Mining Multiple Private Databases Using a kNN Classifier *

Li Xiong
Emory University
lxiong@mathcs.emory.edu

Subramanyam Chitti, Ling Liu
Georgia Institute of Technology
chittis, lingliu@cc.gatech.edu

ABSTRACT

Modern electronic communication has collapsed geographical boundaries for global information sharing but often at the expense of data security and privacy boundaries. Distributed privacy preserving data mining tools are increasingly becoming critical for mining multiple databases with a minimum information disclosure. We present a framework including a general model as well as multi-round algorithms for mining horizontally partitioned databases using a privacy preserving k Nearest Neighbor (kNN) classifier. A salient feature of our approach is that it offers a trade-off between accuracy, efficiency and privacy through multi-round protocols.

Categories and Subject Descriptors: H.2.8 [Database Management]: Database Applications

Keywords: Privacy, k Nearest Neighbor, Classification, Distributed Databases

1. INTRODUCTION

The information age has enabled many organizations to collect large amounts of data. Privacy-preserving data mining [26] becomes an important enabling technology for mining data from multiple private databases provided by different and possibly competing organizations.

Motivating Scenarios Classification plays an important role in data mining and kNN classification is one type of lazy classification algorithm that offers many advantages. The need for building such a classifier across multiple private databases is driven by applications from various domains. For example [6], many insurance companies collect data on disease incidents, seriousness of the disease and patient background. One way for the Center for Disease Control to identify disease outbreaks is to train a classifier across the data held by the various insurance companies for patterns that are indicative of disease outbreaks and use it to classify a query pattern as an outbreak or the opposite. However, commercial and legal reasons prevent the insurance companies from revealing their data. It is important and beneficial to have a distributed data mining algorithm that is capable of identifying potential outbreaks while respecting the privacy requirements of its participants.

In industry collaboration, industry trade groups want to identify best practices to help members, however, some practices may be trade secrets. We would like to discover patterns (like "manufacturing using chemical supplies from supplier X have high failure rates") while preserving secrets of individual organizations (like "manufacturing process Y gives low failure rates").

Research Challenges and Design Goals. One solution to privacy preserving data classification across multiple private databases is to have a trusted third party (TTP). The nodes send their data along with the query to the TTP, which constructs the classifier using the data, classifies the query and sends the results back to all nodes. However, in practice it is difficult to find a TTP which is trusted by all nodes. If the TTP is compromised, the privacy of all nodes is compromised.

Another possibility one might consider is to construct a privacy preserving classifier using secure multiparty computation techniques [11, 10], a subject that has received significant attention in cryptography research. However, these techniques have a high communication overhead and are feasible only if the inputs are very small. This deficiency of secure multiparty schemes has led to the search for efficient, privacy preserving data mining algorithms.

We identify three important dimensions that we should consider when designing a privacy preserving classification algorithm, namely, *accuracy, efficiency, and privacy*. Ideally, we would like the algorithm to have a comparable accuracy to its non-privacy preserving counterpart, and an absolute privacy wherein no information other than the trained classifier and the classification of the query instance should be revealed to any node. At one end of the spectrum, we have the non-privacy preserving classifier algorithms, which are highly efficient but are not secure. At the other end, we have the secure multi-party computation protocols [10], using which we can construct classifiers which are provably secure in the sense that they reveal the least amount of information and have the highest accuracy; but are very inefficient. Our design goal is to look for algorithms that can provide a desired level of tradeoff between the accuracy of the classifier constructed and the stringency of the privacy requirements while maintaining efficiency.

Thinking of the design space in terms of these three dimensions presents many advantages. At one end of the spectrum, we have the non-privacy preserving classifier algorithms, which are highly efficient but are not secure. At the other end, we have the secure multi-party computation protocols, using which we can construct classifiers which are provably secure in the sense that they reveal the least amount of information and have the highest accuracy; but are very inefficient. Our design goal is to look for algorithms that can provide a desired level of tradeoff between the accuracy of the classifier constructed and the stringency of the privacy requirements while maintaining efficiency.

Contributions. With these design objectives in mind, we present a privacy-preserving framework, Private$k$NN, for constructing a kNN classifier across multiple horizontally partitioned private databases. This framework consists of a general model for kNN classification across private databases (Section 2) and a set of concrete algorithms for realizing this model (Section 3). We discuss how well our algorithm achieves the specified requirements as demonstrated by an extensive experimental evaluation (Section 4). To the best of our knowledge, this is the first paper to show how kNN classification can be achieved for horizontally partitioned private data without a centralized trusted third party. Our approach has an important trait − it offers a trade-off between accuracy, efficiency and privacy, allowing our privacy-preserving kNN classifier to be applied in a variety of problem settings and meeting different optimization criteria.

2. THE MODEL

In this section, we describe the kNN classification problem and discuss how we can solve it in a distributed, privacy preserving manner.

Problem Definition and Threat Model Consider n private databases distributed at n different nodes where all databases have the same schema, i.e. data is horizontally partitioned. We consider the problem where the nodes want to train a kNN classifier on the union of their databases while revealing as little information as possible to the other nodes during the construction of the classifier (*training* phase) and the classification of a new query (*test* phase).

We assume a semi-honest model for the participating nodes in the sense that they correctly follow the protocol specification, yet attempt to learn additional information about other nodes by analyzing the transcript of messages received during the execution of the protocol. One of our ongoing efforts is to develop a decentralized kNN classification protocol that is resilient against malicious nodes.

k**NN Classification Model** To solve the kNN classification problem, we need to adapt the basic distance weighted kNN classification algorithm to work in a distributed setting in a privacy preserving manner. The central thesis of our proposed model is to divide the problem into two sub-problems, and to ensure that each step is accomplished in a privacy preserving manner.

Given a query instance (point), a kNN classifier uses the k nearest neighbors of the query to classify it. In a distributed setting, the k nearest neighbors of a query point could be distributed among the n nodes. Thus, each node will have some points in its database which are among the k nearest neighbors of the query point. So, for a node to calculate its local classification of the query point, it has to first determine which points in its database are among the k

nearest neighbors of this query point. Then, it can calculate its classification of the query instance. Finally, the nodes need to combine their local classifications, in a privacy preserving manner, to compute the global classification of the query point over the n private databases. Thus we can divide the kNN classification problem into the following two sub-problems.

1. **Nearest neighbor selection:** Given a query instance x to be classified, the databases need to identify all points that are among the k nearest neighbors of x in a privacy preserving manner.

2. **Classification:** Each node calculates its local classification of x and then cooperate to determine the global classification of x in a privacy preserving manner.

It is important to note that if executed naively, the above steps can violate the privacy requirements of the individual databases. Given a query point x, we should ensure that the instances in a database are not revealed to other databases in the nearest neighbor selection, and that the local classification of each database is not revealed to other databases during global classification.

3. THE ALGORITHM

In this section, we present our protocol setting and concrete algorithms for realizing the classification model. In the protocol, nodes are mapped onto a ring randomly; thus each node has a predecessor and successor. An initialization module is designed to select the starting node among the n participating nodes, and initialize a set of parameters used in the local computation algorithms. The nodes then engage in a multi-round protocol by executing a local algorithm and communicates the result of this algorithm to its successor.

Nearest Neighbor Selection In order to determine the points in their database that are among the k nearest neighbors of x, each node calculates k smallest distances between x and the points in their database (locally) and then we can use a privacy preserving algorithm to determine k smallest distances between x and the points in the union of the databases or kth nearest distance (globally). We can assume that the distance is a one-way function so that nodes do not know the exact position of each other node by distance. There has been privacy preserving algorithms recently proposed [1] for finding kth element that we can use for implementing this step. Although information-theoretically secure, it is still computationally expensive. In this paper, we adapt the multi-round top$k$ algorithm proposed in [29] to determine k smallest distances before determining kth smallest distance and achieve a tradeoff between accuracy, efficiency and privacy.

Algorithm 1 shows a sketch of the multi-round privacy preserving nearest distance selection protocol used in the nearest neighbor selection step. The key to this protocol is the design of the local algorithm which is a probabilistic algorithm that injects certain amount of randomization, such that the probability of data value disclosure at each node is minimized while the eventual result of the protocol is guaranteed to be correct. For detailed description of the local algorithm, please refer to [29].

Classification After each node determines the points in its database which are within the kth nearest distance from x, each node computes a local classification vector of the

Algorithm 1 Selection of k Nearest Distances

Input: ldv_i, local k nearest distance vector between each node and query intance x
Output: gdv, global k nearest distance vector between all nodes and query intance x

- The starting node initializes a global distance vector (gdv_i) of length k where each element is initialized to be the maximum possible distance between any query and any data point, and sends it to its successor.

- Every node i, upon receiving a vector gdv_{i-1} from its predecessor, uses a randomized algorithm to compute gdv_i and transmits it to its successor.

- Repeat the above step for r rounds

- When the starting node receives a vector $gdv_i(r)$ from its predecessor, it broadcasts it to all the nodes.

---

query instance where the ith element is the amount of vote the ith class received from the points in this node's database which are among the k nearest neighbors of x. The nodes then participate in a privacy preserving term-wise addition of these local classification vectors to determine the global classification vector. For the term-wise addition, we use the multi-round privacy-preserving addition protocol suggested in [7].

Algorithm 2 shows a sketch of the single-round privacy preserving addition protocol used in the classification step. Once each node knows the global classification vector, it can find the class with the global majority of the vote by determining the index of the maximum value in the global classification vector.

Algorithm 2 Addition of Local Classification Vectors

Input: lcv_i, the local classification vectors of each node
Output: gcv, global classification vector ($gcv = \sum_{i=1}^{n} lcv_i$)

- The starting node initializes a global classification vector (gcv) to a random vector rv, and sends it to its successor.

- Every node, upon receiving a vector vi from its predecessor, transmits $lcv_i + vi$ to its successor.

- When the starting node receives a vector vi from its predecessor, it broadcasts ($vi - rv + lcv$) to all the nodes.

Complete Algorithm Putting things together, Algorithm 3 shows an integrated solution, Private$k$NN, that builds a kNN classifier across multiple private databases.

4. EXPERIMENTAL EVALUATION

We conducted a formal analysis as well as an experimental evaluation on the proposed algorithms in terms of its correctness, efficiency, and privacy characteristics. We omit the analytical results in this paper due to the space limitations. We present the experimental results in this section that show the relative accuracy of the proposed algorithm

---

Algorithm 3 kNN Classification

Input: x, an instance to be classified
Output: $classification(x)$, classification of x

- Each node computes the distance between x and each point y in its database, $d(x, y)$, selects k smallest distances (locally), and stores them in a local distance vector ldv.

- Using ldv as input, the nodes use the privacy preserving nearest distance selection protocol 1 to select k nearest distances (globally), and stores them in gdv.

- Each node selects the kth nearest distance Δ: $\Delta = gdv(k)$.

- Assuming there are v classes, each node calculates a local classification vector lcv for all points y in its database: $\forall 1 \le i \le v$, $lcv(i) = \sum_y w(d(x,y)) * [f(y) == i] * [d(x,y) \le \Delta]$, where $d(x,y)$ is the distance between x and y, $f(y)$ is the classification of point y, and $[p]$ is a function that evaluates to 1 if the predicate p is true, and 0 otherwise.

- Using lcv as input, the nodes use the privacy preserving classification protocol **??** to calculate the global classification vector gcv.

- Each node assigns the classification of x as $classification(x) \leftarrow \arg\max_{i \in V} gcv(i)$.

---

as compared to a distributed, ordinary kNN classification algorithm and its sensitivity to various parameters.

4.1 Experimental Setup

We use three publicly available datasets in our experiments. The first dataset, GLASS [9], contains data regarding various physical characteristics of different types of glass. The classification problem is to identify the type of glass from its physical characteristics. The study of classification of types of glass was motivated by criminological investigation - at the scene of a crime, the glass left behind can be used as evidence if it is correctly identified. This data set contains 214 instances belonging to 7 different classes, with each instance having 9 attributes. The second dataset, PIMA [21], is a medical dataset used for diagnostic purposes - for predicting whether a patient shows signs of diabetes given data like the 2-hour serum insulin concentration and Body Mass Index. This dataset contains 768 instances belonging to 8 different classes, with each instance having 8 different attributes. The third dataset, ABALONE [28], was used to predict the age of abalone from its physical characteristics. This data set contains 4177 instances belonging to 29 different classes, with each instance having 8 attributes.

In each experiment, we performed 100 separate runs on each different dataset. In each run, we randomly partitioned the data into two parts - a training set containing $\frac{3}{4}$ of the data and a test set containing $\frac{1}{4}$ of the data. The results reported are averaged over the 100 runs of each experiment. We summarize the notation we use to describe the experimental results in Table 4.1.
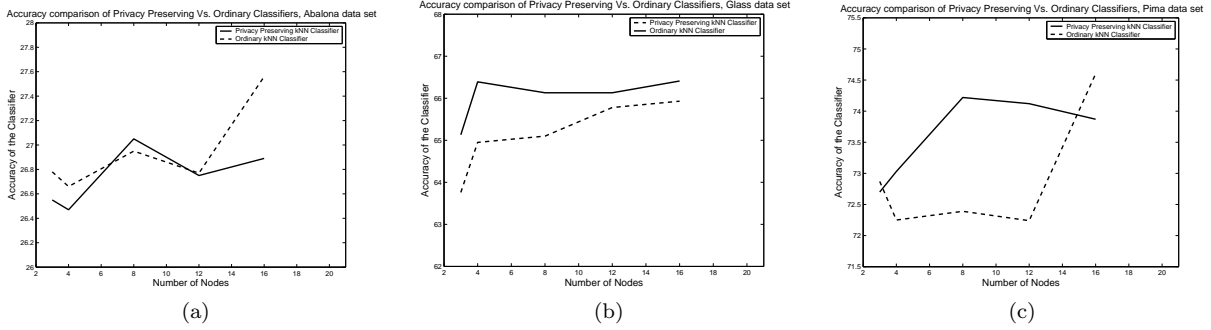
4.2 Accuracy

Figure 1: Accuracy Comparison of Private$k$NN vs. Distributed kNN Classifier
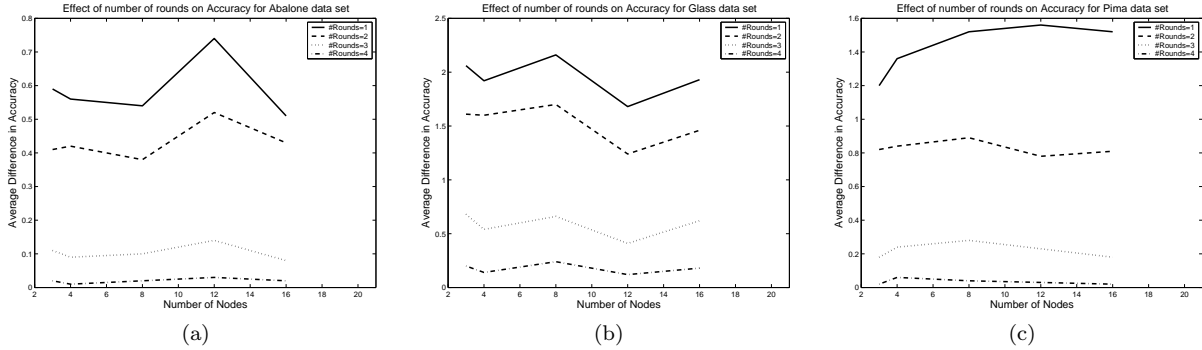

Figure 2: Relative Accuracy with Varying Number of Rounds in Nearest Neighbor Selection

Table 1: Experiment Parameters

Param.	Description
n	# of nodes in the system
k	kNN parameter
P_0	initial randomization prob. in neighbor selection
d	dampening factor in neighbor selection
r	# of rounds in neighbor selection

In this set of experiments, we compare the accuracy of Private$k$NN against a distributed kNN classifier. Of the two steps in our classifier model, the first step, nearest neighbor selection, is probabilistic in nature while the second step, classification, is deterministic and provably accurate. Thus, the overall accuracy of our classifier is determined by the accuracy of the nearest neighbor selection step. First, we would like to determine the accuracy of our classifier when it is optimized for efficiency. Then we study how the algorithmic parameters in the neighbor selection step as well as the parameter k affect the accuracy of our algorithm at the cost of efficiency.

In the first experiment, we run the nearest neighbor selection step for one round, with the initial probability $P_0 = 1$ and the randomization factor $d = 0.5$. With these parameter values, our algorithm is very efficient with communication complexity only 3 times that of an ordinary, distributed kNN classifier. We notice from Figure 1 that although we have chosen very conservative settings, the accuracy obtained by our classifier is still very high as compared to an ordinary classifier, and even higher in some of the cases. This is not a contradiction but rather an interesting insight, that an incorrect result of the nearest neighbor selection step could actually produce a value of k for which the kNN algorithm performs reasonably well or even better. This experiment indicates that our algorithm matches the accuracy of a distributed, ordinary kNN classifier without sacrificing its efficiency.

Effect of Number of Rounds. In this experiment, we verify whether the accuracy of the Private$k$NN classifier approaches that of the distributed kNN classifier when we run the nearest distance selection protocol (Algorithm 1) for a larger number of rounds. To do this, we measure the *absolute* value of the difference between the accuracies of the two classifiers when trained and tested on the same data. The results of our experiments are presented in Figure 2. It confirmed that we are able to make the classifier as accurate as an ordinary classifier by running nearest distance selection for a larger number of rounds.

Effect of Randomization Factor. In this experiment, we investigate the effect of varying the amount of randomization inserted into the computation in each round on the accuracy of the classifier. We do this by varying the value of d, the randomization factor. We note that if d is large, this means that more randomization is inserted into the computations, thus increasing the privacy guarantees of the algorithm. A very low value of d makes the computations almost deterministic and thus provides almost no privacy guaran-
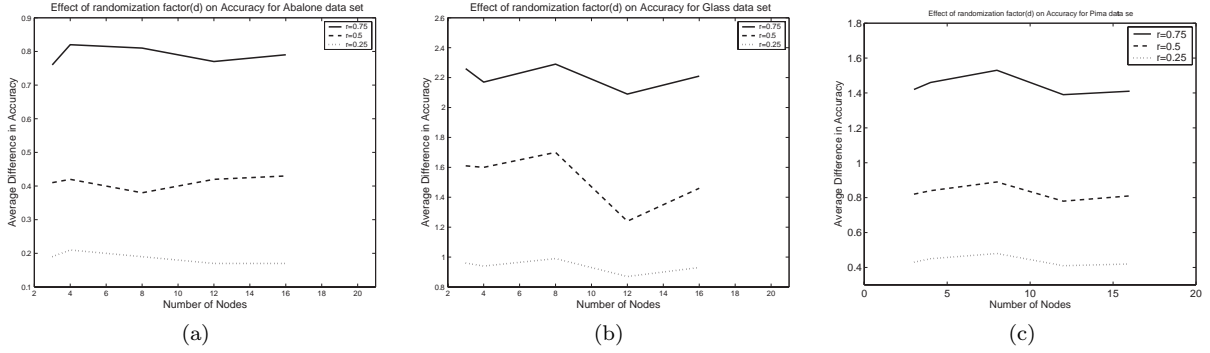
Figure 3: Relative Accuracy with Varying Randomization Parameter in Nearest Neighbor Selection
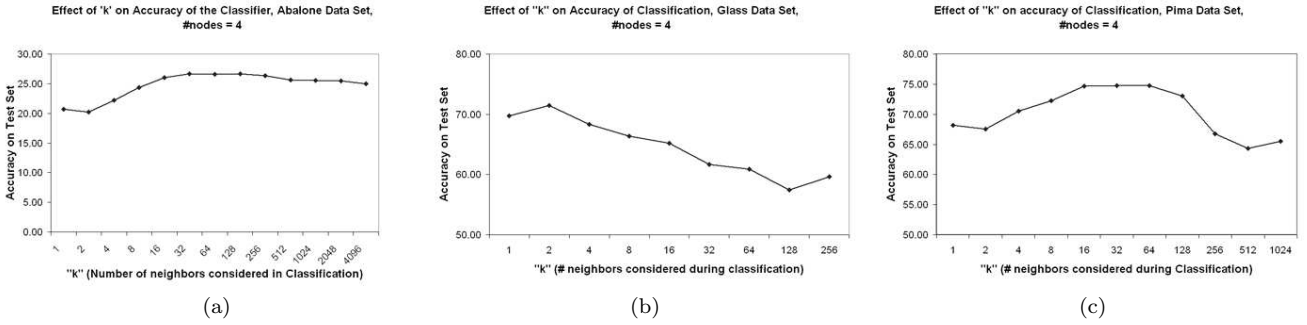

Figure 4: Relative Accuracy with Varying k

tees. The results of our experiments are shown in Figure 3. We note that with decreasing values of d, the accuracy of Private$k$NN becomes closer to that of the ordinary classifier. This is because smaller values of d increase the probability that the output of the nearest neighbor selection step is correct, and thus increase the probability of the two classifiers performing identical classification.

Effect of k. In this experiment, we investigate the effect of k, the number of neighbors considered during classification, on the accuracy of both Private$k$NN and ordinary kNN classifiers. We report the average accuracy when we used different values of k for classification in Figure 4. We note that the accuracy of the classifier decreases when k is very large. If we recall, we discussed in Section 2 that setting k equal to all the points in the union of the databases results in a kNN classification algorithm which has very good privacy guarantees. So, we would like k to be large to avoid any information leak due to our model. From the graphs, we note that it is possible to pick a k to achieve both good accuracy and good privacy.

4.3 Privacy

In this section, we study the information leak of our algorithm. Since the classification step is proven to be secure, we measure the distance information revealed during the nearest neighbor selection step. In this experiment, we run the protocol for $r = 2$ rounds, with $P_0 = 1$ and $d = 0.5$. We measure the probability that a node is able to correctly identify a distance value in the global vector it received from its predecessor as belonging to its predecessor. We present these results in Figure 5. We observe that for all values of n,

there is a large range of k such that the probability of a node revealing information to its successor is less than half. With very large values of k however, a node has a higher probability of inserting its values in the global vector and this increases its chances of revealing its values to its successor.

Our experiments indicate that even if we run the algorithm for a larger number of rounds, and use a range of values for the randomization factor d, the probability that a node reveals its values to its successor is still very low. However, space restrictions prevent us from presenting the graphs for these results.

5. RELATED WORK

Privacy related problems in databases have been an active and important research area. Research in secure databases, Hippocratic databases and privacy policy driven systems [15, 4, 2] has been focused on enabling access of sensitive information through centralized role-based access control. In database outsourcing and privacy preserving data analysis and publishing [12, 13, 26, 5, 8], the main approach is to use data generalization, data perturbation, and data partitioning and binning techniques to minimize the disclosure of precise information. In particular, there have been research focused on privacy preserving data transformation in the context of classification problems [14, 19].

The approach of protecting privacy of distributed sources was first addressed by the construction of decision trees [20]. This work closely followed the traditional secure multiparty computation approach and achieved perfect privacy. There has since been work to address association rules [22, 16],
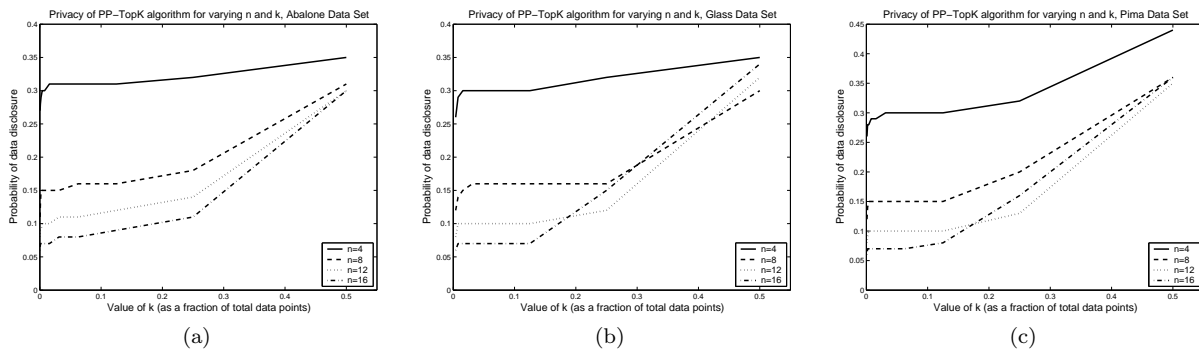
Figure 5: Privacy with Varying n and k

naive Bayes classification [18, 24, 30], and k-means clustering [23]. As a recent effort, there is also research on privacy preserving top$k$ queries [25] and privacy preserving distributed k-NN classifier [17], both across vertically partitioned data using k-anonymity privacy model. Agrawal et al. [3] also introduced the paradigm of minimal information sharing in information integration domain and proposed a privacy preserving join protocol between two parties. A few specialized protocols have been proposed, typically in a two party setting, e.g., for finding intersections [3], and kth ranked element [1]. [27] studied the problem of integrating private data sources with vertically partitioned data while satisfying k-anonymity of the data.

In contrast, our protocol computes kNN classifier selection across horizontally partitioned data. It leverages the multiparty network ($n > 3$) and utilizes a probabilistic multiround scheme to achieve minimal information disclosure and minimal overhead.

6. CONCLUSION

In this paper, we have tackled the problem of constructing a k-Nearest Neighbor classifier across horizontally partitioned private databases. Our work continues on thoroughly analyzing the efficiency and privacy properties of the algorithms under various circumstances such as repeated classifications.

7. REFERENCES

[1] G. Aggarwal, N. Mishra, and B. Pinkas. Secure computation of the kth ranked element. In *IACR Conference on Eurocrypt*, 2004.

[2] R. Agrawal, P. Bird, T. Grandison, J. Kieman, S. Logan, and W. Rjaibi. Extending relational database systems to automatically enforce privacy policies. In *ICDE*, 2005.

[3] R. Agrawal, A. Evfimievski, and R. Srikant. Information sharing across private databases. In *ACM SIGMOD Conference*, 2003.

[4] R. Agrawal, J. Kieman, R. Srikant, and Y. Xu. Hippocratic databases. In *International Conference on Very Large Databases (VLDB)*, 2002.

[5] E. Bertino, B. Ooi, Y. Yang, and R. H. Deng. Privacy and ownership preserving of outsourced medical data. In *ICDE*, 2005.

[6] C. Clifton. Tutorial on privacy, security, and data mining. In *13th European Conference on Machine Learning and 6th European Conference on Principles and Practice of Knowledge Discovery in Databases*, 2002.

[7] C. Clifton, M. Kantarcioglu, X. Lin, J. Vaidya, and M. Zhu. Tools for privacy preserving distributed data mining. In *SIGKDD Explorations*, 2003.

[8] J. Gehrke. Models and methods for privacy-preserving data analysis and publishing. In *ICDE*, 2006.

[9] B. German. In *ftp://ftp.ics.uci.edu/pub/machine-learning-databases/glass*.

[10] O. Goldreich. Secure multi-party computation, 2001. Working Draft, Version 1.3.

[11] S. Goldwasser. Multi-party computations: past and present. In *ACM Symposium on Principles of Distributed Computing (PODC)*, 1997.

[12] H. Hacigumus, B. Iyer, C. Li, and S. Mehrotra. Executing sql over encrypted data in the database service provider model. In *ACM SIGMOD Conference*, 2002.

[13] B. Hore, S. Mehrotra, and G. Tsudik. A privacy-preserving index for range queries. In *ACM Symposium on Principles of Distributed Computing (PODC)*, 1997.

[14] V. S. Iyengar. Transforming data to satisfy privacy constraints. In *SIGKDD*, 2002.

[15] S. Jajodia and R. Sandhu. Toward a multilevel secure relational data model. In *ACM SIGMOD Conference*, 1991.

[16] M. Kantarcioglu and C. Clifton. Privacy preserving data mining of association rules on horizontally partitioned data. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 16(9), 2004.

[17] M. Kantarcioglu and C. Clifton. Privacy preserving k-nn classifier. In *ICDE*, 2005.

[18] M. Kantarcoglu and J. Vaidya. Privacy preserving naive bayes classifier for horizontally partitioned data. In *IEEE ICDM Workshop on Privacy Preserving Data Mining*, 2003.

[19] K. LeFevre, D. DeWitt, and R. Ramakrishnan. Workload-aware anonymization. In *SIGKDD*, 2006.

[20] Y. Lindell and B. Pinkas. Privacy preserving data mining. *Journal of Cryptology*, 15(3), 2002.

[21] V. Sigillito. Pima. In *ftp://ftp.ics.uci.edu/pub/machine-learning-databases/pima-indians-diabetes*.

[22] J. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In *SIGKDD*, 2002.

[23] J. vaidya and C. Clifton. Privacy-preserving k-means clustering over vertically partitioned data. In *SIGKDD*, 2003.

[24] J. Vaidya and C. Clifton. Privacy preserving naive bayes classifier for vertically partitioned data. In *SIGKDD*, 2003.

[25] J. Vaidya and C. Clifton. Privacy-preserving top-k queries. In *ICDE*, 2005.

[26] V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis. State-of-the-art in privacy preserving data mining. *ACM SIGMOD Record*, 33(1), 2004.

[27] K. Wang, B. C. M. Fung, and G. Dong. Integrating private databases for data analysis. In *IEEE ISI*, 2005.

[28] S. Waugh. In *ftp://ftp.ics.uci.edu/pub/machine-learning-databases/abalone*.

[29] L. Xiong, S. Chitti, and L. Liu. Topk queries across multiple private databases. In *25th International Conference on Distributed Computing Systems (ICDCS)*, 2005.

[30] Z. Yang, S. Zhong, and R. N. Wright. Privacy-preserving classification of customer data without loss of accuracy. In *SIAM SDM*, 2005.