.46 and 102.4 minutes.
Creates encrypted layer labels of the nearest 10 neighbors under K = 512 and 1024 bits, respectively.
Also, when k = 5, we find that approximately 57.30% of the cost in Phase 1 is due to SMINn.
There are k starts in PPkNN (once in each round). Also, the costs incurred due to SMINn increase from
57.30 to 79.18%, when k increases from 5 to 25

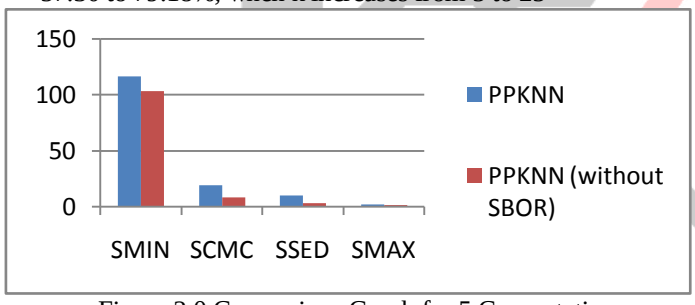

Figure 2.0 Comparison Graph for 5 Computations

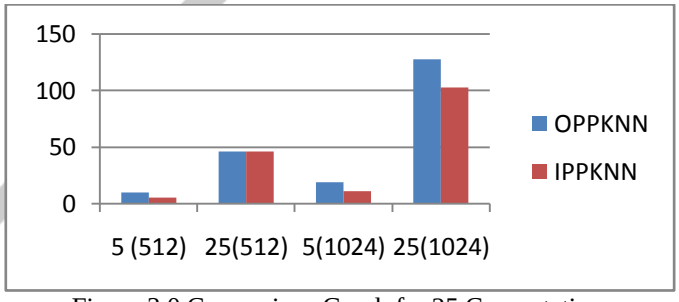

Figure 3.0 Comparison Graph for 25 Computations

It is clear that the computational cost of Step 1 is significantly higher than Step 2 in PPkNN. In particular, we observe that the computation time of step 1 is at least 99% of the total time in the PPkNN. For example, when k = 10 and K = 512 bits, the calculation cost of Steps 1 and 2 is 5.73. Min. Under this scenario, the cost of step 1 is 99.98% of the total cost of the PPkNN. We also note that the total computation time of the PPkNN will increase almost linearly with n and k.

On the other hand, Bob's computing costs in PPkNN are largely due to encoding his input data. In Bob's calculation, the cost of computing was 8.24 milliseconds, when K was 512 bits, respectively. Obviously, PPkNN was very efficient from Bob's computational perspective. Resources with limited resources.

V. Limitation & Future Scope

The proposed PPkNN model is not very effective without the use. parallelization However, our work is the first to offer a secure PPkNN solution in a semi-accurate format. Due to the increased demand for data mining as a service in cloud computing, we believe our work will benefit the cloud community to further stimulate research. Say Hopefully there will be further development of the PPkNN approach. (By optimizing model or finding alternatives) in the near future.

VI. Conclusion

In order to protect the privacy of users, personal data privacy techniques have been introduced over the past decade. Existing techniques do not apply to data that is encrypted on a third-party server. This article presents the k-NN classification model for securing personal data using cloud encrypted data. Our model protect the confidentiality of user input and stealth data. We also evaluate the performance of our model under various parameter settings. Since SMINn's performance improvement is an important first step for improving our PPkNN model performance, we are planning to explore more effective and alternative solutions to SMINn issues in our future work. In addition, we will examine and expand our research into other classification algorithms.

References

[1] P. Mell and T. Grance, "The nist definition of cloud computing (draft)," NIST special publication, vol. 800, p. 145, 2011.

[2] S. De Capitani di Vimercati, S. Foresti, and P. Samarati, "Managing and accessing data in the cloud: Privacy risks and approaches," in CRiSIS, pp. 1 –9, 2012.

[3] P. Williams, R. Sion, and B. Carbunar, "Building castles out of mud: practical access pattern privacy and correctness on untrusted storage," in ACM CCS, pp. 139–148, 2008.

[4] P. Paillier, "Public key cryptosystems based on composite degree residuosity classes," in Eurocrypt, pp. 223–238, 1999.

[5] B. K. Samanthula, Y. Elmehdwi, and W. Jiang, "k-nearest neighbor classification over semantically secure encrypted relational data." eprint arXiv:1403.5001, 2014.

[6] C. Gentry, "Fully homomorphic encryption using ideal lattices," in ACM STOC, pp. 169–178, 2009.

[7] C. Gentry and S. Halevi, "Implementing gentry's fullyhomomorphic encryption scheme," in EUROCRYPT, pp. 129–148, Springer, 2011.

[8] A. Shamir, "How to share a secret," Commun. ACM, vol. 22, pp. 612–613, Nov. 1979.

[9] D. Bogdanov, S. Laur, and J. Willemson, "Sharemind: A framework for fast privacy-preserving computations," in ESORICS, pp. 192–206, Springer, 2008.

[10] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in ACM Sigmod Record, vol. 29, pp. 439–450, ACM, 2000.

[11] Y. Lindell and B. Pinkas, "Privacy preserving data mining," in Advances in Cryptology (CRYPTO), pp. 36–54, Springer, 2000.

[12] P. Zhang, Y. Tong, S. Tang, and D. Yang, "Privacy preserving naive bayes classification," ADMA, pp. 744–752, 2005.

[13] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, "Privacy preserving mining of association rules," Information Systems, vol. 29, no. 4, pp. 343–364, 2004.

[14] R. J. Bayardo and R. Agrawal, "Data privacy through optimal k-anonymization," in IEEE ICDE, pp. 217–228, 2005.

[15] H. Hu, J. Xu, C. Ren, and B. Choi, "Processing private queries over untrusted data cloud through privacy homomorphism," in IEEE ICDE, pp. 601–612, 2011.

[16] M. Kantarcioglu and C. Clifton, "Privately computing a distributed k-nn classifier," in PKDD, pp. 279–290, 2004.

[17] L. Xiong, S. Chitti, and L. Liu, "K nearest neighbor classification across multiple private databases," in CIKM, pp. 840–841, ACM, 2006.

[18] Y. Qi and M. J. Atallah, "Efficient privacy-preserving k-nearest neighbor search," in IEEE ICDCS, pp. 311–319, 2008.

[19] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preserving encryption for numeric data," in ACM SIGMOD, pp. 563– 574, 2004.

[20] H. Hacig¨um¨ us¸, B. Iyer, C. Li, and S. Mehrotra, "Executing sql over encrypted data in the database-service-provider model," in ACM SIGMOD, pp. 216–227, 2002.

[21] B. Hore, S. Mehrotra, M. Canim, and M. Kantarcioglu, "Secure multidimensional range queries over outsourced data," The VLDB Journal, vol. 21, no. 3, pp. 333–358, 2012.

[22] W. K. Wong, D. W.-l. Cheung, B. Kao, and N. Mamoulis, "Secure knn computation on encrypted databases," in ACM SIGMOD, pp. 139–152, 2009.

[23] X. Xiao, F. Li, and B. Yao, "Secure nearest neighbor revisited," in IEEE ICDE, pp. 733–744, 2013.

[24] Y. Elmehdwi, B. K. Samanthula, and W. Jiang, "Secure k- nearest neighbor query over encrypted data in outsourced environments," in IEEE ICDE, pp. 664–675, 2014.

[25] M. Bohanec and B. Zupan. The UCI KDD Archive, 1997. http: //archive.ics.uci.edu/ml/datasets/Car+Evaluation.