ABSTRACT

PhD Dissertation

PRIVACY-PRESERVING GEO-STATISTICS

Bülent TUĞRUL

Anadolu University
Graduate School of Sciences
Computer Engineering Program

Supervisor: Assoc. Prof. Dr. Hüseyin POLAT
2014, 115 pages

Geo-statistics deals with spatial data and tries to find out relationship between locations and measured data. Methods used in geo-statistics interpolations rely on the principle that things are closer to each other more alike than the things are farther apart. *Inverse distance weighting* and *kriging* are the most well-known and applied methods in geo-statistics. It is important to perform such methods without violating data confidentiality due to privacy reasons. Also, their accuracy depends on the total number of sample points. If there are insufficient sample points due to financial or privacy reasons, accuracy of the predictions produced by these methods may become unconvincing. There are cases in which institutions obtain measurements for the same or neighbor region. To create more accurate models, they may want to collaborate. However, they do not want to share their private data.

In this thesis, privacy-preserving methods are proposed to provide inverse distance weighting- or kriging-based predictions for different data partitioning schemas including central server-based case. The proposed solutions are analyzed with respect to privacy, performance, and accuracy. Different sets of experiments are conducted using real data sets to analyze the proposed methods. Empirical outcomes show that the methods are able to provide accurate predictions while preserving privacy.

Keywords: Privacy, Geo-statistics**,** Distributed Data, Inverse Distance Weighting, Kriging, Accuracy

ÖZET

Doktora Tezi

GİZLİLİK TABANLI JEOİSTATİSTİK

Bülent TUĞRUL

Anadolu Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Hüseyin POLAT
2014, 115 sayfa

Jeoistatistik uzaysal veri ile ilgilenir ve konum ve ölçüm verileri arasındaki ilişkiyi ortaya çıkarmaya çalışır. Jeoistatistikte kullanılan enterpolasyon yöntemleri yakın nesnelerin uzak nesnelere göre daha çok birbirine benzediği prensibine dayanır. Mesafeyle ters ağırlıklandırma ve kriging jeoistatistikte en iyi bilinen ve uygulanan yöntemlerdir. Gizlilik endişelerinden dolayı bu işlemleri gizliliği ifşa etmeden gerçekleştirmek önemlidir. Ayrıca bu yöntemlerin doğruluğu ölçüm noktalarının toplam sayısına bağlıdır. Eğer ekonomik veya gizlilik nedenleriyle yetersiz ölçüm noktası var ise bu yöntemlerle üretilen tahminlerin doğruluğu inandırıcı olmayabilir. Bazı durumlarda kurumlar aynı veya komşu bölge için ölçümler elde edebilirler. Daha doğru modeller oluşturmak için işbirliği yapmak isteyebilirler. Ama gizli verilerini paylaşmak istemezler.

Bu tezde merkezi sunucu tabanlı şemayı da içeren farklı veri paylaştırma şemaları için gizliliği koruyan mesafeyle ters ağırlıklandırma veya kriging çözümleri önerilmiştir. Çözüm önerileri gizlilik, performans ve doğruluk açısından analiz edilmiştir. Bu amaçla gerçek veri setleri kullanılarak değişik deneyler yapılmıştır. Deneysel sonuçlar önerilen yöntemlerin gizliliği koruyarak doğru öneriler ürettiklerini göstermiştir.

Anahtar Kelimeler: Gizlilik, Jeoistatistik, Dağıtık Veri, Mesafeyle Ters Ağırlıklandırma, Kriging, Doğruluk

ACKNOWLEDGEMENTS

I would like to thank my advisor Assoc. Prof. Dr. Hüseyin POLAT for the support he showed me throughout my research. I am sure it would have been impossible without his motivation. His guidance helped me in all the time of research and writing of this thesis. During my research period, it has been a great pleasure to work with him.

Besides my advisor, I would like to thank my thesis committee: Prof. Dr. Yücel GÜNEY, Assoc. Prof. Dr. Cüneyt AKINLAR, Assoc. Prof. Dr. Suat ÖZDEMİR, and Assist. Prof. Dr. Cihan KALELİ for insightful comments and valuable contributions.

Last but not the least, I would like to thank my family for their endless patience during this onerous period.

Bülent TUĞRUL

January, 2014

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

ADD	:	Arbitrarily Distributed Data
G	:	Number of Neighbor Points
HE	:	Homomorphic Encryption
IDW	:	Inverse Distance Weighting
KID	:	Kriging on Integrated Data without Privacy
KPD	:	Kriging on Partitioned Data without Privacy
MAE	:	Mean Absolute Error
MS	:	Master Server
OT	:	Oblivious Transfer Protocol
P2P	:	Peer-to-Peer
PPDM	:	Privacy-Preserving Data Mining
PPDDM	:	Privacy-Preserving Distributed Data Mining
PKDD	:	Private Kriging on Distributed Data
PKPD	:	Private Kriging on Partitioned Data
RMSE	:	Root Mean Squared Error
S	:	Server

I dedicate this thesis to my wife Dilvin, and my beloved children Esma and Ömer for their support and unconditional love.

1. INTRODUCTION

1.1. Geo-statistics

Geo-statistics has been receiving increasing attention since the work conducted by Tobler (1979). It has found applications in many areas such as hydrology, meteorology, geography, forestry, agriculture, soil science, etc. Its application in those areas is found to be very useful (Krasilnikov et al., 2008). It is assumed that sample points and their corresponding values are related (Armstrong, 1998). One of the main tasks in geo-statistics is interpolation, which is a method of estimating new data points from a discrete set of known data points. *Inverse distance weighting* (IDW) interpolation is one of the eminent interpolation techniques (Ly et al., 2011, Jang, 2012). The first of the two steps in IDW interpolation is determining the neighbors of the interpolated point. The second step is taking the weighted average of the observation values within the neighborhood. In IDW interpolation, the weights are a decreasing function of distance. IDW allows users to choose a power value (*m*), which controls the significance of known points (Naoum and Tsanis, 2004). Such significance is determined on the distance between known points and the location for which prediction is sought.

Kriging is also one of the most preferred methods in geo-statistics interpolation. Kriging has two phases (Johnston et al., 2001). The first phase is to investigate the gathered data to create a semi-variogram model. The second phase is to make prediction for unobserved coordinate. The concept of kriging was first introduced by a mining engineer Krige (Krige, 1951). Kriging formulates the Tobler's first law of geography (Tobler, 1979). Tobler's law assumes that things are closer to each other more alike than the things are farther apart. A short summary of kriging method is given by Rojas-Avellaneda and Silván-Cárdenas (2006). Basic assumptions and formulas of kriging are presented by Kleijnen (2009). In a traditional kriging interpolation, there are two participating parties. One of them is referred to as *server*, which holds measurements for a specific region to make predictions for some locations in the same region. The second party is called *client*. Unlike the server, it does not hold measurements; thus, it looks for

predictions. It may need predictions to make a commercial decision in the same region for which the server has measurements.

Shahbeik et al. (2013) compare IDW and ordinary kriging methods based on error estimation in the Dardevey iron ore deposit in Iran. They show that ordinary kriging performs better than IDW with respect to accuracy. Joseph et al. (2013) test a variety of interpolation methods for 8-*h* ozone. Their results show that kriging performs better than other interpolation techniques. Triki et al. (2013) compare ordinary kriging with co-kriging with respect to estimation error; and show that ordinary kriging is superior to co-kriging. Kalivas et al. (2013) compare block kriging, block co-kriging, and IDW for the Municipal Forest of Skyros Island. Their empirical results show that block kriging gives more accurate results than the other two methods. Meng et al. (2013) investigate seven GIS interpolation methods. They conclude that regression kriging is a powerful interpolation technique.

To conduct geo-statistics interpolations, measurements for some sample points in a region are needed. Given a set of data values and the locations in which they were observed, predictions are made for selected points with unknown values. Data collected for geo-statistics are considered confidential and it is imperative to perform geo-statistics while preserving data owners' privacy. It is also vital to make enough measurements for estimating correct predictions. Without sufficient data, it becomes challenging to provide accurate and reliable predictions. Although it is possible to offer predictions with inadequate data, they might not be precise and accountable enough. Sometimes it might be costly and time consuming to collect enough measurements for sample points in a region. Or data collected for geo-statistics interpolation methods might be partitioned between two or more parties. In other words, companies, even competing ones, might collect measurements for some sample points in the same region. However, due to scarce resources and time, they might not gather enough data. Thus, data holders with inadequate data might decide to collaborate and provide predictions on their integrated data. They spend considerable effort (money, time, etc.) to gather measurements. They want to get back what they spent; and moreover, make benefit out of such effort. Hence, such measurements are considered valuable and they are often held confidential by their owners. Moreover, the locations from which the measurements are taken are

considered confidential and they are kept secret (Leitner and Curtis, 2006). Therefore, due to privacy and financial concerns, the parties might hesitate to collaborate or refuse to share data at all. On one hand, they need each other for better services. On the other hand, they do not want to disclose their private and valuable data values.

1.2. Applications

With recent developments in computer technology, geo-statistics methods have been applied in many disciplines from archeology to zoology. Geo-statistics is based on exploring the spatial correlation between measured data and location. IDW and kriging are fundamental geo-statistics methods, which are widely used in many application areas. With the invention of interpolation methods, petroleum industry became interested in geo-statistics. The industry use data from seismic surveys and wells to enhance geometry of oil reservoirs models. Other than the petroleum industry, IDW and kriging have been used in diverse fields such as health, oceanography, soil science, ecology, hydrology, environment, and so on.

1.3. Geo-privacy

Privacy is a general concept regarding the protection of confidential data. Geo-privacy refers to the protection of geo-information. The objective of geo-privacy is to protect point mapping of individual information and it is very sensitive in studies of health and crime data (Young et al., 2009). It concerns the location of sensitive data (Kwan et al., 2004). The authors study how to protect geo-privacy while making individual data available in such a way that analytical results are not significantly affected. Leitner and Curtis (2006) propose a general framework for presenting the location of confidential point data. They study how to identify geographic masking methods that protect confidentiality of individual locations. Geographic and health records of patients are used to create analytic maps to explore the relation between illness and location data. Gambs et al. (2013) propose to utilize MapReduce paradigm to efficiently perform privacy analysis on large

geo-located data sets. Exeter et al. (2014) propose a geographical privacy-access framework that guides how geographic and health data should be publicized without jeopardizing privacy of individuals. Social networks and location-based services are trendy topics of the Internet. People who use location-based services like Foursquare and Twitter are not aware of publishing private data about themselves. Li and Goodchild (2013) study how to guess the locations of home and work address of twitter users.

1.4. Privacy-Preserving Data Mining

Privacy has become a major issue for many people and companies. Some people might want to selectively reveal information if they can receive benefits in return (Cranor et al., 2000). According to the survey conducted by Cranor et al. (2000), 17% of respondents are privacy fundamentalists, 56% of respondents are concerned about data usage, and the remaining 27% are marginally concerned. Privacy-preserving data mining (PPDM) became very popular after the works by Agrawal and Srikant (2000) and Lindell and Pinkas (2000). Providing predictions to customers about some unseen or not purchased products while preserving customers' privacy has been receiving increasing attention. To provide recommendations while preserving users' privacy, various schemes have been proposed in the literature. Canny (2002a, 2002b) proposes two schemes in which users iteratively compute a public "aggregate" of their data adding vectors of user data employing cryptographic methods to privately encrypt and decrypt vectors without exposing individual data. Polat and Du (2005) utilize randomized perturbation techniques to mask customers' private data while estimating recommendations to users. Sachan et al. (2013) review the existing efficient methods in PPDM. Kaleli and Polat (2010) discuss how to provide private predictions on binary ratings in peer-to-peer (P2P) networks. Like randomized perturbation techniques, randomized response methods are also used to achieve privacy while generating recommendations on binary data (Polat and Du, 2006). Zhang et al. (2006) propose a two-way communication privacy-preserving scheme

for estimating predictions to customers, where users mask their preferences for each item according to the server's guidance.

1.5. Data Partitioning Schemes

Although it is possible to generate predictions from one of the parties' data, they might not be reliable and accurate. It is more likely to produce dependable and precise predictions from integrated data. Thus, data owners might decide to provide services on their combined data. However, data collected for interpolation purposes are considered confidential and valuable assets. The companies often do not want to reveal such data.

Data collected for various purposes might be horizontally, vertically, or hybrid (arbitrarily) partitioned. In vertical and horizontal partitioning, data are partitioned with clean-cut lines. However, in arbitrary partitioning, which is most common, there is no definite lines for data distribution. Vertical partitioning is not practical in geo-statistics methods.

1.6. Data Distribution Scenarios

Data (measurements for some sample points in a region and their location information) collected for interpolation are usually held by a single company or a server. The data owner then can provide predictions on available data to query owners. Such data distribution scenario is referred to as central-based. If each measurement and its location are held by a party or a company, this case is called P2P. Although P2P data holding is very common in some applications like recommender systems, it is not practical in geo-statistics. On one extreme, there is central-based case, while on the other extreme, there is P2P case. Between these two, data collected for interpolation purposes might be partitioned between two parties only. Such case is referred to as partitioned data-based. Likewise, data can be distributed among more than two parties or M parties. This case is called as distributed data-based. Notice that M is a constant and $2 < M << m$, where m represents the number of measurements.

1.6.1. Central-based data distribution

To estimate an unknown measurement for a location *i*, referred to as P_i, distances between the locations *i* and every nearby point *j*, referred to as D_{ij}, need to be computed. Likewise, measurements at each location *j*, referred to as P_j, are needed, where each location is defined by two coordinate values (*x*, *y*). In IDW interpolation, *m* is also needed. An example of a traditional interpolation is depicted in Figure 1.1. As seen from the figure, assume that a server *C* owns measurements (P_j values) of sample points (S_1, S_2, …, S_{25}) for region *A*. The client *Q* sends coordinates of the location *i* (x_i, y_i) for which she is seeking a prediction (P_i) to *C*. First, *C* determines *i*'s neighbours (the closest *G* points to *i*). It then estimates the prediction using IDW or kriging methods. It finally sends P_i back to *Q*. Notice that the closest five points (S_3, S_4, S_6, S_8, and S_9) are selected as neighbours, as seen from Figure 1.1. In other words, $G = 5$.

1.6.2. Two-party data distribution

It might not be an easy task to perform some measurements over a region for prediction purposes. Moreover, measurements collected by a company might not be sufficient for satisfactory predictions. Two companies may have data collected for the same region to provide predictions. Figure 1.2 shows an example of partitioned data-based interpolation between two parties. Notice that some of the measurements in Figure 1.1 are held by server *C* (measurements for sample points S_{C1}, S_{C2}, …, S_{C12}) and the remaining measurements for sample points S_{V1}, S_{V2}, …, S_{V13} are held by server *V*. When the client *Q* asks a prediction for a location *i*, she sends coordinates of the location *i* (x_i, y_i) for which she is seeking a prediction P_i to the master company *C*. *C* and *V* collaboratively estimate P_j and the prediction is sent back to *Q*.

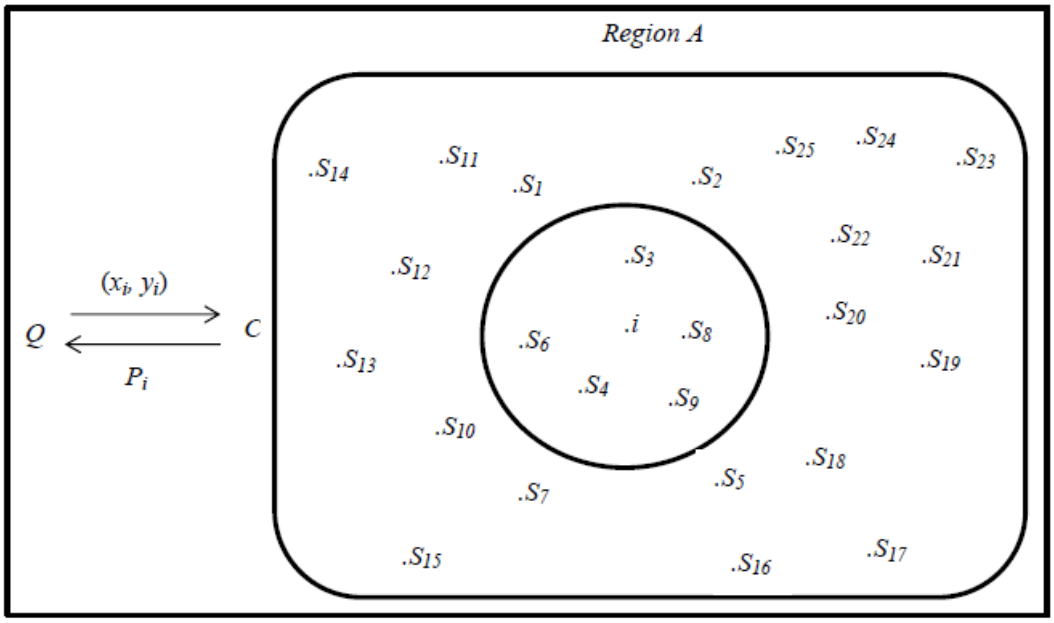

Figure 1.1. Central-based data distribution

To estimate P_j, the parties need to collaborate. Suppose that $G = 5$. When the parties estimate the prediction collaboratively, the five closest sample points (S_{V2}, S_{V6}, S_{C4}, S_{C8}, and S_{C9}), which lie inside the straight line circle in Figure 1.2, are selected as neighbours. However, if the master party C wants to estimate the prediction by itself, then the closest five sample points (S_{C1}, S_{C2}, S_{C4}, S_{C8}, and S_{C9}) whose measurements are held by C are selected as neighbours. Such points lie inside the dashed line circle in Figure 1.2. Similarly, if V wants to estimate the prediction by itself, then the closest five sample points (S_{V1}, S_{V2}, S_{V5}, S_{V6}, and S_{V7}) whose measurements are held by V are selected as neighbours. Such points lie inside the dotted line circle in Figure 1.2.

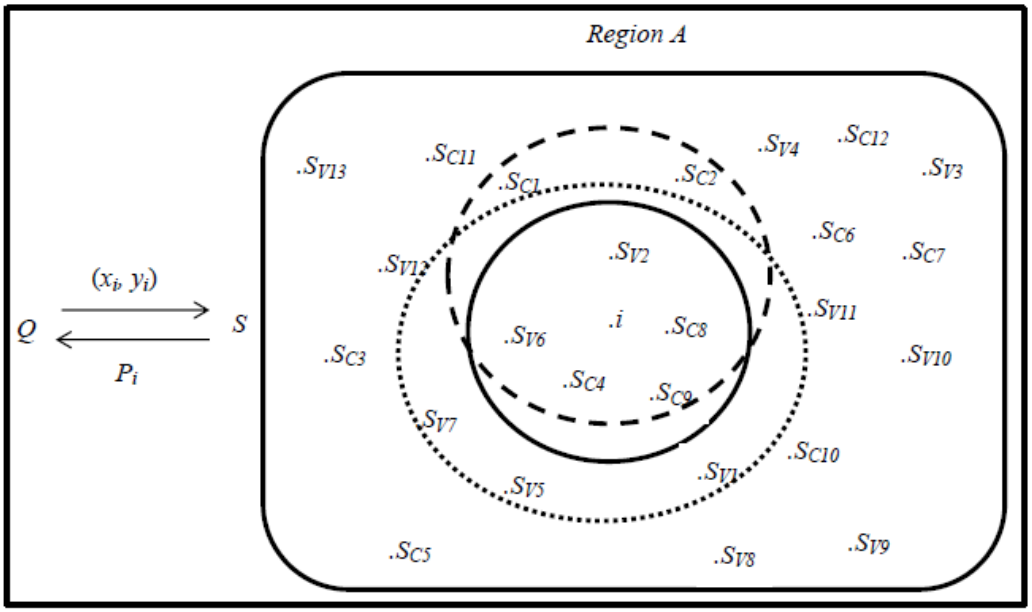

Figure 1.2. Two-party data distribution

1.6.3. Multi-party distribution

Accuracy of geo-statistics methods depends on total number of measurement points. Geo-statistics measurements require very high cost and time. Therefore, in some situations, more than two companies or institutions may join their data to provide more accurate results. As mentioned before, location and measurement data are valuable assets of such companies. In addition to this, the prediction coordinate and the estimated value should be kept secret; otherwise, the client may lose a critical economic advantage.

As seen from Figure 1.3, in a given region-region *A*, some measurements are held by one party (P_{C1j} values held by *C1*), while some are owned by another party (P_{C2j} values held by *C2*), and the remaining are held by another company (P_{C3j} values held by *C3*) for some *j*. Given a specified region, various companies can collect measurements for geo-statistical purposes, as seen from Figure 1.3. The parties then perform IDW or kriging interpolations using their data collaboratively without violating their privacy and the clients' privacy.

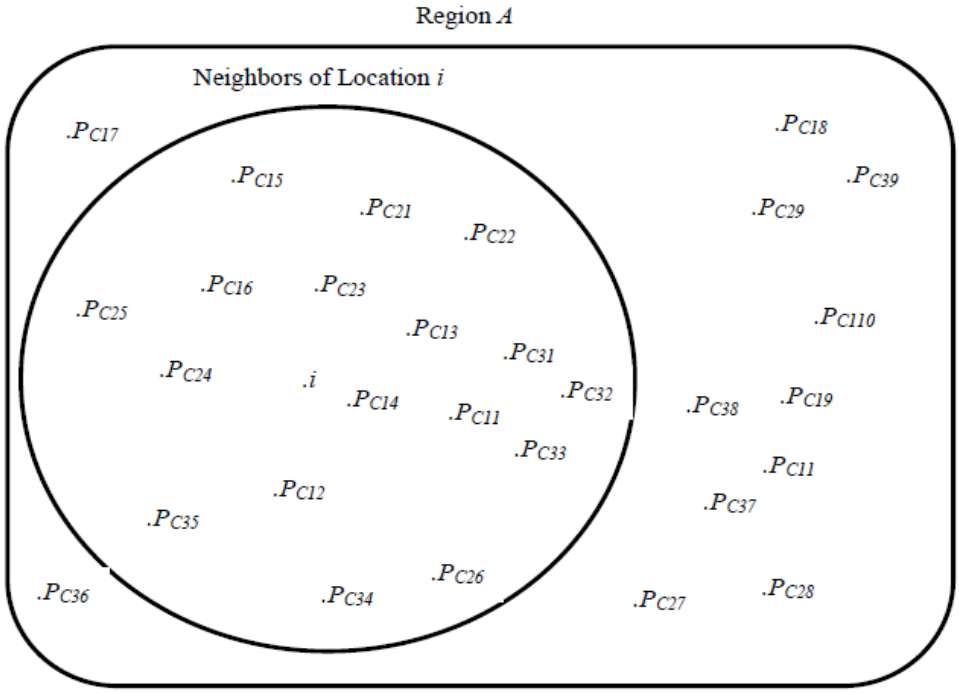

Figure 1.3. Multi-party data distribution

1.7. Privacy in Two-Party Case

There are various studies for performing different computations on partitioned data between two parties with privacy. There are horizontal, vertical, or hybrid partitioning configurations. Hybrid or arbitrary partitioning is more common over others. Nayak and Devi (2011) provide a review of the state-of-the-art methods privacy-preserving distributed data mining (PPDDM), present the related representative techniques, and point out their merits and demerits. The authors review horizontally or vertically partitioned data-based classifiers like ID3, naïve Bayesian, support vector machine, *k*-means clustering, and association rule mining with privacy. Since there are different protocols foe secure two-party computations, it is imperative to select the most appropriate ones. Thus, Kerschbaum et al. (2013) propose automatic protocol selection that chooses a protocol for each operation resulting the best performance.

Aggarwal and Yu (2008) discuss PPDDM. The authors investigate possible computational and theoretical limits that might occur in high dimensional data sets

while performing privacy-preserving functionalities. A comprehensive view on a set of metrics utilized in various PPDM algorithms including the partitioned data-based ones are provided to design more effective measurements (Bertino et al., 2008). Han and Ng (2007a) propose a protocol for secure genetic algorithms when data are arbitrarily partitioned between two parties. The parties seek to perform genetic algorithms to discover a better set of rules without jeopardizing their confidentiality. Bansal et al. (2011) present a privacy-preserving algorithm for neural network learning when data are arbitrarily partitioned between two parties. They show that the algorithm leaks no knowledge about the other's party data except the final weights. Li et al. (2011) propose a scheme for detecting outliers using distance-based approach over arbitrarily partitioned data with privacy. Bringer et al. (2013) show how to apply secure two-party computation to biometric identification. They utilize encryption to achieve privacy. Henecka and Schneider (2013) propose a secure two-party protocol, which is faster than previous implementations.

1.8. Privacy in Multi-Party Case

Performing various tasks while preserving privacy has been receiving increasing attention. Due to its popularity, distributed data-based computations are also widely accepted. Thus, how to perform distributed data-based tasks with privacy is becoming popular. Clifton et al. (2002) present some PPDDM problems and propose solutions to them as a toolkit. Duan and Canny (2008) propose an effective zero knowledge tools for PPDDM and also offer general tool for implementing many algorithms prevalent in distributed data mining. Compared to horizontal or vertical partitioning, arbitrary partitioning is more recent and the most probably encountered data partitioning case (Jagannathan and Wright, 2005).

Jagannathan and Wright (2005) introduce the concept of arbitrarily distributed data (ADD). ADD can be considered as a combination of horizontal and vertical partitioning. They propose a scheme for *k*-means clustering on ADD while preserving privacy. In order to cluster ADD using BIRCH algorithm with privacy, Prasad and Rangan (2007) propose a method. They also introduce secure protocols

for distance metrics and give a procedure for using these metrics in securely computing clusters over ADD. Han and Ng (2007b) present an efficient method to perform the secure scalar product operation based on ADD among multiple parties and show how to perform decision tree induction algorithm on ADD while preserving data holders' confidentiality.

Majority of the multi-party computation methods require that all parties should be connected during computation phase. This situation may raise different problems. Gordon et al. (2013) foresee the problems brought by being online all time during computation. They propose more efficient solutions. In multi-party computation methods, it is assumed that all parties are honest, which means that they follow the protocol but they try to learn as much information from the others. This is not always true. A party may mislead other parities. Zhang et al. (2013) propose a series of protocols, which are secure and verifiable in terms of computation results. They also analyze the existing protocols against possible attacks. Prabhakaran and Sahai (2013) provide a comprehensive body of basic and advanced material on secure multi-party computation including classical and recent protocols. Some real world applications require that a query should be performed on different databases owned by multi parties. Sepehri et al. (2013) propose a secure solution, which allows performing equality test among multi databases without revealing private data of each database owner.

1.9. Related Work

Geo-statistical interpolation has been receiving increasing attention since the study conducted by Krige (1951), which forms the basis of geo-statistical works in the literature. Similarly, the study proposed by Shepard (1968) forms the basis of IDW. The author proposes a two-dimensional interpolation function for irregularly-spaced data. After such studies, Tobler (1979) has prompted geo-statistical studies. Along with kriging, IDW is one of the most widely used deterministic models in interpolation (Lu and Wong, 2008). It is relatively fast and straightforward to interpret. IDW interpolation method has been expanded by Bartier and Keller (1996) to allow users to define the expected degree of surface abruptness along

thematic boundaries using a transition matrix. Li et al. (2010) first analyzed the traditional IDW interpolation technique and then improved it into the grade estimation model. Their proposed model promotes the application of IDW method by combining ore body occurrence elements with the interpolation method.

Geo-statistics interpolation methods can be grouped as deterministic and geo-statistical methods (Rivoirard and Romary, 2011). IDW is a widely used deterministic method while kriging is the main tool used as a geo-statistical method (Fritz et al., 2009). Hence, in order to estimate predictions for unmeasured locations, IDW and kriging interpolations are widely used. Krige (1951) who developed kriging proposes to use kriging in order to predict the ore reserves. Armstrong (1998) explains the application of geo-statistics in mine reservoirs to calculate capacity of mine reservoir and errors. Shad et al. (2009) utilize kriging for air pollution prediction, where the authors employ a genetic algorithm to optimize membership functions to improve accuracy. Kriging-based techniques are used to predict and analyze soil properties (Sun et al., 2012). The authors perform some experiments and demonstrate that their approach provides highly accurate outcomes for some specific cases. They also develop a software program to perform local regression kriging automatically. In addition to analyzing soil properties and air pollution prediction, kriging is also utilized to estimate soil contamination (Largueche, 2006). Largueche (2006) investigates whether kriging is a useful tool to estimate the spatial distribution of ground pollutants in contaminated land. The author also discusses the identification of areas that should be subjected to remedial actions. Kaymaz (2005) proposes to apply kriging to structural reliability problems. The author investigates the use of kriging for such problems and compares it with response surface method. Ali et al. (2006) apply kriging to the spatial interpolation of local disease rates. Their approach helps researchers incorporate the pattern of spatial dependence into the mapping of risk values.

Privacy-preserving and secure multi-party computation methods give us opportunities to conduct data mining methods without revealing information to other parties. Agrawal and Srikant (2000) propose randomized methods to hide sensitive information. The authors show that accurate predictive models can be created from a large number of perturbed data items. Evfimievski (2002) discusses

perturbation levels against privacy levels and presents some methods to measure privacy. Li and Sarkar (2006) propose a perturbation method for categorical data to prevent disclosure of private data. Their scheme is based on two steps consisting of linear programming and swapping. In (Chen and Liu, 2011), the authors propose geometric data perturbation for preserving confidential data and discuss different aspects of such method. Taur et al. (2012) propose an enhanced method based on table look-up in order to improve the performance of substitution data hiding method. They present a general form of the method and show that their scheme significantly improves the amount of hidden data. Likewise, Guo et al. (2012) propose a new data hiding scheme establishing an injection mapping. Their empirical outcomes show that their method has a stable and efficient embedding capacity. In (Choi et al., 2012), the authors propose a solution to the problem of preserving mining accuracy and privacy in publishing sensitive time-series data. The authors propose both naïve solutions and advanced one, where they discuss randomization-based solutions. Li and Wang (2012) propose a classification method based on singular value decomposition with privacy. Meskine and Bahloul (2012) study and analyze privacy-preserving *k*-means algorithms and classify them based on data distribution, where they discuss advantages and disadvantages of each proposed protocol. Baboulin et al. (2013) propose a random transformation, which can be performed efficiently while providing sufficient accuracy.

Performing interpolations with privacy on central data, partitioned, or distributed data has not been studied in the literature before. On the other hand, Tugrul and Polat (2013a) show how to perform kriging-based predictions while preserving the client's and the server's privacy. The authors consider a central server-based scenario in which the data are held by a single party, referred to as the server. In another study, Tugrul and Polat (2013b) study how to provide IDW-based predictions from centralized data without violating the client's and the server's confidentiality. In both studies, the authors focus on central server-based schemes rather than partitioned data-based methods.

1.10. Contributions

Geo-statistics and privacy-preserving schemes have been used in many applications. As mentioned before, collecting data for geo-statistics analysis requires so much time and money. Moreover, amount of data used for prediction effects the accuracy of geo-statistics methods. In general, there are more than 20 different geo-statistics methods in the literature. On the other hand, privacy-preserving schemes enable to apply various methods without publishing sensitive data to competitive parties. To the best of our knowledge, our study is the pioneer of using privacy-preserving schemes in two of the geo-statistics methods. IDW and kriging, which are two common methods used as geo-statistics interpolation methods, are chosen. Solutions for central data distribution scheme are first proposed. In these schemes, data are held by one company only and clients request prediction for a specific location, where they are interested in. The location in which the client asks prediction and measurement values are accepted as sensitive data of client and server. The proposed IDW and kriging-based solutions protect sensitive data of both parties.

In some situations, companies may collect data for the same region or neighbor regions in respect to economic or legislation reasons. It is assumed that there may be two or more competitive companies. This first scenario is called as two-party. The second scenario is depicted as multi-party. IDW and kriging-based methods are proposed for both scenarios, as well. In these scenarios, coordinates and measurement values of each company and prediction location for which client is interested in are assumed as sensitive data. It is also assumed that the companies are semi-honest. In other words, they follow the protocol as required; however, they try to acquire as much data as possible about each other's private data.

For the clients, the location and the estimated prediction are considered confidential and valuable asset because the clients plan their investments according to predicted measurements for specific locations. Similarly, locations of surrounding points and their measurements are confidential data for the servers. They do not want to disclose them. Moreover, they utilize such data to provide predictions to their clients in return of some benefits. Thus, they are also considered

valuable assets and the servers do not want to share them with the third parties. IDW- or kriging-based predictions should be estimated without revealing the confidential data. However, the area in which the servers have measurements is considered public.

Individual users or companies might not want to spend money and/or time to estimate unknown measurements of some locations due to scarce resources. Instead of using their own effort, they prefer obtaining estimations from those who have enough measurements and provide prediction services to others. Suppose that the client wants to get a prediction from the server. The client must send the location information to the server, which then must estimate the prediction based on the location information it receives and the measurements for surrounding points it has. However, the server does not want to reveal information about the measurements and their locations due to privacy and financial reasons. Similarly, the client does not want to reveal the location information for which it is looking for prediction and the estimated prediction to the server. Hence, neither the server nor the client wants to disclose their confidential data (locations, measurements, and predictions) to each other. The problem is *how to estimate predictions using IDW or kriging for the clients without revealing the server's and the clients' private data*? Thus, privacy-preserving methods are proposed in order to provide predictions using IDW or kriging interpolations. The proposed methods protect both the server's and the client's confidentiality against each other. They are able to provide accurate predictions without greatly jeopardizing privacy.

Performance and accuracy of interpolations success are mainly dependent on number of sample points to be used and quality of the collected measurements for prediction purposes. It is unlikely to provide accurate and dependable predictions from insufficient and/or false data. Measurements collected for interpolation purposes might be partitioned between two parties, even competing companies. Instead of collecting all measurements due to limited time and budget, such parties might decide to provide interpolations based on their integrated data. To offer correct and reliable predictions, they might decide to collaborate. However, due to privacy and financial concerns, they do not want to disclose their confidential data. Without privacy protection, they hesitate to collaborate for better interpolation

services. Thus, the problem is *how such parties perform IDW- or kriging-based interpolations while preserving their privacy and the clients' confidentiality*? In order to provide partitioned data-based predictions using IDW or kriging interpolations while preserving privacy, different methods are proposed. Due to privacy measures, the proposed solutions might bring extra costs like storage, communication, and computation. However, such costs should not prevent the servers from providing predictions efficiently. Moreover, accuracy losses are inevitable due to the utilized privacy measures. On one hand, it is hypothesized that accuracy improves if the servers decide to collaborate. On the other hand, privacy measures cause accuracy losses. However, overall gains due to collaboration should compensate the losses due to privacy concerns. Empirical outcomes show that the proposed methods provide accurate predictions efficiently with privacy.

As in data mining and statistics applications, the main requirement for performing dependable and truthful interpolations is to gather enough amounts of data. For interpolations, data values are measurements representing some quantities for some pre-determined locations in a geographical region. Various companies, firms, institutes, organizations, and so on measure some quantities for interpolation purposes. Even if a few numbers of organizations do not expect any benefit from such measurements, most of them plan to make money out of such collected data. There are mutual benefits between such data owners and those seeking predictions. On one hand, data collectors spend noteworthy efforts including but not limited to budget, time, labor, and so on in order to gather measurements for interpolation. It is reasonable for them to try to compensate what they spent out of such values. On the other hand, some companies, even rival ones, might not have enough resources to collect data. They prefer to buy services from those who willing to offer predictions in return of some benefits.

When a company owns adequate data for providing reliable and correct predictions, it is a straightforward task for it to offer such services. However, due to scarce resources, it might not be possible to assemble enough measurements. Moreover, different companies gathering measurements from the same region might decide to provide predictions on their integrated data collaboratively. It is more likely to sustain more accurate predictions from joint data than the ones on

split data only. Predictions with decent accuracy help service providers keep existing customers (query owners) and recruit new ones. That results more benefits for them. On the other hand, query owners obtain more truthful predictions. Due to such mutual advantages (between collaborating parties or data and query owners), it becomes interesting and inevitable to perform interpolations on distributed data for both service providers and clients. However, privacy and financial concerns might prevent them from conducting such services without any protection. Therefore, the problem is *how to offer predictions using IDW- or kriging-based on distributed data while preserving privacy*? In order to address the problem, various privacy-preserving methods are proposed. The schemes make it possible for the servers and the clients to perform interpolations on distributed data without violating confidentiality. The solutions provide correct and dependable predictions efficiently while preserving privacy.

1.11. Organization of the Dissertation

The remainder of the thesis is organized as follows. In the following chapter, IDW and kriging methods are explained briefly. In addition, auxiliary methods that are used in our solutions are described. Privacy-preserving IDW and kriging-based schemes on central data are presented in Chapter 3 and Chapter 4, respectively. The proposed solutions for estimating predictions using IDW on partitioned or distributed data are presented in Chapter 5. The next chapter explores solutions of kriging on partitioned or distributed data. Finally, in the last chapter, conclusions are presented and future research directions are pointed out.

2. PRELIMINARIES

2.1. Geo-statistics in General

Interpolation methods used in geo-statistics have been played a significant role in planning, resource management, decision making, and risk assessment. With the help of powerful geographic information systems and modeling tools, geo-statistics analyses become popular between scientists to develop accurate predictions. There are several methods, which can be categorized into three sections like non-geo-statistical methods, geo-statistical methods, and combined methods (Li, 2008). IDW is an example of non-geo-statistical methods. Kriging, on the other hand, is a geo-statistical method. Although there are different kriging methods, ordinary kriging is studied because it is the basis of other types of kriging schemes.

2.2. Inverse Distance Weighting-based Interpolation

To predict a value for an unknown point, IDW assigns a weight to each of the surrounding points. Weights get smaller as a function of distance. In other words, weights are proportional to the inverse of the distance raised to power m. If $m = 0$, then the weights do not depend on distance. If $m > 1$, the effects of distant points get smaller. The measure in location i (referred to as P_i) using IDW approach can be estimated as follows (Armstrong, 1998):

$$P_i = \frac{\sum_{j=1}^{G}\left[P_j/(D_{ij})^m\right]}{\sum_{j=1}^{G}\frac{1}{(D_{ij})^m}} \tag{2.1}$$

in which G represents the number of neighbor points, P_j shows the measurement in point j, and D_{ij} represents the distance between points i and j. In a relatively small area with appropriate coordinate system, D_{ij} can be calculated using Euclidean distance metric, which measures the straight-line distance between any two points. For larger areas in which there are no straight lines, geodesic distance is utilized to compute D_{ij}, where geodesic distance is the distance measured along the shortest route between two points on the Earth's surface (Johnston et al., 2001).

2.3. Kriging-based Interpolation

Kriging is a widely used technique for interpolating some unknown measurements. It has been applied in many engineering fields such as petroleum, mining engineering, forestry, remote sensing, meteorology, and so on. Kriging interpolation is divided into two tasks: finding a variogram model using all measured points and making predictions.

As explained previously, there are two involving parties in a traditional kriging interpolation. The server *S* owns measurements (*P* values) for some *G* sample locations with their related coordinates (*x*, *y*) in a given region *A*. The client *C* asks a prediction for some location, referred to as unmeasured location (*q*) from *S*. The steps of such interpolation are given as follows (Johnston et al., 2001):

1. *S* computes distances between any two measured locations *i* and *j* in *A* using Euclidean distance measure. The distance between *i* and *j* (d_{ij}) can be calculated as follows:

$$d_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}. \tag{2.2}$$

2. Then, *S* calculates semi-variances (*s* values) between any two measured locations, *i* and *j* as follows:

$$s_{ij} = 0.5 \times \left[P_i - P_j\right]^2. \tag{2.3}$$

3. *S* then groups sample points using binning and finds average semi-variances and distances for each bin.
4. *S* plots average semi-variances versus average distances and finds the formula to estimate semi-variance at any given distance. Semi-variances can be denoted as follows:

$$Semi\ variance = f(distance), \tag{2.4}$$

where *f* is a function representing the relationship between semi-variances and distances.

5. *S* then creates **Γ** matrix, which is a $(G + 1) \times (G + 1)$ symmetric matrix including the estimated semi-variances between any two locations using Eq. (2.3). Note that the last row and column are filled with 1s, except the diagonal entry, which is set to 0.

6. Next, *S* finds $\mathbf{\Gamma^{-1}}$ matrix, which is again a $(G + 1) \times (G + 1)$ symmetric matrix including γ values.
7. *C* sends the coordinates of q (x_q, y_q) in *A*, for which she is looking for prediction, to *S*.
8. *S* computes distances between q and each measured location in *A* using Eq. (2.2). It creates the matrix **g**, which is a $(G + 1) \times 1$ matrix including the semi-variances estimated between q and each measured location using Eq. (2.3).
9. *S* then solves the kriging weights ($\boldsymbol{\lambda}$ matrix) as follows:

$$\boldsymbol{\lambda} = \mathbf{\Gamma^{-1}} * \mathbf{g} \tag{2.5}$$

in which $\boldsymbol{\lambda}$ is a $(G + 1) \times 1$ matrix.

10. Finally, *S* estimates the final prediction for unmeasured location (referred to as P_q) by multiplying the weight for each measured location and the related measure or value; and adds them together. If $\boldsymbol{\lambda}$ and **P** are considered as vectors of length *G*, then P_q can be estimated by finding the scalar product of $\boldsymbol{\lambda}$ and **P** as follows:

$$P_q = \boldsymbol{\lambda} \cdot \mathbf{P} = \sum_{i=1}^{G} \lambda_i * P_i. \tag{2.6}$$

2.4. Homomorphic Encryption

Encryption methods are widely used to provide privacy. There are symmetric and asymmetric encryption algorithms. Although symmetric encryption is based on one common secret key, asymmetric algorithms utilize one private key and one public key. Homomorphic encryption (HE) methods are based on asymmetric encryption. HE allows an addition or a multiplication operation to be conducted on encrypted data without decrypting them. Untrusted parties can perform operations on encrypted data without knowing real value of the other party. Several HE systems are available and examples include the systems proposed by Benaloh (1994), Paillier (1999), and Cheon et al. (2013). Aguilar-Melchor et al. (2013) survey about the recent advances in HE with respect to both cryptography and software engineering. Huang et al. (2013) investigate HE with respect to ad hoc networks. They show that their proposed scheme is secure and practical in ad hoc networks and cloud computing. Niu et al. (2013) utilize HE in order to detect

whether smart meter data is correct or not. In other words, energy suppliers can understand whether the data packet is tampered or not.

The scheme is basically described by Senyurek and Yakut (2013) as follows: HE scheme consists of key generation, encryption, and decryption. In order to generate keys, two large prime numbers (p and q) are uniformly randomly selected. Then, $n = pq$ and $\lambda = lcm\ (p - 1, q - 1)$ are calculated, where λ and lcm are Carmichel's function and least common multiplier, respectively. After that integer numbers γ and δ are selected from Z_n (set of integers n) to determine the generator $f = (\gamma n + 1)\ \delta^n$ **mod** n^2. Then, the public key for encryption generated as (n, f) and the private key for decryption generated as (λ, μ), where $\mu = (L(f^{\lambda}$ **mod** $n^2))^{-1}$ **mod** n and $L\ (u) = (u - 1)/n$. The message M in Z_n can be encrypted as follows: $M' = f^M r^n$ **mod** n^2. Note that M' represents the related cipher text, r is a random number selected from Z_n* (set of integers co-prime to n). The cipher text M' can be decrypted as follows: $M = L\ ((M')^{\lambda}$ **mod** $n^2)\mu$ **mod** n.

Assume that the task is to compute $\xi_e(M_1 + M_2)$ from $\xi_e(M_1)$ and $\xi_e(M_2)$. The cipher texts can be found as $M_1' = f^{M_1} r_1^n\ mod\ n^2$ and $M_2' = f^{M_2} r_2^n\ mod\ n^2$, where r_1 and r_2 are random numbers. If the cipher texts are multiplied, $M_1' M_2'\ mod\ n^2 = f^{M_1} r_1^n \times f^{M_2} r_2^n\ mod\ n^2 = f^{M_1 + M_2} (r_1 \times r_2)^n\ mod\ n^2$ are obtained. If the encrypted result is decrypted, the outcome will be $M_1 + M_2$.

2.5. Oblivious Transfer Protocol

In an untrusted environment, two parties might want to collaborate to achieve a common goal. To allow one of the parties to get its choices only while preventing the other party from learning such choices, 1-out-of-n oblivious transfer protocol (OT) proposed by Even et al. (1985) and Brassard et al. (1987) can be used. Naor and Pinkas (1999) propose an efficient protocol. By combining it with the one by Cachin et al. (1999), OT could be achieved with poly-logarithmic (in n) communication complexity. Tzeng (2002) presents efficient OT schemes. Asharov et al. (2013) present optimizations and efficient implementations of OT. They also offer a novel OT. The author builds OT from fundamental cryptographic techniques. Kolesnikov and Kumaresan (2013) propose an improved OT for

transferring short secrets. OT is beneficial in the proposed schemes because it allows one party to select required values only while preventing the other party from deriving which values are chosen.

At the beginning of this protocol, one party, Bob has n messages $M_1, M_2, \ldots, M_n$ and at the end of the protocol the other party, Alice, learns one of the inputs M_α for some $1 \leq \alpha \leq n$ of her choice, without learning anything about the other inputs and without allowing Bob to learn anything about α. Tzeng (2002) basically explains the protocol as follows: Let the sender's (Bob) input is $M_1, M_2, \ldots, M_n$ in H_q and the receiver's (Alice) choice is α. Remember that $M_1, M_2, \ldots, M_n$ represent n messages. H_q is an order-q group, where q is a prime. Let a and b be two generators in H_q and r is a random number, which is uniformly randomly selected from a set R. The receiver computes $y = a^r b^\alpha$ and sends y to the sender. After receiving y, the sender first computes $d_i = \left(a^{r_i}, M_i \left({}^{y}/_{b^i}\right)^{r_i}\right)$, where r_i is a random number selected uniformly randomly from a set R by the sender and $1 \leq i \leq n$. He then sends d_i values for all i to the receiver. Let $d_\alpha = (X, Y)$, then the receiver computes $M_\alpha = Y/X^r$. Notice that M_α is the choice that Alice is looking for.

2.6. Data Sets

To analyze the proposed methods, different sets of experiments are conducted using real data sets. Two real data sets obtained from the U. S. National Geochemical Survey Database are utilized in the experiments. The first data set contains Sodium (Na) content of soil in the Illinois State. There are 1,331 measurements. Minimum and maximum Na contents measured 0.0180 and 1.2730, respectively. Mean and median of the data set are 0.6023 and 0.5970, respectively. Figure 2.1 shows the distribution of the Illinois data set.

The second data set contains Sodium (Na) content of soil in the Colorado State. As seen from Figure 2.2, there are 1,150 measurements in total along the state. Minimum and maximum Na contents measured 0.0630 and 3.2300, respectively. Mean and median of the data set are 0.9986 and 0.9115, respectively.

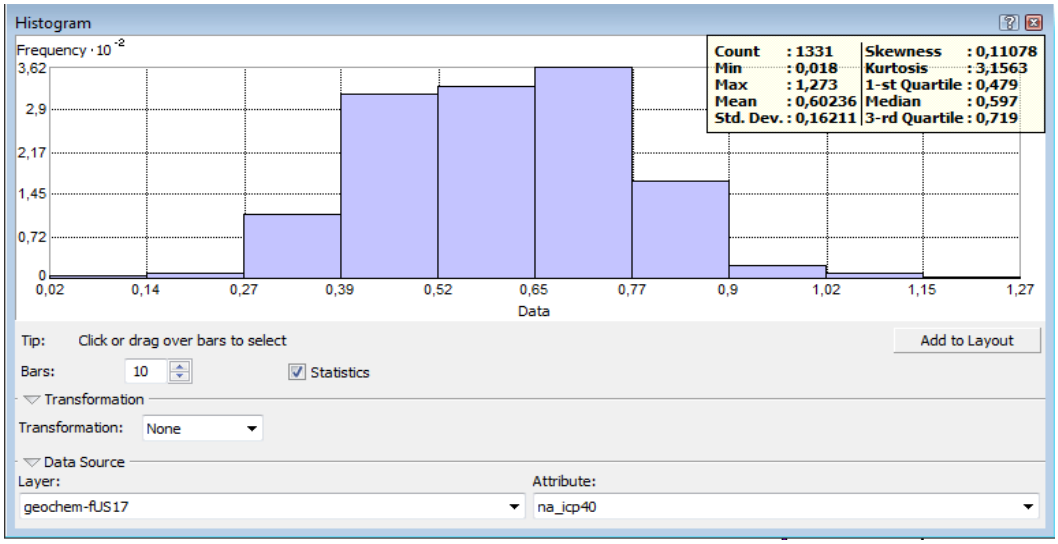

Figure 2.1. Histogram of the Illinois data set

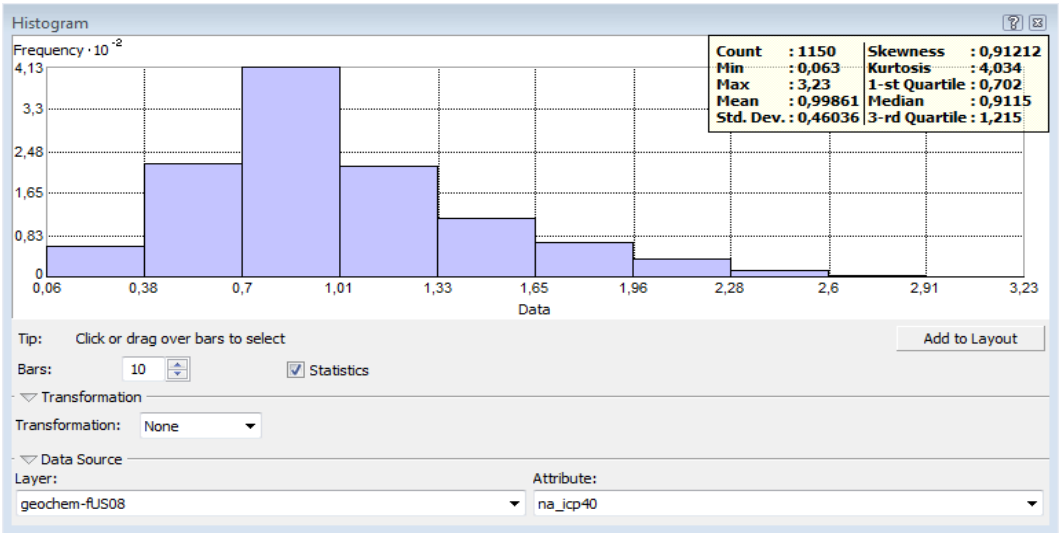

Figure 2.2. Histogram of the Colorado data set

2.7. Evaluation Metrics

There are various evaluation metrics that can be used to assess IDW and kriging methods (Li, 2008). Mean absolute error (MAE) and root mean squared error (RMSE) measures are utilized in the experiments. The smaller they are, the more accurate a scheme is. Thus, smaller MAE and RMSE values mean that

predictions of a scheme are very close to the observed measurements. MAE can be formulized as follows:

$$MAE = \frac{1}{M}\sum_{i=1}^{M}\left|o_i - p_i\right|. \tag{2.7}$$

and similarly, RMSE can be formulized as follows:

$$RMSE = \left[\frac{1}{M}\sum_{i=1}^{M}\left(o_i - p_i\right)^2\right]^{1/2} \tag{2.8}$$

in which M is the number of observed measurements in the test set, o_i and p_i are original measurement and predicted output of our proposal, respectively.

Leave-one-out methodology is utilized in the experiments. It means that one location is left out and the others are used to estimate a prediction for that one. The same thing is done for all locations. Observed measurements are withheld and estimated predictions are found for each one. Then they are compared and the error for each point j is calculated as follows: $Error_j = Observed_j - Prediction_j$. After error calculation, RMSE and MAE values are computed. The experiments are conducted several times and overall averages are displayed.

3. PRIVACY-PRESERVING IDW INTERPOLATION

Without privacy concerns, it is an easy task to predict unknown measures for specific locations using IDW interpolation technique. Basically, the client sends location information for which it does not know the measurement to the server. After estimating the prediction based on the measurements it has using IDW interpolation, the server then sends it back to the client. However, if they want to achieve such task with privacy, then it becomes a challenging task.

3.1. Method

As seen from Eq. (2.1), to estimate a prediction for a location i, referred to as P_i, distances between the location i and every one of G nearby point j (D_{ij}) need to be computed. Moreover, measures at each location j and m are needed. It is assumed that each location i is defined by two coordinate values (x_i, y_i). The goal is to provide schemes that allow the client (Alice) and the server (Bob) estimate predictions without violating their confidentiality.

The proposed solutions are described in the following starting from the naïve one. Naïve solution assumes that coordinate values and measurements held by the server are private. However, prediction coordinate for which Alice needs a prediction and prediction value itself are public. Second solution assumes that coordinate values held by Bob are public. On the other hand, prediction coordinate and estimated value are considered confidential data. Finally, the complete solution protects coordinates and measurements held by Bob. It also considers prediction coordinate and estimated prediction for Alice as private data.

3.1.1. Naïve solution

The naïve scheme is based on OT. The server and the client aim to hide their confidential data from each other. The steps of the method are as follows:

1. Alice first uniformly randomly generates n-1 bogus locations.

2. She then hides the location *i* for which she is looking for a prediction among such *n*-1 spurious locations. She simply permutes such *n* locations.
3. Next, she sends $G_j = (x_j, y_j)$ to Bob, where $j = 1, 2, \ldots, n$ and the location for $i = j$ is the real one.
4. For each point *j*, Bob adds random number to optimum power value m_o values by uniformly randomly selecting m_{zj} values over the range $[m_o-\theta, m_o+\theta]$.
5. Bob then estimates predictions for each point *j* using the Eq. (2.1) and corresponding m_{zj} values.
6. Alice finally uses OT to obtain the prediction P_i for her true location *i*.

In this scheme, the server can guess the prediction for location *i* with probability of 1/*n*. Notice that with increasing *n* values, this probability becomes smaller.

3.1.2. Second scheme: Relaxed privacy constraints

In this method, it is assumed that the coordinate values held by Bob are public. On the other hand, other data values are considered confidential data. The steps of the proposed approach are as follows:

1. Bob first sends *G* sample points' coordinate values to Alice.
2. Alice computes distances between *i* and each location *j* received from Bob (finds D_{ij} values).
3. She adds some randomness to optimum power values m_o by uniformly randomly selecting m_{zj} values over the range $[m_o-\theta, m_o+\theta]$.
4. Next, she estimates $\sum_{j=1}^{G} \frac{1}{(D_{ij})^{m_j}}$ and $1/(D_{ij})^{m_j}$ values for all $j = 1, 2, \ldots, G$.
5. After finding $\xi_{KA}(1/(D_{ij})^{m_j})$ values for all $j = 1, 2, \ldots, G$ using an HE scheme, she sends them to Bob, where *KA* represents Alice's public key.
6. Since Bob knows the measures for each location *j* (P_j values), he first finds $[\xi_{KA}(1/(D_{ij})^{m_j})]^{P_j} = \xi_{KA}(P_j/(D_{ij})^{m_j})$ values for all $j = 1, 2, \ldots, G$ using an HE scheme.
7. After computing $\prod_{j=1}^{G} \xi_{KA}\left(P_j/\left(D_{ij}\right)^{m_j}\right) = \xi_{KA}\left(\sum_{j=1}^{G}\left(P_j/\left(D_{ij}\right)^{m_j}\right)\right)$ using the HE property, he sends the encrypted sum to Alice.

8. Since Alice knows the decryption key, which is her corresponding private key, she decrypts the received encrypted sum and obtains $\sum_{j=1}^{G}\left(P_j/\left(D_{ij}\right)^{m_j}\right)$.
9. She finally estimates P_i as follows:

$$P_i = \frac{\sum_{j=1}^{G}\left(P_j/\left(D_{ij}\right)^{m_j}\right)}{\sum_{j=1}^{G}\frac{1}{(D_{ij})^{m_j}}}$$

3.1.3. The complete solution

In this final scheme, sample points coordinates values and their corresponding measurements held by Bob are confidential. Alice wants to hide the location for which she is looking for a prediction. She also wants to prevent Bob from learning the prediction value. In other words, all confidential data described previously are wanted to be protected. The steps of this approach are as follows:

1. Bob first uniformly randomly generates *N* bogus locations. Now, he has $g = G + N$ sample points.
2. For every sample point *j*, the parties perform the followings, where $j = 1, 2, \ldots, g$.
   a. Bob uniformly randomly creates *n*-1 bogus points and hides *j* among them.
   b. He then sends such *n* points coordinates to Alice.
   c. Alice adds some randomness to optimum power value m_o by uniformly randomly selecting m_{zj} values over the range $[m_o\text{-}\theta, m_o\text{+}\theta]$.
   d. After finding the distances between *i* and each received point *z*, she finds $1/(D_{iz})^{m_z}$ values for $z = 1, 2, \ldots, n$, where $m_z = m_j$.
   e. She encrypts $1/(D_{iz})^{m_z}$ values for $z = 1, 2, \ldots, n$ using an HE scheme and her public key *KA*; and obtains $\xi_{KA}\ (1/(D_{iz})^{m_z})$ values.
   f. Bob uses OT and requires $\xi_{KA}\ (1/(D_{iz})^{m_z})$ for $z = j$. In other words, Bob gets the distance between points *i* and the real location *j* in encrypted form.
3. After performing the abovementioned steps for all *g* points, Bob can choose those encrypted values for *G* real locations by removing the results for bogus points.

4. Bob then estimates $\xi_{KA}\left(\sum_{j=1}^{G}\left(1/\left(D_{ij}\right)^{m_j}\right)\right) = \prod_{j=1}^{G}\xi_{KA}\left(1/(D_{ij})^{m_j}\right)$ using the HE scheme.
5. He also computes $\xi_{KA}\left(\sum_{j=1}^{G}\left(P_j/\left(D_{ij}\right)^{m_j}\right)\right) = \prod_{j=1}^{G}\xi_{KA}\left(1/(D_{ij})^{m_j}\right)^{P_j}$ using the HE property.
6. Finally, Bob sends such encrypted aggregate sums to Alice.
7. Since Alice knows the corresponding decryption key, which is her private key, she decrypts them and obtains $\sum_{j=1}^{G}\left(P_j/\left(D_{ij}\right)^{m_j}\right)$ and $\sum_{j=1}^{G}\left(1/\left(D_{ij}\right)^{m_j}\right)$ values.
8. She finally estimates P_i using the Eq. (2.1).

3.2. Analysis

The last scheme is analyzed because it achieves all privacy requirements. The proposed scheme is scrutinized in terms of privacy, accuracy, and performance because like in many other applications, they are three major goals that various approaches should accomplish. However, they are conflicting goals; and improving one or two makes the other(s) worse. Thus, the proposed scheme should find equilibrium among them.

Privacy means that the client should not be able to learn the locations of surrounding points and their measurements; and the server should not be able to know the estimated predictions and the locations for which the client is looking for predictions. Accuracy can be defined as follows: Predictions estimated with privacy concerns should be as close as possible to the ones estimated without privacy concerns. And finally, efficiency can be described as follows: Additional costs due to privacy concerns should be negligible. The proposed scheme should not introduce too much supplementary communication, computation, and storage costs. Although extra costs are inevitable and they cause some overheads, online efficiency is not critical in IDW-based predictions. In some real-time applications like obtaining recommendations from e-commerce sites, it is imperative to provide recommendations to many customers in a limited time. However, in geo-statistics, online time restrictions are tolerable.

3.2.1. Supplementary costs analysis

Additional costs like storage, communication (number of communications and amount of transferred data), and computation costs are first analyzed. Extra storage costs are needed for the server to save the randomly chosen N locations and randomly generated n-1 locations for each location $j = 1, 2, \ldots, g$. Thus, supplementary storage costs for the server are in the order of $O(ng)$. For the client, she has to save additional n encrypted values for each $j = 1, 2, \ldots, g$. Hence, extra storage costs for the client are in the order of $O(ng)$. On average, the scheme causes additional storage costs in total, which are in the order of $O(ng)$, where $g = G + N$.

In a traditional IDW interpolation system, the client sends the location information to the server. The server returns an estimated prediction. Thus, without privacy concerns, number of communications is *two* only. In the proposed scheme, the server communicates with the client to send location information for all g points. It also performs OT protocol g times to receive an encrypted value in each time, where remember that each protocol could be achieved with poly-logarithmic (in n) communication complexity. Finally, the server sends two encrypted partial sums to the client. Thus, number of communications are $(g + ng + 2)$, which are in the order of $O(ng)$; and it increases from two to $(g \times (n + 1) + 2)$.

Besides number of communications, amount of transferred data is also important. Without privacy concerns, amount of data sent from the client to the server are about eight bytes, assuming that four bytes are needed for each coordinate value. Similarly, since the server returns a prediction to the server, amount of transferred data are about four bytes. In the proposed scheme, since the server sends n pairs of coordinate values rather than one pair, amount of transferred data are about $8n$ bytes. Thus, amount of exchanged data to send location information increases n times. During data transfer via OT, encrypted values are transferred. The size of the encrypted value produced by the block cipher encryption can be computed as follows: *size of plain text* + *block size* - (*size of plain text* **mod** *block size*) (Obviex, 2012). Assuming that 256-bit key or 32 bytes blocks are utilized, 8 + 32 – (8 **mod** 32) = 32 bytes are needed. Thus, amount of transferred data in each ng communications are about 32 bytes. Finally, the server sends two encrypted

values in which amount of data sent are about 64 bytes. Although number of communications and amount of transferred data increases due to the proposed scheme; however, as mentioned previously, online interaction is not critical for IDW interpolation methods. Such communications can be considered performed off-line.

The proposed method also causes additional computation costs due to privacy concerns. As seen from Eq. (2.1), without privacy concerns, the server conducts G number of multiplications, distance calculations, exponentiations, and $(G + 1)$ divisions, where addition is omitted. Thus, the computation complexity is in the order of $O(G)$. In the scheme, the client performs ng encryptions, divisions, and exponentiations due to bogus locations. She also conducts two decryptions and one final division. Similarly, the server performs G exponentiations and multiplications. Hence, the computation complexity of the method is in the order of $O(ng)$. Since the scheme includes encryptions and decryptions, their running times can be determined using the benchmarks for the CRYPTO++ toolkit from http://www.cryptopp.com/ (2002b). Also note that the additional costs due to selecting random locations and randomly choosing power values are not considered because they are negligible compared to cryptographic functions, multiplications, and exponentiations.

3.2.2. Privacy analysis

There are four confidential data items that should be protected. The first one is location information of surrounding points held by the server. Notice that to mask them, the server first creates N bogus location. Hence, it now owns $g = G + N$ points rather than G points. For the client, the probability of guessing the true G locations out of g locations is 1 out of $C\binom{g}{G}$, where $C\binom{X}{Y}$ represents the number of ways of picking Y unordered outcomes from X possibilities. Moreover, since the server hides each true location of G points among n-1 bogus locations, for the client, the probability of guessing one of them is 1 out of n. Then, the probability of guessing G points is 1 out of n^G. Thus, the probability of guessing the true G points held by

the server is 1 out of $C\binom{g}{G} \times n^G$. With increasing n and g (N respectively) values, the probability is becoming smaller. However, since increasing such values also increases supplementary costs, the server and the client should find a balance between efficiency and privacy level. The optimum values of n and N can be determined based on how much privacy and efficiency the client and the server want. Since efficiency is not that critical for the overall performance, they prefer having higher privacy level over performance. Also, the client can try to learn true locations from the received encrypted sum value $\xi_{KA}\left(\sum_{j=1}^{G}\left(1/\left(D_{ij}\right)^{m_j}\right)\right) = \prod_{j=1}^{G}\xi_{KA}\left(1/(D_{ij})^{m_j}\right)$. However, it is not possible to learn G unknown locations from an aggregate of G distances.

The second data item that the server wants to hide is measurements. The server uses HE property to calculate $\xi_{KA}\left(\sum_{j=1}^{G}\left(P_j/\left(D_{ij}\right)^{m_j}\right)\right) = \prod_{j=1}^{G}\xi_{KA}\left(1/(D_{ij})^{m_j}\right)^{P_j}$ values. Since the server sends an encrypted aggregate to the client, even if the client knows the decryption key and obtains the aggregate $\sum_{j=1}^{G}\left(P_j/\left(D_{ij}\right)^{m_j}\right)$ value, she cannot determine the unknown measurements for G points from one known aggregate value. In other words, it is not possible to derive G unknown measurements from one known value.

For the client, confidential data items are the final prediction and the location i. The server cannot learn the prediction from $\xi_{KA}\left(\sum_{j=1}^{G}\left(1/\left(D_{ij}\right)^{m_j}\right)\right) = \prod_{j=1}^{G}\xi_{KA}\left(1/(D_{ij})^{m_j}\right)$ and $\xi_{KA}\left(\sum_{j=1}^{G}\left(P_j/\left(D_{ij}\right)^{m_j}\right)\right) = \prod_{j=1}^{G}\xi_{KA}\left(1/(D_{ij})^{m_j}\right)^{P_j}$, because it does not know the corresponding decryption key, which is known by the client; and it is not able to learn m_j values, which are chosen uniformly randomly over a range by the client (corresponding θ values are known by the client only).

The location information for which the client is looking for a prediction is also private. The server should not learn that information either. Remember that the client uses varying m_j values determined uniformly randomly over a range (θ values known by the client) and encrypts $1/(D_{iz})^{m_z}$ values using HE property using her public key. Thus, the server cannot learn the location, because it does not know the m_j values and the corresponding decryption key, which is known by the client only.

In the proposed method, HE and OT are employed. Paillier (1999) shows that HE is semantically secure for inference of input values. In other words, the parties cannot derive any information from the exchanged encrypted values. Similarly, OT is also secure (prevents the client and the server from learning the selected data item and data values computed for all points other than the real one, respectively), as shown by (Noar and Pinkas, 1999). To sum up, the final solution achieves all four privacy requirements. It prevents the participating parties from deriving information about each other's confidential data. Any client and a server can employ the proposed method to perform IDW interpolation with privacy.

3.2.3. Accuracy analysis

As explained previously, accuracy, privacy, and efficiency are three conflicting goals. Therefore, due to privacy-preserving measures, accuracy might become worse. It is previously shown that privacy measures introduce overhead costs. In this section, the proposed scheme is scrutinized in terms of accuracy. How the proposed method affects the quality of the predictions is demonstrated in this section. Notice that HE and OT are employed, which do not affect accuracy. The controlling parameters that can affect the accuracy of the prediction is the range (θ_2 - θ_1) over which the client select m_j values uniformly randomly and number of surrounding points G. Thus, different sets of experiments are conducted using real data sets to show how varying range and G values affect accuracy.

3.3. Experiments

A set of experiments are run to show how varying θ values affect accuracy. Besides varying θ values, G is also another controlling parameter that might affect accuracy. Thus, another set of trials are performed to show how accuracy is affected with varying G values. For this purpose, three different θ values 0.05, 0.15 and 0.25 are used, while G values are varied from 5 to 50. After calculating RMSE and MAE values for both the Illinois and the Colorado data sets, overall outcomes are displayed in Table 3.1, Table 3.2, Table 3.3 and Table 3.4, respectively.

Table 3.1. Effects of varying θ and G values on RMSE (Illinois data set)

	θ/G	**5**	**10**	**15**	**20**	**30**	**40**	**50**
	Optimal	0.1162	0.1140	0.1136	0.1134	0.1126	0.1124	0.1124
Random	**±0.05**	0.1163	0.1141	0.1137	0.1135	0.1127	0.1125	0.1125
	±0.15	0.1165	0.1142	0.1138	0.1136	0.1129	0.1127	0.1126
	±0.25	0.1167	0.1144	0.1140	0.1138	0.1131	0.1129	0.1128

Table 3.2. Effects of varying θ and G values on MAE (Illinois data set)

	θ/G	**5**	**10**	**15**	**20**	**30**	**40**	**50**
	Optimal	0.0839	0.0823	0.0821	0.0821	0.0815	0.0814	0.0814
Random	**±0.05**	0.0840	0.0823	0.0821	0.0821	0.0815	0.0814	0.0815
	±0.15	0.0840	0.0824	0.0822	0.0822	0.0816	0.0815	0.0815
	±0.25	0.0841	0.0825	0.0823	0.0823	0.0818	0.0816	0.0816

As seen from Table 3.1 and Table 3.2, with increasing G values from 5 to 50, accuracy becomes better. For G values 30, 40, and 50, RMSE and MAE values are almost equal. As expected, adding random values to optimum power values worsen the accuracy. Adding a random number 0.25 to optimum power value when G is 50 increases the RMSE value by 0.44% and MAE value by 0.25%.

Table 3.3. Effects of varying θ and G values on RMSE (Colorado data set)

	θ/G	**5**	**10**	**15**	**20**	**30**	**40**	**50**
	Optimal	0.3190	0.3096	0.3117	0.3124	0.3148	0.3156	0.3159
Random	**±0.05**	0.3192	0.3097	0.3117	0.3124	0.3148	0.3156	0.3159
	±0.15	0.3195	0.3099	0.3118	0.3126	0.3150	0.3158	0.3161
	±0.25	0.3200	0.3103	0.3120	0.3128	0.3153	0.3161	0.3165

Table 3.4. Effects of varying θ and G values on MAE (Colorado data set)

	θ/G	**5**	**10**	**15**	**20**	**30**	**40**	**50**
	Optimal	0.2258	0.2176	0.2199	0.2202	0.2219	0.2226	0.2231
Random	**±0.05**	0.2259	0.2177	0.2199	0.2203	0.2219	0.2226	0.2231
	±0.15	0.2261	0.2179	0.2199	0.2203	0.2219	0.2226	0.2231
	±0.25	0.2264	0.2182	0.2201	0.2205	0.2220	0.2227	0.2231

As seen from Table 3.3 and Table 3.4, the minimum RMSE and MAE values are observed when G is 15. There is a steady decrease for G values from 5 to 15; however, RMSE and MAE values are getting worse after G is 15. Adding a random value 0.25 to optimum power value when G is 15 increases the RMSE value by 0.10% and MAE value by 0.09%.

3.4. Conclusion

In order to perform IDW interpolations on central data while preserving privacy, various schemes are proposed. The complete method is analyzed and it is shown that it preserves confidentiality. It is able to efficiently provide predictions with decent accuracy. Empirical outcomes show that accuracy losses due to the privacy measures used in the proposed schemes are negligible. Additionally, the final method is able to provide sufficient accuracy while preserving the server's and the client's privacy as expressed in privacy analysis section. Therefore, the companies, which are in role of either client or server, can use the proposed methods in order to achieve IDW-based predictions with privacy.

4. PRIVACY-PRESERVING KRIGING INTERPOLATION

Although kriging is widely used in many applications for prediction purposes, it fails to protect private data. Due to privacy risks, participating parties might not feel comfortable and they may decide not to involve in kriging interpolation. Hence, two methods are proposed, which allows the servers and the clients to perform kriging without divulging their confidential data to each other.

4.1. Proposed Schemes

As explained previously, *S* first needs to create a model (formula for determining semi variances-Eq. (2.3) and $\mathbf{\Gamma}^{-1}$ matrix) using its data for a given region *R* in order to estimate a prediction for any unmeasured location *q*. However, it needs *q*'s coordinates to generate matrix **g** so that it can estimate matrix $\boldsymbol{\lambda}$ and determine P_q. In the following, two schemes are described assuming that *S* has already created the model given *R*. In other words, how *S* creates **g**, estimates $\boldsymbol{\lambda}$, and determines P_q without jeopardizing privacy constraints are explained. In the following, naïve solution is first described and then enhanced method is explained in detail.

4.1.1. First solution: Naïve scheme

The proposed schemes' major concern is to protect confidential data of involving parties. Therefore, during kriging-based interpolation process, sample locations (their coordinates) and their related measurements and unmeasured location *q* (its coordinates) and the estimated prediction P_q should not be disclosed to *C* and *S*, respectively. The following naïve scheme is proposed in order to estimate kriging-based predictions without jeopardizing data owners' privacy. The naïve scheme is based on randomness (creating bogus locations) and OT. The steps of the naïve method can be listed as follows:

1. *S* first creates a model (formula for determining semi variances or Eq. (2.3) and $\mathbf{\Gamma}^{-1}$ matrix) using its data for the given region *R*.

2. *C* generates *n*-1 bogus locations in order to mask her real location.
3. She hides the unmeasured location *q* among such fake locations; and sends the coordinates of *n* locations including *q* to *S*.
4. For each location $j = 1, 2, \ldots, n$; *S* performs the following steps:
   a. First, it estimates distances between *j* and each measured location using Eq. (2.2).
   b. It then computes *semi variances* using Eq. (2.3).
   c. Next, it creates **g** matrix.
   d. It then estimates the weights using Eq. (2.5).
   e. Finally, it computes P_j using Eq. (2.6).
5. After estimating predictions for all *n* locations, *C* utilizes OT in order to get the prediction for her real location *q* only. Due to OT, which is shown to be secure (Noar and Pinkas, 1999), *S* cannot learn which prediction is obtained by *C*; and *C* cannot know other predictions rather than P_q. OT allows *C* learns one of the *n* inputs (P_q) held by *S* without learning anything about the other inputs and without allowing *S* to learn anything about *q*.

Due to bogus locations, *S* cannot learn the real location *q*. However, it can guess it with probability of 1/*n* because there *n* possibilities. With increasing *n*, such probability becomes smaller. Similarly, for *S*, the probability of guessing the estimated prediction is 1/*n* because it estimates predictions for *n* locations and one of them is for the real location. *S* does not want any client obtains more than one prediction during a single process due to financial reasons. Service suppliers provide estimated predictions in return of some benefits. To prevent *C* from receiving predictions for more than one location, OT is utilized. OT forces *C* to get estimated prediction for her real location only; and at the same time, it prevents *S* from learning which prediction is obtained by *C*. Due to aggregate outcome (estimated prediction), *C* cannot derive useful information about locations and their measurements held by *S* from received prediction.

In addition to the naïve scheme, the following scheme is also proposed, referred to as improved scheme (IS). Details of the IS are described in the following.

4.1.2. Second solution: Improved scheme

As explained previously, given *R*, *S* first can create a model using its data. It then needs to estimate distances between *q* and each sample point it holds in the region *R*. After that it is supposed to estimate semi variances in order to create **g** matrix. It finally needs to estimate P_q. The second scheme is based on HE. HE scheme proposed by Paillier (1999) is utilized to hide confidential data. If it is assumed that ξ is an encryption function and *K* is a public key, and x_{j1} and x_{j2} are private data values to be hidden, then Paillier's HE scheme allows to compute $\xi_K(X) = \prod_{j=1}^{n}(\xi_K(x_{j1}))^{x_{j2}}$ values. The steps of the IS are as follows:

1. The first step is calculating distances between *q* and each sample location. Such distances between *q* and each location *j* = 1, 2, …, *G* can be computed using Eq. (2.2) while preserving confidentiality as follows:
   a. Eq. (2.2) can be written as follows:

$$d_{jq} = \sqrt{(x_j - x_q)^2 + (y_j - y_q)^2} = \sqrt{x_j^2 + y_j^2 + x_q^2 + y_q^2 - 2 * (x_j x_q + y_j y_q)} = \sqrt{S_j + C_q - 2x_j x_q - 2y_j y_q} \quad (4.1)$$

   As seen from Eq. (4.1), *S* and *C* can compute S_j and C_q, respectively without needing each other. However, to estimate $x_j x_q$ and $y_j y_q$ values, they need to collaborate.

   b. Using HE scheme, *C* finds $\xi_{KC}(-2x_q)$, $\xi_{KC}(-2y_q)$, and $\xi_{KC}(x_q^2 + y_q^2)$ encrypted values, where note that ξ represents encryption function and *KC* is *C*'s public key.

   c. She then sends such encrypted values to *S*. Since the related private key is known by *C* only, *S* cannot learn x_q and y_q values.

   d. For each location *j* = 1, 2, …, *G*, using HE scheme, *S* determines $\xi_{KC}(D_{jq}) = \xi_{KC}(-2x_q)^{x_j} * \xi_{KC}(-2y_q)^{y_j} * \xi_{KC}(x_q^2 + y_q^2)^1 * \xi_{KC}(x_j^2 + y_j^2)^1 = \xi_{KC}(x_j^2 + y_j^2 + x_q^2 + y_q^2 - 2x_j x_q - 2y_j y_q)$.

   e. To find distances, square root of such encrypted values must be computed. Thus, *S* then sends $\xi_{KC}(D_{jq})$ values for all *j* = 1, 2, …, *G* to *C*.

f. *C* then decrypts $\xi_{KC}(D_{jq})$ values using the related private key, obtains D_{jq} values; and finds d_{jq} distance values between q and each location j by taking the square roots of D_{jq} values. Since D_{jq} values are aggregate values, *C* cannot learn the related x_j and y_j values from them. Even if she knows the distances, she cannot determine the true coordinates. She only learns that any location j is on the circle whose center is q and radius is d_{jq}.

g. Using HE scheme, *C* encrypts d_{jq} values using her public key *KC* and sends $\xi_{KC}(d_{jq})$ to *S*.

2. *S* finds estimated semi variances in encrypted form for location q using Eq. (2.4) and HE property; and creates the **g** matrix including the encrypted values, $\xi_{KC}(g_j)$ values for all $j = 1, 2, \ldots, G$.
3. Now, *S* needs to compute weights or λ values using Eq. (2.5). Using HE scheme, *S* computes $\xi_{KC}(\lambda_j) = \xi_{KC}(g_1)^{\gamma_{j1}} * \xi_{KC}(g_2)^{\gamma_{j2}} * \cdots * \xi_{KC}(g_G)^{\gamma_{jG}} = \xi_{KC}(g_1 * \gamma_{j1} + g_2 * \gamma_{j2} + \cdots + g_G * \gamma_{jG})$.
4. *S* then can estimate the prediction P_q using Eq. (2.6) as follows: $\xi_{KC}(P_q) = \xi_{KC}(\lambda_1)^{P_1} * \xi_{KC}(\lambda_2)^{P_2} * \cdots * \xi_{KC}(\lambda_G)^{P_G} = \xi_{KC}(\lambda_1 * P_1 + \lambda_2 * P_2 + \cdots + \lambda_G * \gamma P_G)$.
5. Finally, *S* sends $\xi_{KC}(P_q)$ to *C*. Due to encryption, *S* cannot know the estimated prediction.
6. Since *C* knows the related decryption key, she decrypts the received encrypted value and gets P_q. Due to aggregate estimation, *C* cannot derive information about *S*'s confidential data.

4.2. Analysis of the Improved Scheme

There are basically two evaluation criteria for prediction algorithms. They are called performance and accuracy. Performance means that how effectively a prediction algorithm can estimate predictions. It can be measured with respect to off-line and online costs like storage, computation, and communication (number of communications and amount of transferred data) costs. Hence, performance analysis can be done in terms of off-line and online costs. Compared to online costs,

off-line costs are not that critical for overall performance. Online efficiency requirements differ for various applications. For some applications, there are very hard online performance requirements. For example, recommender systems should be able to return many recommendations to their customers simultaneously in a very short time during an online interaction. However, performance requirements might be soft for some applications like geo-statistics. Online time limitations are not that rigid in geo-statistical predictions. For example, if one petroleum company looks for oil reserves in a given region, it might ask prediction from those that owns enough measurements in that region. Since investments in energy take some time and considerable amount of budgets, oil companies spend some time to get reliable and accurate predictions. Obtaining dependable and precise predictions is much more important than receiving predictions in a short time. Therefore, it can be said that online performance constraints are soft in kriging-based interpolations. Performance criterion covers the time needed to perform a single prediction, number of communications spent for a prediction (and/or amount of transferred data), and amount of storage space are needed. Resources spent for interpolations should be minimized for performance reasons.

The second criterion, accuracy, shows how accurate the estimated predictions are. Accuracy is measured in terms of the closeness between the estimated predictions and their true values. Estimated interpolations should be as close as to their observed values. Since predictions are estimated values based on available observed measurements, their values should be as close as possible to their expected values. Therefore, predictions generated by the proposed scheme with privacy concerns should be as close as possible to their true values.

In addition to performance and accuracy, privacy is another evaluation metric, which is used to investigate privacy-preserving prediction schemes. Privacy-preserving algorithms should be able to protect confidential data. Privacy requirements state that involving parties in interpolation processes cannot derive useful information about each other's private data. Thus, privacy, in this context, means that confidential data should be hidden to those but the intended parties. In other words, the proposed privacy-preserving scheme should be able to hide confidential data held by *S* and *C* against each other.

The enhanced method is analyzed in terms of privacy, accuracy, and efficiency. Additional costs due to privacy concerns are also scrutinized even though online performance requirements are not that rigid. Accuracy losses due to privacy-preserving measures are expected because accuracy and privacy are conflicting goals. Finally, it is shown that the scheme does not violate privacy constraints.

4.2.1. Accuracy analysis

In privacy-preserving prediction schemes, privacy measures usually make accuracy worse due to the conflicting nature of confidentiality and preciseness. However, in the proposed scheme, privacy-preserving methods do not cause any loss in accuracy. In other words, predictions estimated by the proposed method with privacy concerns are the same as the ones provided by traditional kriging scheme without confidentiality fears. Since cryptographic techniques are employed, which preserve data originality, accuracy is not affected. Thus, the proposed scheme is able to provide the same predictions while preserving confidentiality.

4.2.2. Performance analysis

The proposed scheme is investigated in terms of supplementary costs due to privacy concerns. Additional storage costs are first analyzed. The scheme does not cause any extra storage costs. Involving parties (*S* and *C*) do not need additional spaces required to save data caused by confidentiality measures. Thus, storage costs will not be affected by privacy concerns.

As shown in Figure 1.1, in a traditional kriging-based prediction process, number of communications is *two* only because *C* and *S* communicates two times only. However, number of communications in the proposed method increases due to privacy measures. As described in Figure 1.2, number of communications is *four* in the proposed scheme. In other words, number of communications increases *two times* due to privacy concerns. Amount of transferred data are also important. In a conventional interpolation, *C* sends coordinates of location *q* and *S* returns a

prediction. If it is assumed that four bytes are needed to save a coordinate and four bytes are enough to store an estimated prediction, then amount of sent data from *C* to *S* are about eight bytes while it is about four bytes from *S* to *C* in a traditional method. In the proposed scheme, *C* first sends three encrypted values to *S*. The size of an encrypted value is imperative. As explained in (Obviex, 2012), the size of an encrypted value produced by block cipher encryption can be computed as *size of plain text + block size* – (*size of plain text* **mod** *block size*). For example, if it is assumed that size of plain text is four bytes, block size is 16 bytes, and then 16 bytes are needed for an encrypted value. Thus, amount of sent data during this communication are about 48 bytes. After computing *G* encrypted aggregates, *S* then sends them to *C*. Assuming again that 16 bytes are needed for a single encrypted value, amount of sent data are about 16*G* bytes. During the second turn, *C* sends *G* encrypted values to *S*. Thus, amount of transferred data are again 16*G* bytes. And finally, *S* returns an encrypted value to *C*. Hence, amount of sent data are about 16 bytes. To sum up, like number of communications, amount of transferred data also increase due to privacy measures.

Supplementary computation costs caused by the proposed scheme are also inevitable. In addition to multiplications and additions, the method includes encryptions, decryptions, and exponentiations because of privacy measures. Number of encryptions is in the order of $O(G)$ and similarly, number of decryptions is in the order of $O(G)$. On the other hand, number of exponentiations is in the order of $O(G^2)$. Notice that *G* is a constant representing number of measured locations in the region *R*. Cryptographic functions are usually costly operations. In order to find out the running times of cryptographic operations, benchmarks for the CRYPTO++ toolkit from http://www.cryptopp.com/ can be used (Canny, 2002b).

Although the proposed scheme does cause some extra communication (in terms of number of communications and amount of transferred data) and computation costs, they are not critical due to the nature of kriging-based interpolation schemes. Unlike some real time applications, online performance requirements are softer for kriging-based prediction methods.

4.2.3. Privacy analysis

Our privacy requirements state that private data values should not be disclosed during prediction process. Notice that the measured locations (or their coordinates) and the related measurements are confidential for *S*. Similarly, the unmeasured location (or its coordinates) and the estimated prediction are private for *C*. The parties cannot learn each other's confidential data during the scheme. *C* sends her location coordinates in encrypted form rather than plain form. Since the related decryption key is known by *C* only, *S* cannot decrypt the received values and learn coordinates. After performing required computations using HE, *S* sends encrypted aggregates to *C*. Since *C* knows decryption key, she can decrypt the received values and find distances between *q* and each measured location. Although *C* learns the distances, she cannot learn the true coordinates. For each measured point *j*, given *q* and the related distance d_{jq}, the only information that *C* can derive is that *j* is in somewhere on the circle whose center is *q* and its radius is d_{jq}.

S returns the estimated prediction in encrypted form. Since the decryption key is known by *C* only, *S* cannot decrypt it and learn the prediction. When *C* obtains the prediction, which is an aggregate value, she cannot learn the measurements of *G* sample points. Paillier (1999) shows that HE is semantically secure for inference of input values. In other words, the parties cannot derive any information from the exchanged encrypted values. The proposed method prevents *C* from learning the measured location coordinates and their related measurements. It also prevents *S* from deriving useful information about the unmeasured location coordinates and the estimated prediction.

4.3. Conclusion

Obtaining geo-statistics data requires so much time and budget. Companies may be interested in specific locations of the region. It is meaningless to collect whole data for a vast region. There may be another company that already has collected measurements for the foregoing region. However, the company is not eager to give such valuable data whoever needs a prediction. The client also does

not want to publicize the coordinate, where it needs prediction. Therefore, a privacy-preserving solution is required for both parties. As given in details above, the proposed scheme claims that private data of the client and server are protected. It is also able to provide predictions efficiently without compromising on accuracy. As a result, both parties can use the proposed solution without any privacy consideration.

5. PRIVACY-PRESERVING IDW ON DISTRIBUTED DATA

Data collected for IDW interpolation purposes might be distributed between various parties, even competing ones. Measurements for some sample points might be partitioned between two parties only. Similarly, such measurements can be split among more than two parties. It is important to estimate IDW-based predictions based on integrated data while preserving data holders' confidentiality because accurate and dependable predictions can only be estimated from sufficient data. In this chapter, the methods, which are proposed to provide predictions from partitioned or distributed data with privacy are explained.

5.1. Privacy-Preserving IDW on Partitioned Data

Although it is possible to generate predictions from one of the parties' data, the results might not be reliable and accurate. It is more likely to produce a dependable and precise prediction from integrated data. Thus, data owners might decide to provide services on their combined data. However, data collected for prediction purposes are considered confidential and valuable assets. The companies often do not want to reveal such data.

Data owners consider collected measurements and their corresponding location coordinates private. *Q* might ask a prediction from the master server (MS) or *C*, which works together with the collaborating company *V*; and estimates the prediction without deeply violating data owners' privacy. In this context, privacy can be described as follows: *C* and *V* should not be able to learn the sample points' coordinates and their corresponding measurements while conducting IDW interpolations on integrated data. Moreover, the parties should not be able learn the location for which *Q* is seeking prediction and the estimated prediction, because they are considered confidential data items for *Q*. Hence, the problem is then how to conduct IDW interpolations on combined data while preserving the participating parties' privacy.

5.1.1. Naïve scheme

When privacy is not a concern, it is an easy job to perform IDW interpolations on partitioned data. *Q* sends the required data in plain form to *C*, which forwards the received data to *V*. After *C* and *V* compute necessary values (partial sums computed by each party using their own data to estimate P_i in Eq. (2.1)), *V* sends them back to *C* without any protection. *C* then combines the partial aggregates, finds P_i; and finally returns it back to *Q*. However, with privacy concerns, it becomes challenging to offer predictions on partitioned data. Our proposed naïve scheme (NS) helps the involving parties protect their confidential data while conducting IDW interpolations on partitioned data as follows:

1. *Q* uniformly randomly generates *n*-1 bogus locations.
2. She hides the location *i* among spurious locations. She simply permutes *n* locations.
3. She sends $I_{Qz} = (x_{Qz}, y_{Qz})$ values and her public key *KQ* to *C*, where *z* = 1, 2, …, *n*. Notice that *z* = *i* represents the true location.
4. *C* then sends such received data to *V*.
5. For each point I_{Qz}, *V* and *C* perform the followings:
   a. Both parties uniformly randomly select m_{zj} over the range $[m_o\text{-}\theta, m_o\text{+}\theta]$ in which θ is called performance parameter and m_o represents the optimum power value. Note that since the power value *m* is data dependent, the optimum power value m_o is utilized for prediction estimation.
   b. For each point I_{Qz}, *V* and *C* then estimate $\sum_{V1z} = \sum_{j=1}^{G_V} P_j/(D_{zj})^{m_{zj}}$ and $\sum_{V2z} = \sum_{j=1}^{G_V} 1/(D_{zj})^{m_{zj}}$; and $\sum_{C1z} = \sum_{j=1}^{G_C} P_j/(D_{zj})^{m_{zj}}$ and $\sum_{C2z} = \sum_{j=1}^{G_C} 1/(D_{zj})^{m_{zj}}$ values, respectively, where G_V and G_C are known by *V* and *C* only, respectively.
   c. *V* and *C* then find $\xi_{KQ}(\sum_{V1z})$ and $\xi_{KQ}(\sum_{V2z})$ and $\xi_{KQ}(\sum_{C1z})$ and $\xi_{KQ}(\sum_{C2z})$, respectively using an HE scheme.
   d. *V* sends the encrypted aggregates for all *n* points to *C*.
6. Using HE scheme, *C* finds $\xi_{KQ}(\sum_{V1z} + \sum_{C1z})$ and $\xi_{KQ}(\sum_{V2z} + \sum_{C2z})$ values for all *z* = 1, 2, …, *n* points.

7. *Q* uses OT and obtains the corresponding values for her real location *i*. Thus, she receives $\xi_{KQ}\ (\sum_{V1i} + \sum_{C1i})$ and $\xi_{KQ}\ (\sum_{V2i} + \sum_{C2i})$.
8. She then decrypts them using her corresponding private key *Kq* and obtains $\sum_{V1i} + \sum_{C1i}$ and $\sum_{V2i} + \sum_{C2i}$ values, which are required to estimate P_i.
9. She finally estimates $P_i = (\sum_{V1i} + \sum_{C1i})/(\sum_{V2i} + \sum_{C2i})$.

In this protocol, *C* can act as a *Q* in multiple scenarios to derive data from *V*. If it acts as *Q*, since it knows the encryption and decryption keys, it can obtain the plain values. It cannot derive useful information (measurements and their locations) from them, because such plain values are aggregates. Although *C* cannot derive information about *V*'s data from such aggregates, it can estimate *n* predictions rather than one during a single collaboration only. It means that it does not need *V* for further joint works. When *C* acts as *Q*, it might ask aggregates for *n* real locations. *V* thinks that one single prediction will be estimated while *C* estimates *n* predictions. Hence, *C* might not ask help from *V* for *n* predictions. Since they provide such services in return of some benefit, *V* does not want to lose such gains. To overcome this weakness, following protocol is proposed.

5.1.2. Enhanced scheme

Although enhanced scheme (ES) follows the similar steps, it overcomes the weakness that the NS faces. The steps of the improved scheme are as follows:

1. *Q* uniformly randomly generates *n*-1 bogus locations and hides her location *i* among them.
2. She sends $I_{Qz} = (x_{Qz}, y_{Qz})$ values to *C*, where $z = 1, 2, \ldots, n$. Notice that $z = i$ represents the true location.
3. *C* then forwards such received data to *V*.
4. For each point I_{Qz}, *V* and *C* perform the followings, assuming that *V*'s public key (*KV*) is also known by *C*.
   a. Both parties uniformly randomly select m_{zj} over the range $[m_o\text{-}\theta, m_o\text{+}\theta]$.
   b. For each point I_{Qz}, *V* and *C* then estimate $\sum_{V1z} = \sum_{j=1}^{G_V} P_j/(D_{zj})^{m_{zj}}$ and $\sum_{V2z} = \sum_{j=1}^{G_V} 1/(D_{zj})^{m_{zj}}$ values; and $\sum_{C1z} = \sum_{j=1}^{G_C} P_j/(D_{zj})^{m_{zj}}$ and

$\sum_{C2z} = \sum_{j=1}^{G_C} 1/(D_{zj})^{m_{zj}}$ values, respectively. Notice again that the parties do not know the numbers of sample points held by each other and involved in computations. In other words, G_V and G_C values are known by *V* and *C* only.

c. *V* and *C* then find $\xi_{KV}(\sum_{V1z})$ and $\xi_{KV}(\sum_{V2z})$; and $\xi_{KV}(\sum_{C1z})$ and $\xi_{KV}(\sum_{C2z})$, respectively, using an HE scheme.

d. *V* sends the encrypted aggregates for all *n* points to *C*.

5. Using HE scheme, *C* finds $\xi_{KV}(\sum_{V1z} + \sum_{C1z})$ and $\xi_{KV}(\sum_{V2z} + \sum_{C2z})$ values for all $z = 1, 2, \ldots, n$.
6. *Q* uses OT and obtains the corresponding encrypted aggregate values for *i*. Thus, she receives $\xi_{KV}(\sum_{V1i} + \sum_{C1i})$ and $\xi_{KV}(\sum_{V2i} + \sum_{C2i})$.
7. *Q* uniformly randomly generates two random numbers, *R1* and *R2*; and finds $\xi_{KV}(R1)$ and $\xi_{KV}(R2)$ using an HE scheme.
8. It adds them to the received encrypted aggregates using the homomorphic property, as follows: $\overline{Q1} = \xi_{KV}(\sum_{V1i} + \sum_{C1i}) * \xi_{KV}(R1) = \xi_{KV}(\sum_{V1i} + \sum_{C1i} + R1)$ and $\overline{Q2} = \xi_{KV}(\sum_{V2i} + \sum_{C2i}) * \xi_{KV}(R2) = \xi_{KV}(\sum_{V2i} + \sum_{C2i} + R2)$. It then sends $\overline{Q1}$ and $\overline{Q2}$ to *C*.
9. *C* forwards them to *V*, which decrypts them using its corresponding private key (*Kv*).
10. *V* sends $Q1 = \sum_{V1i} + \sum_{C1i} + R1$ and $Q2 = \sum_{V2i} + \sum_{C2i} + R2$ aggregates back to *C*, which forwards them to *Q*.
11. Since *Q* knows random numbers, it subtracts them from such received sums and gets $q1 = \sum_{V1i} + \sum_{C1i}$ and $q2 = \sum_{V2i} + \sum_{C2i}$ values.
12. It finally estimates $P_i = q_1/q_2$.

The improved method overcomes the weakness that the NS has. In other words, *V* prevents *C* from obtaining required data to estimate *n* predictions rather than one. Therefore, the ES eliminates all privacy, including geo-privacy, and financial concerns that the involving parties have. Since performance, privacy, and accuracy are three major goals that IDW interpolation methods are expected to achieve, the ES is analyzed in terms of these three goals. Notice that these three goals are conflicting with each other. Thus, utilizing privacy-preserving measures might make accuracy worse and introduce some extra costs affecting performance.

Also note that off-line costs are not critical like online costs; and moreover, unlike real-time applications, online performance is not that critical for IDW interpolation schemes.

5.2. Performance and Privacy Analysis

5.2.1. Performance analysis

The enhanced method is investigated with respect to additional costs like storage, communication, and computation costs. Supplementary storage spaces are needed due to randomly generated fake locations. Such extra storage costs are in the order of $O(n)$ because extra storage spaces are used for saving coordinates of n-1 random locations, randomly selected m_z values for n-1 fake locations, and encrypted aggregates for such bogus sample points.

Number of communications conducted during an IDW interpolation is also important. Since the scheme is based on partitioned data with privacy, data holders need to exchange data performing more communications. In a traditional scheme, number of communications is *two* only. In the proposed method, the parties perform *one* OT and *seven* communications. Note that OT could be achieved with poly-logarithmic (in n) communication complexity. Thus, due to privacy measures (especially OT), communication complexity increases from $O(1)$ to $O(n)$.

Without confidentiality, as seen from Eq. (2.1), C performs G number of multiplications, distance calculations, exponentiations, and $(G + 1)$ divisions. Hence, if addition is considered negligible, computation complexity is $O(G)$ only. In the proposed scheme, V performs $2n$ encryptions and two decryptions. Similarly, C conducts $(2n + 2)$ encryptions. Q also performs four encryptions. Hence, number of encryptions and decryptions are $(4n + 6)$ and two, respectively. Since the computations are repeated for n locations, the parties perform nG multiplications, exponentiations, and divisions. Hence, without cryptographic computations, the proposed scheme's computation complexity is in the order of $O(nG)$, which increases by n times due to bogus locations. Encryptions' and decryptions' running times can be determined using the benchmarks for the CRYPTO++ toolkit from

http://www.cryptopp.com/ (Canny, 2002b). Since computation costs due to choosing random locations and power values, and addition are insignificant compared to cryptographic operations, additional costs due to them are omitted.

5.2.2. Privacy analysis

The proposed scheme should prevent the involving parties from deriving information about each other's confidential data. It is investigated to show whether it is able to protect private data of *Q*, *V*, and *C* against each other.

First of all, the method is analyzed in terms of *Q*'s privacy. *Q* wants to hide *i* and P_i from both *C* and *V*. To mask her real location, *Q* creates *n*-1 bogus locations; and sends *n* rather than one location to *C*, which forwards them to *V*. Although *C* and *V* do not know the real location *i*, the probability of guessing it is 1 out of *n*. Remember that *Q* adds random numbers, *R*1 and *R*2, to partial encrypted aggregates to prevent *C* and *V* from learning P_i. Since such bogus values are known by *Q* only, they cannot derive information about the partial sums q_1 and q_2. Recall that $q1 = \sum_{V1i} + \sum_{C1i}$ and $q2 = \sum_{V2i} + \sum_{C2i}$, which are required to estimate P_i in the ES as follows: $P_i = q_1/q_2$. To learn P_i with 1 out of *n* probability, they must collude. In other words, they need to exchange partial aggregates for all *n* points. If one of the parties acts as *Q*, it might obtain partial aggregates necessary for future predictions for *n* locations; and it does not need the collaborating party for estimating predictions for such locations. Thus, one of them might lose competitive edge over the other.

V hides the measurements of sample points and their corresponding locations against *C* and *Q*. To do so, it first adds some randomness to m_o values by uniformly randomly selecting m_{zj} values over the range $[m_o\text{-}\theta, m_o\text{+}\theta]$. If it selects m_{zj} values around m_o for $j = 1, 2, \ldots, G_V$ and sets θ to a small value, then it can ensure privacy while providing accurate predictions. Since smaller θ values keep m_{zj} values around m_o while still add randomness to optimal power values, accuracy losses becomes lesser due to such randomness. The value of the performance parameter θ can be determined based on accuracy and privacy levels that the involving parties want to achieve. The main motivation behind uniformly randomly selecting m_{zj} values

around m_o is to prevent the parties from learning the individual sample points and the measurements from partial aggregates without compromising much on accuracy. Hence, since *V* uniformly randomly chooses m_{zj} values, the other party cannot figure out the individual sample points and the real measurements from partial aggregates, which are estimated based on variable m_{zj} values. Second, they cannot know how many sample points are involved in computing partial sums, because G_V values are known by *V* only. Finally, after calculating partial aggregates using variable m_{zj} and G_V values for all *n* locations, *V* encrypts them using its public key. Since the corresponding private key is required to decrypt such values and that key is known by *V* only, *Q* and *C* cannot learn partial sums sent by *V*. Without learning such partial aggregates, it is impossible to derive information about the sample locations and their measures. Even if they learn such partial sum values, *Q* and *C* cannot learn G_V unknowns from one known value. Thus, our scheme protects *V*'s privacy.

C also does not want to disclose its sample locations and their measurements to *Q* or *V*. To protect its confidential data, *C* also uses variable m_{zj} values, which are uniformly randomly chosen over a range. It also utilizes inconstant numbers of sample points, G_C values, for each *n* location. It computes encrypted partial sums after it receives required data from *V*. From such aggregates, it is not possible to determine confidential data. Also note that *Q* uses OT to obtain the results for *i* only. OT prevents her from learning the results for other locations. Even if *V* acts as *Q* to derive information about *C*'s private data, it cannot figure out measurements and their locations from a single known value.

Notice also that the ES uses HE and OT as privacy-preserving measures. Its privacy then depends on their privacy. Recall that Paillier (1999) shows that HE is secure and prevents malicious parties from deriving any information from the exchanged encrypted values. As shown by Naor and Pinkas (1999) and Cachin et al. (1999), OT is also secure and prevents *Q* from obtaining the results for bogus locations and avoids *C* from learning the values that *Q* receives.

5.2.3. Accuracy analysis: Experiments

To assess the scheme in terms of accuracy, different sets of experiments are conducted using real data. The goal of collaboration between those companies with insufficient data is to improve accuracy. Thus, the experiments are performed to show how collaboration affects precision. Moreover, another set of trials are conducted to investigate how privacy measures affect accuracy.

5.2.3.1. Effects of collaboration on accuracy

The first set of trials are performed to verify the effects of collaboration. It is hypothesized that predictions on combined data are expected to be more accurate than the ones on split data only. To verify this hypothesis and show how much accuracy is improved if two parties decide to estimate predictions on their integrated data, real data-based experiments are conducted. Consider the data partitioning scenario depicted in Figure 1.2, where G is assumed to be 5 and the measurements for some sample locations are partitioned between C and V. If the party C wants to estimate P_i based on its data only, the nearest neighbors of the location i are S_{C1}, S_{C2}, S_{C4}, S_{C8}, and S_{C9}. Similarly, when V wants to estimate P_i based on its data only, the nearest neighbors of the location i are S_{V1}, S_{V2}, S_{V5}, S_{V6}, and S_{V7}. Hence, due to insufficient data, each party includes sample points that are further away to estimate predictions. However, if they collaborate, the nearest neighbors of the location i will be S_{V2}, S_{C4}, S_{V6}, S_{C8}, and S_{C9}. In case of collaboration, thus, those sample points closer to the point of interest are used for interpolation, which give more accurate results.

In the experiments, G values are varied from 5 to 50 to show how varying total amount of data, partitioned between two parties, affects accuracy. It is assumed that the training data are evenly partitioned between two parties. Predictions are first estimated using each party's data only. Then, their errors are averaged and displayed as split data results. Next, all train data are used as integrated data and predictions are provided for the same test data. Finally, their errors are averaged and displayed as integrated data results. The MAEs and RMSEs are presented in

Table 5.1, Table 5.2, Table 5.3, and Table 5.4 in which the different columns represent different *G* values.

Table 5.1. Effects of collaboration on RMSE (Illinois data set)

	5	**10**	**15**	**20**	**30**	**40**	**50**
Integrated	0.1162	0.1140	0.1136	0.1134	0.1126	0.1124	0.1124
Split	0.1250	0.1220	0.1212	0.1207	0.1201	0.1200	0.1200

Table 5.2. Effects of collaboration on MAE (Illinois data set)

	5	**10**	**15**	**20**	**30**	**40**	**50**
Integrated	0.0839	0.0823	0.0821	0.0821	0.0815	0.0814	0.0814
Split	0.0902	0.0886	0.0880	0.0877	0.0875	0.0875	0.0876

Table 5.3. Effects of collaboration on RMSE (Colorado data set)

	5	**10**	**15**	**20**	**30**	**40**	**50**
Integrated	0.3190	0.3096	0.3117	0.3124	0.3148	0.3156	0.3159
Split	0.3318	0.3268	0.3260	0.3263	0.3271	0.3282	0.3290

Table 5.4. Effects of collaboration on MAE (Colorado data set)

	5	**10**	**15**	**20**	**30**	**40**	**50**
Integrated	0.2258	0.2176	0.2199	0.2202	0.2219	0.2226	0.2231
Split	0.2360	0.2332	0.2329	0.2336	0.2346	0.2354	0.2358

The results verify the hypothesis. Quality of predictions estimated on integrated data is better than the ones on split data only. For example, when *G* is 5, accuracy improves from 0.1200 to 0.1125 when *G* is 50 for the Illinois data set in terms of RMSE. Similar improvements are observed for all *G* values in both data sets. Due to such gains, if data owners decide to work in pairs, they can offer more powerful prediction services. Such improvements make collaboration attractive.

5.2.3.2. Effects of unevenly partitioned data

It is assumed that data are equally split between two companies. However, data might be unevenly partitioned. Thus, trials are done to assess the effects of unevenly partitioned data. For this purpose, β is defined as splitting percentage. It is assumed that β_C percent of the sample points are held by C and the remaining are held by V. Notice that when β_C is 50, data are evenly partitioned. After computing MAE and RMSE values for varying β_C values, they are displayed in Table 5.5, Table 5.6, Table 5.7, and Table 5.8 in which the different rows represent different β_C values. The related outcomes for integrated data are displayed in the last row.

Table 5.5. Effects of unevenly partitioned data on RMSE (Illinois data set)

β_C (%)/G	5	10	15	20	30	40	50
20	0.1375	0.1328	0.1309	0.1301	0.1294	0.1295	0.1296
35	0.1328	0.1297	0.1282	0.1275	0.1269	0.1266	0.1264
50	0.1250	0.1220	0.1212	0.1207	0.1201	0.1200	0.1200
65	0.1231	0.1202	0.1196	0.1189	0.1184	0.1182	0.1182
80	0.1187	0.1163	0.1158	0.1152	0.1145	0.1144	0.1143
100	0.1162	0.1140	0.1136	0.1134	0.1126	0.1124	0.1124

Table 5.6. Effects of unevenly partitioned data on MAE (Illinois data set)

β_C (%)/G	5	10	15	20	30	40	50
20	0.1020	0.0983	0.0969	0.0966	0.0965	0.0970	0.0972
35	0.0966	0.0942	0.0928	0.0927	0.0925	0.0924	0.0924
50	0.0902	0.0886	0.0880	0.0877	0.0875	0.0875	0.0876
65	0.0886	0.0868	0.0864	0.0858	0.0856	0.0856	0.0857
80	0.0862	0.0844	0.0842	0.0839	0.0834	0.0835	0.0835
100	0.0839	0.0823	0.0821	0.0821	0.0815	0.0814	0.0814

Table 5.7. Effects of unevenly partitioned data on RMSE (Colorado data set)

β_C (%)/G	5	10	15	20	30	40	50
20	0.3507	0.3428	0.3407	0.3406	0.3417	0.3430	0.3442
35	0.3411	0.3366	0.3353	0.3351	0.3363	0.3375	0.3381
50	0.3318	0.3268	0.3260	0.3263	0.3271	0.3282	0.3290
65	0.3272	0.3186	0.3180	0.3190	0.3200	0.3209	0.3220
80	0.3214	0.3140	0.3147	0.3158	0.3174	0.3178	0.3187
100	0.3190	0.3096	0.3117	0.3124	0.3148	0.3156	0.3159

Table 5.8. Effects of unevenly partitioned data on MAE (Colorado data set)

β_C (%)/G	5	10	15	20	30	40	50
20	0.2538	0.2479	0.2473	0.2476	0.2489	0.2505	0.2515
35	0.2428	0.2391	0.2383	0.2385	0.2395	0.2404	0.2410
50	0.2360	0.2332	0.2329	0.2336	0.2346	0.2354	0.2358
65	0.2324	0.2258	0.2257	0.2266	0.2275	0,2283	0.2290
80	0.2281	0.2221	0.2229	0.2234	0.2246	0.2252	0.2258
100	0.2258	0.2176	0.2199	0.2202	0.2219	0.2226	0.2231

As seen from the tables, data owners holding smaller amount of measurements gain more benefits due to collaboration. With decreasing amount of data (decreasing β_C for C), accuracy becomes worse. On the other hand, correctness improves for V because while C's data is decreased, V's data is augmented. Therefore, those holding lesser amount of measurements are expected to be more eager for collaboration. As seen from the tables, the outcomes based on integrated data are better than the outcomes on split data for all β_C values. Hence, although accuracy advantages for the party having greater amount of data seems to be smaller, it still gets gains due to collaboration.

5.2.3.3. Effects of masking optimum power values

Remember that optimum power values are masked by random values. Instead of using m_o values, the parties choose m_{zj} values uniformly randomly over the range

[m_o-θ, m_o+θ]. Since randomness is added into power values; thus, finally another set of experiments are performed to show how varying θ values or amounts of randomness affect accuracy. θ values are changed from 0.05 to 0.25 to show their effects. Similar trends are observed with respect to RMSE and MAE values. Thus, RMSE and MAE values for G is 50 or 15 for the Illinois and the Colorado data sets, respectively are displayed in Table 5.9, Table 5.10, Table 5.11, and table 5.12, where minimum values are observed for both sets.

Table 5.9. Effects of masking optimum power values on RMSE (Illinois data set)

β_C (%)/θ	**0.05**	**0.15**	**0.25**
20	0.1297	0.1299	0.1303
35	0.1265	0.1267	0.1269
50	0.1201	0.1204	0.1207
65	0.1182	0.1184	0.1187
80	0.1144	0.1146	0.1148
100	0.1125	0.1126	0.1128

Table 5.10. Effects of masking optimum power values on MAE (Illinois data set)

β_C (%)/θ	**0.05**	**0.15**	**0.25**
20	0.0972	0.0973	0.0975
35	0.0925	0.0926	0.0927
50	0.0876	0.0877	0.0878
65	0.0857	0.0858	0.0859
80	0.0835	0.0836	0.0837
100	0.0815	0.0815	0.0816

Table 5.11. Effects of masking optimum power values on RMSE (Colorado data set)

β_C (%)/θ	**0.05**	**0.15**	**0.25**
20	0.3429	0.3433	0.3437
35	0.3367	0.3369	0.3372
50	0.3269	0.3273	0.3278
65	0.3187	0.3190	0.3195
80	0.3140	0.3142	0.3145
100	0.3097	0.3099	0.3103

Table 5.12. Effects of masking optimum power values on MAE (Colorado data set)

β_C (%)/θ	**0.05**	**0.15**	**0.25**
20	0.2480	0.2484	0.2488
35	0.2391	0.2391	0.2393
50	0.2332	0.2334	0.2336
65	0.2259	0.2261	0.2263
80	0.2221	0.2222	0.2223
100	0.2177	0.2179	0.2182

Due to privacy concerns, accuracy is expected to become worse because privacy and accuracy are conflicting goals. As seen from the tables, accuracy slightly becomes worse due to the randomness added to optimum power values. However, the results based on the proposed scheme are better than the ones on split data only. Due to privacy concerns, accuracy losses are very small. Moreover, with increasing *G* values, accuracy losses become stable and smaller. Accuracy gains due to collaboration compensate accuracy losses due to privacy-preserving measures. Therefore, the proposed method helps data owners provide accurate IDW-based predictions on combined data while protecting their confidentiality and query owner's privacy.

5.3. Private IDW on Distributed Data

The problem for which a solution is proposed can be described as follows: *How to perform IDW interpolations if measurements are distributed among multiple parties while preserving all involving parties' privacy*? The participating parties include Q, master party C_1 from which Q asks a prediction, and helping parties C_2, C_3, …, C_M, where M represents number of collaborating companies including C_1. The steps of the proposed scheme are as follows:

1. In order to prevent involving companies (C_1, C_2, …, C_M) from learning her real location i, Q first uniformly randomly creates n-1 bogus locations. It then hides her actual location i among them; and sends n locations including the real one to C_1.
2. C_1 forwards n locations to collaborating companies (C_2, C_3, …, C_M) agreed to join distributed data-based IDW interpolation.
3. For each location s = 1, 2, …, n, each participating company $C = C_1, C_2, \ldots, C_M$ performs the followings:
   a. Uniformly randomly selects m_{sj} over the range $[\theta_l, \theta_u]$ in which θ_l and θ_u represent the lower and upper bound of power value m, respectively.
   b. Computes $\sum_{Csn} = \sum_{j=1}^{G_C} P_j/(D_{sj})^{m_{sj}}$ and $\sum_{Csd} = \sum_{j=1}^{G_C} 1/(D_{sj})^{m_{sj}}$ values required for determining the values of nominator and denominator in Eq. (2.1), respectively. Notice that G_C value is known by C only and number of sample points involving in prediction estimation might be different for each party.
   c. Encrypts the calculated values using an HE scheme with its public key; and obtains $\xi_{KC}\left(\sum_{Csn} = \sum_{j=1}^{G_C} P_j/(D_{sj})^{m_{sj}}\right)$ and $\xi_{KC}\left(\sum_{Csd} = \sum_{j=1}^{G_C} 1/(D_{sj})^{m_{sj}}\right)$ values. Note that KC is C's public key and the related private key is known by C only. HE scheme proposed by Paillier (1999) allows data owners to conduct multiplication and addition based on encrypted values without decrypting them as follows: Assume that X_i and Y_i represent some confidential data. Then, the HE scheme helps data

holders compute $\prod_{i=1}^{n} \xi_K(Y_i)^{X_i} = \xi_K(\sum_{i=1}^{n} X_i \times Y_i)$ in which *K* represents public key and ξ is HE function.

4. After computing encrypted partial sums for *n* locations (required fractional aggregates for calculating the denominator and nominator of Eq. (2.1)), each collaborating party sends such masked quantities for *n* locations to *C1*.
5. *Q* utilizes OT and receives the encrypted partial sum values for her real location *i*. Through OT, *Q* gets what it needs only without letting *C1* knows and *C1* (accordingly all collaborating parties) prevents *Q* from obtaining more than it is supposed to get.
6. *Q* uniformly randomly generates bogus data values, *RJ* and *VJ* values, for perturbing partial sums Σ_{Csn} and Σ_{Csd} values, respectively, where $J = 1, 2, \ldots, M$.
7. She adds such random values to corresponding partial sums using HE property and obtains $\xi_{KC}\left(\sum_{j=1}^{G_C} P_j/(D_{sj})^{m_{sj}} + RJ\right)$ and $\xi_{KC}\left(\sum_{j=1}^{G_C} 1/(D_{sj})^{m_{sj}} + VJ\right)$.
8. She then sends them to the master company *C1*, which forwards the corresponding values to related cooperating parties.
9. Each collaborating party *C* decrypts the received encrypted partial sums using its corresponding private key and gets $\sum_{j=1}^{G_C} P_j/(D_{sj})^{m_{sj}} + RJ$ and $\sum_{j=1}^{G_C} 1/(D_{sj})^{m_{sj}} + VJ$ values. Then, each party encrypts the results using *Q*'s public key (*KQ*), finds $\xi_{KQ}\left(\sum_{j=1}^{G_C} P_j/(D_{sj})^{m_{sj}} + RJ\right)$ and $\xi_{KQ}\left(\sum_{j=1}^{G_C} 1/(D_{sj})^{m_{sj}} + VJ\right)$; and finally such values are gathered in *C1*.
10. After collecting such encrypted values from collaborating parties, *C1* returns them, including the one found by itself, back to *Q*.
11. Since *Q* knows the corresponding decryption key, she first decrypts them and gets $\sum_{j=1}^{G_C} P_j/(D_{sj})^{m_{sj}} + RJ$ and $\sum_{j=1}^{G_C} 1/(D_{sj})^{m_{sj}} + VJ$ values. Notice that she also knows the added random values, *RJ* and *VJ* values. She subtracts corresponding random values from related partial disguised sums; and obtains $\sum_{j=1}^{G_C} P_j/(D_{sj})^{m_{sj}}$ and $\sum_{j=1}^{G_C} 1/(D_{sj})^{m_{sj}}$ values.

12. She adds $\sum_{j=1}^{G_C} P_j/(D_{sj})^{m_{sj}}$ values up and gets $\sum_{j=1}^{G}\left[P_j/\left(D_{ij}\right)^m\right]$. She also adds $\sum_{j=1}^{G_C} 1/(D_{sj})^{m_{sj}}$ values up and obtains $\sum_{j=1}^{G}\frac{1}{(D_{ij})^m}$.

13. She finally finds $P_i = \frac{\sum_{j=1}^{G}\left[P_j/(D_{ij})^m\right]}{\sum_{j=1}^{G}\frac{1}{(D_{ij})^m}}$, where participating parties do not know the estimated prediction.

5.4. Performance and Privacy Analysis

The aim of the proposed method is to achieve three conflicting goals: privacy, accuracy, and performance. Hence, it is scrutinized in terms of supplementary costs, which affect overall performance, privacy, and accuracy.

5.4.1. Performance analysis

Supplementary costs are inevitable because privacy and performance conflict with each other. Due to privacy-preserving measures, additional costs are expected. Such overheads can be storage, communication, and computation costs. It is important to analyze the scheme in terms of extra costs due to confidentiality.

In a traditional IDW interpolation, *Q* sends location information to *C1*; and *C1* estimates a prediction using Eq. (2.1) and returns it to *Q*. In the proposed method, *Q* creates bogus locations and sends *n* location information to *C1* forwarding them to helping parties. Partial results are computed for *n* locations rather than single one. *C1* must save fractional sums for *n* locations. Moreover, *Q* saves 2*M* random numbers used to hide partial aggregates. Thus, storage costs increase due to privacy concerns by in the order of *O* (*n*).

Number of communications increases as a result of privacy-preserving measures in the recommended scheme. Without privacy concerns, since data owners can integrate their data, only *two* communications are needed to get a prediction. In other words, *Q* sends a message to *C1* asking a prediction for location *i* and *C1* returns the estimated prediction. In the scheme, *Q* utilizes one OT, which could be achieved with poly-logarithmic (in *n*) communication complexity. In

addition to OT, *C1* exchanges data with helping parties and *Q*, resulting (4*M* – 1) numbers of communications, where notice that *M* shows the number of collaborating parties. Since *M* is much smaller than *n* and it is usually a small constant, communication overheads occur due to OT. Hence, number of communications increases about by *O* (*n*) times.

The proposed method causes extra computations costs, as well. Compared to multiplications, divisions, exponentiations, and cryptographic calculations, computation costs due to generating random locations and numbers, additions, and subtractions are negligible. Hence, in the suggested approach, supplementary computation costs performed by *Q* like creating (*n* – 1) bogus locations and generating 2*M* random values; and performing 2*M* subtractions are considered insignificant. Similarly, overheads due to determining power values randomly are also trivial. Collaborating parties conduct multiplications, exponentiations, and distance computations in a traditional IDW interpolation, as seen from Eq. (2.1). Although they perform such computations for one location only when privacy is not a concern, they do the same calculations for *n* locations rather than one in the recommended scheme. Therefore, computation costs, consisting of multiplications, exponentiations, and distance computations, increase by in the order of *O* (*n*). In addition to such additional costs, the proposed method also includes cryptographic calculations like encryption and decryption. There are 4*M* encryptions and 2*M* decryptions. Running times of encryptions and decryptions can be determined using the benchmarks for the CRYPTO++ toolkit from http://www.cryptopp.com/ (Canny, 2002b).

As expected, privacy measures cause some additional costs. Storage costs can be handled easily due to increasing hardware technology. Communication and computation costs are more critical for overall performance. As stated previously, unlike real-time applications, online efficiency is not that critical in IDW interpolation. Therefore, supplementary costs caused by the recommended scheme are tolerable. The method helps data owners perform IDW interpolation to estimate accurate predictions efficiently.

5.4.2. Privacy analysis

In order to call the recommended method as a secure one, it should satisfy all involving parties' privacy requirements. Participating parties can be listed as *Q*, master party *C1*, and helping companies *C2*, *C3*, …, *CM*. First of all, it should be shown how the method protects *Q*'s privacy. Notice that *Q* hides her real location *i* and the estimated prediction P_i. Collaborating parties cannot know the location *i* due to bogus locations. However, they can guess it because it is hidden among (*n* – 1) random locations. Thus, for cooperating companies, the probability of guessing *i* is 1 out of *n*. Although with increasing *n* values, the probability decreases, supplementary costs increase. Participating parties can determine the value of *n* based on their privacy and performance requirements. In order to prevent the cooperating companies from learning P_i, *Q* creates random values (*RJ* and *VJ* values) and adds them to the related fractional sums computed for nominator and denominator parts in Eq. (2.1) by each party. Since the range over which random numbers are generated, the bogus data, and the real location *i* are known by *Q* only, the parties cannot learn the partial sums and P_i, accordingly. Even if the parties collude to guess the estimated probability (share partial sums for *n* locations in plaintext form), the probability is still 1/*n*. However, if they conspire to guess one estimated prediction, one of the parties acting maliciously can get required data for computing predictions for *n* real locations without paying any money, which violates their financial constraints.

Collaborating parties including the master party hide their measurements and the related locations against *Q* and each other. *Q* cannot learn measurements and their locations due to the following reasons: First, it receives aggregates of G_C measurements from each party. Given one value, it is not possible to learn G_C unknowns. Second, *Q* does not know G_C values, which are known by the related party *C* only. Third, received values are encrypted and *Q* cannot decrypt them, because it does not know the corresponding decryption keys. And finally, *Q* utilizes OT, which prevents it from learning more data than it needs. It is shown by Even et al. (1985), Brassard et al. (1987), Naor and Pinkas (1999), and Cachin et al. (1999),

OT prevents *Q* from obtaining the results for bogus locations and allows it get the partial sums for *i* only.

Cooperating parties other than *C1* try to prevent *C1* from deriving information about their confidential data. To achieve their privacy, first of all, they utilize variable power values rather than fixed ones. Since the master party or any party acting as *Q* can try to derive information through multiple IDW interpolation processes, the parties use varying *m* values, selected uniformly randomly over a specified range, for each sample point in every IDW interpolation process. Secondly, they send aggregate results rather than single values. It is not possible to learn more than one unknown from one cumulative value. Thirdly, number of sample points involved in prediction estimation process is known by the related party only; and moreover, that value changes for different locations. Finally and the most importantly, the parties encrypt their cumulative quantities using their public keys. In order to decrypt them, corresponding private keys are required, where they are known by the related parties only. It is shown by Paillier (1999) that HE prevents malicious parties from deriving any information from the exchanged encrypted values.

The same reasons can be listed for explaining why the cooperating parties cannot learn each other's data. Encryption, variable *m* values, exchanging summative values, and variable and unknown number of samples, in general, help data holders protect their privacy.

5.4.3. Accuracy analysis: Experiments

Distributed data-based computations are expected to provide more accurate results. Thus, it is hypothesized that predictions on integrated data are more accurate than the ones on split data only. Since privacy and accuracy are conflicting goals, it is expected that the quality of the predictions diminishes due to our privacy-preserving measures. However, such losses should be compensated by the gains due to collaboration. Furthermore, there are some controlling parameters like varying power values uniformly randomly selected over a range and number of sample points that can affect accuracy of our privacy-preserving scheme. Therefore,

various sets of experiments are conducted using real data to show how varying parameters affect correctness.

5.4.3.1. Effects of collaboration on accuracy

A set of experiments are run first to verify the premise about collaboration. To do so, given an entire training set, it is assumed that measurements are distributed among *M* parties, where *M* is varied from 1 to 5 (set it to 1, 3, and 5). If *M* = 1, then it means that data are held by a single company. Optimum power values are utilized for each *G* value. Overall averages are calculated and the final RMSE and MAE values for the Illinois data set are displayed in Table 5.13 and Table 5.14, respectively. The results for the Colorado data set are presented in Table 5.14 and Table 5.15.

Table 5.13. Effects of collaboration on RMSE (Illinois data set)

G	**5**	**10**	**15**	**20**	**30**	**40**	**50**
Integrated	0.1162	0.1140	0.1136	0.1134	0.1126	0.1124	0.1124
3-Party	0.1314	0.1285	0.1270	0.1263	0.1259	0.1256	0.1255
4-Party	0.1335	0.1291	0.1281	0.1276	0.1270	0.1270	0.1272
5-Party	0.1366	0.1325	0.1309	0.1301	0.1295	0.1296	0.1297

Table 5.14. Effects of collaboration on MAE (Illinois data set)

G	**5**	**10**	**15**	**20**	**30**	**40**	**50**
Integrated	0.0839	0.0823	0.0821	0.0821	0.0815	0.0814	0.0814
3-Party	0.0954	0.0932	0.0919	0.0918	0.0916	0.0917	0.0917
4-Party	0.0979	0.0951	0.0948	0.0946	0.0944	0.0946	0.0948
5-Party	0.1019	0.0986	0.0974	0.0971	0.0970	0.0974	0.0975

Table 5.15. Effects of collaboration on RMSE (Colorado data set)

G	**5**	**10**	**15**	**20**	**30**	**40**	**50**
Integrated	0.3190	0.3096	0.3117	0.3124	0.3148	0.3156	0.3159
3-Party	0.3417	0.3368	0.3355	0.3355	0.3368	0.3378	0.3385
4-Party	0.3457	0.3375	0.3368	0.3382	0.3402	0.3417	0.3430
5-Party	0.3495	0.3415	0.3398	0.3400	0.3410	0.3424	0.3435

Table 5.16. Effects of collaboration on MAE (Colorado data set)

G	**5**	**10**	**15**	**20**	**30**	**40**	**50**
Integrated	0.2258	0.2176	0.2199	0.2202	0.2219	0.2226	0.2231
3-Party	0.2440	0.2400	0.2395	0.2397	0.2407	0.2414	0.2421
4-Party	0.2482	0.2429	0.2437	0.2447	0.2462	0.2475	0.2486
5-Party	0.2527	0.2470	0.2465	0.2468	0.2480	0.2495	0.2504

As seen from the tables, the results are getting better with decreasing *M* values. These outcomes verify the hypothesis. In other words, collaboration definitely enhances accuracy. When data are distributed among 5 parties and they offer predictions based on their split data only, RMSE value is about 0.1297 when *G* is 50 for the Illinois data set and 0.3398 for the Colorado data set when *G* is 15. However, if they decide to provide IDW interpolation services on their integrated data, they are able to achieve RMSE of 0.1124 and 0.3117, respectively. In other words, accuracy enhances by about 15% for the Illinois and 9% for the Colorado data set due to collaboration. To sum up, the parties are able to provide more accurate outcomes if they offer predictions on combined data.

5.4.3.2. Effects of masking optimum power values

After verifying the hypothesis, another set of experiments are performed for evaluating the suggested scheme in terms of privacy measures. Notice that privacy and accuracy are conflicting goals. Therefore, due to privacy-preserving measures, accuracy losses are expected. It is shown how privacy measures affect accuracy in the following. For this purpose, trials are done while changing θ values. RMSE

values are computed for the Illinois and Colorado data sets and the overall averages are presented in Table 5.17 and Table 5.19. MAE values for both data sets are also displayed in Table 5.18 and Table 5.20.

Table 5.17. Effects of masking optimum power values on RMSE (Illinois data set)

	0.05	**0.15**	**0.25**
Integrated	0.1125	0.1126	0.1128
3-Party	0.1255	0.1257	0.1260
4-Party	0.1273	0.1274	0.1276
5-Party	0.1297	0.1300	0.1303

Table 5.18. Effects of masking optimum power values on MAE (Illinois data set)

	0.05	**0.15**	**0.25**
Integrated	0.0815	0.0815	0.0816
3-Party	0.0917	0.0918	0.0919
4-Party	0.0948	0.0949	0.0950
5-Party	0.0976	0.0977	0.0979

Table 5.19. Effects of masking optimum power values on RMSE (Colorado data set)

	0.05	**0.15**	**0.25**
Integrated	0.3097	0.3099	0.3103
3-Party	0.3368	0.3370	0.3374
4-Party	0.3377	0.3382	0.3389
5-Party	0.3416	0.3420	0.3425

Table 5.20. Effects of masking optimum power values on MAE (Colorado data set)

	0.05	**0.15**	**0.25**
Integrated	0.2177	0.2179	0.2182
3-Party	0.2399	0.2401	0.2403
4-Party	0.2430	0.2434	0.2439
5-Party	0.2472	0.2475	0,2478

As expected, randomization decreases accuracy. If there is no randomness, the accuracy is 0.1297 for 5-party when *G* is 50. After adding random numbers to optimum power value, the accuracy becomes 0.1303. Therefore, there is 0.5% increase in accuracy. Similar observation can be observed for all cases. In addition to this, MAE values present the same trend as RMSE values for both data sets. To sum up, the suggested scheme helps data owners with smaller amount of measurements provide more accurate predictions on their combined data without violating their privacy. Even if there are accuracy losses due to our randomized scheme, collaboration compensates such losses.

5.5. Conclusion

Companies hesitate to share their private data with other parties for their financial future. Therefore, if there is no privacy preserving solutions for IDW method, they are not willing to collaborate with other parties. Due to insufficient measurements, as shown in the experiments, they come up with inaccurate predictions. However, if they use the proposed schemes, they provide more accurate predictions. As privacy analysis signifies that private data of both client and participating parties are kept confident. The recommended method encourages the companies to collaborate with other parties to derive reliable results.

6. PRIVACY-PRESERVING KRIGING ON DISTRIBUTED DATA

Data collected for kriging interpolation purposes might be distributed between two or more parties. Coordinates of some sample points and their related measurements can be distributed among different companies. Such parties might decide to collaborate for better kriging services. However, such collaboration should violate their confidentiality. In this chapter, kriging-based methods are presented, which are proposed to provide predictions from partitioned or distributed data with privacy.

6.1. Private Kriging on Partitioned Data

Unlike traditional kriging interpolation, measurements for specific locations in a given region are arbitrarily held by two servers, $S1$ and $S2$. These two servers can collaboratively provide kriging-based predictions to their clients. The client C can ask a prediction from one of them, which is referred to as the master server (*MS*). Suppose that the $S1$ acts as the *MS*. It is also assumed that the servers are semi-honest. In other words, they follow the protocol as they are expected; however, they try to derive as much information as possible about each other's private data from interim and final results. For the servers, measurements held by each other and their related coordinates are considered private. Similarly, estimated prediction and its corresponding location are regarded as confidential for the client.

To achieve privacy, various methods have been used in PPDM schemes. One of the most extensively used methods is called OT. Hence, OT is utilized in the proposed scheme. In addition to OT, HE is also widely used privacy-preserving method. HE allows addition or multiplication of encrypted values without decrypting them. If ξ is an encryption function and K is a public key, and x_{j1} and x_{j2} are private data values, then Paillier's HE scheme allows to compute $\xi_K(X) = \prod_{j=1}^{n}(\xi_K(x_{j1}))^{x_{j2}}$ values.

Randomization can be used for data masking. Agrawal and Srikant (2000) utilize randomization to achieve privacy. To disguise a private number x, a simple method is to add a random value r to it; so that $x + r$, rather than x, will appear in

the database, where r is a random value drawn from some distribution. Such distribution might be uniform or Gaussian with zero mean and a standard deviation (σ). Randomization is useful for estimating aggregate data from perturbed data. Aggregate data can be estimated with decent accuracy from masked data.

In the following, the recommended protocol is explained, which is referred to as the private kriging on partitioned data (PKPD). Although online performance is not that critical in kriging interpolation, the computations conducted in the protocol can still be grouped as off-line and online.

Off-line Phase: This phase includes calculating distances and semi-variances, performing binning, and creating model.

A. Distance and Semi-variance Estimation: $S1$ and $S2$ are supposed to calculate distances between any two locations i and j using Eq. (2.2) and the related semi-variances using Eq. (2.3). Since there are G sample points in a given area A, it is assumed that G_{S1} and G_{S2} locations are held by $S1$ and $S2$, respectively, where $G = G_{S1} + G_{S2}$. Thus, there are two cases as follows:

I. ***Case I**: Any two locations i and j are held by the same server***: For those locations held by the same server, each server can estimate the distances between them using Eq. (2.2) and the related semi-variances using Eq. (2.3) without the help of the other server because the required data are held by that server as follows:

1. $S1$ estimates distances between any two locations, i and j, where $i = 1, 2, \ldots, G_{S1}\text{-}1$ and $j = i + 1, i + 2, \ldots, G_{S1}$. It also computes the corresponding semi-variances using Eq. (2.3).
2. $S2$ finds distances between any two locations, i and j, where $i = 1, 2, \ldots, G_{S2}\text{-}1$ and $j = i + 1, i + 2, \ldots, G_{S2}$. It also finds the related semi-variances using Eq. (2.3).

II. ***Case II: Each of any two locations i and j is held by different server***: In this case, the servers need to exchange data to estimate distances and semi-variances collaboratively. They first compute distances using Eq. (2.2), where the equation can be written as follows:

$$d_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} =$$

$$\sqrt{x_i^2 + y_i^2 + x_j^2 + y_j^2 + (-2)x_i x_j + (-2)y_i y_j} \quad (6.1)$$

As seen from Eq. (6.1), *S*1 and *S*2 can compute $(x_i^2 + y_i^2)$ and $(x_j^2 + y_j^2)$ partial sums, respectively. However, they need to collaborate to find $(-2)x_i x_j$ and $(-2)y_i y_j$ values. They then compute semi-variances using Eq. (2.3), where the equation can be similarly written as follows:

$$s_{ij} = 0.5 \times \left[P_i - P_j\right]^2 = \frac{P_i^2}{2} + \frac{P_j^2}{2} + (-1)P_i P_j \quad (6.2)$$

The steps are first described in terms of *S*1. For $i = 1, 2, \ldots, G_{S1}$-1, the servers perform the followings:

1. *S*1 encrypts$-2x_i$, $-2y_i$, and $-P_i$ values using an HE scheme with its public key *KS*1 and obtains $\xi_{KS1}(-2x_i)$, $\xi_{KS2}(-2y_i)$, and $\xi_{KS1}(-P_i)$. It then sends them to *S*2.
2. *S*2 should find a way to prevent *S*1 from learning the coordinates of any location *j* because the distances between *j* and each location $i = 1, 2, \ldots, G_{S1} - 1$ are estimated and *S*1 can determine two unknowns (the coordinates of the location *j*, x_j and y_j values) from $G_{S1} - 1$ equations. Hence, *S*2 masks the coordinates using randomization. To do so, *S*2 creates two sets of random numbers using uniform distribution with zero mean and σ, where each set includes G_{S2} number of random numbers. One set, including r_j values, is used to hide x_j values and the other set, including v_j values, is utilized to perturb y_j values. Thus, *S*2 utilizes $x_j + r_j$ and $y_j + v_j$ to estimate the corresponding distances. Using an HE method with K_{S1}, *S*2 first finds partial aggregate $dp_{ij} = \xi_{KS1}\left[(x_j^2 + y_j^2) - 2x_i x_j - 2y_i y_j - 2x_i r_j - 2y_i v_j\right]$ for $j = i + 1, i + 2, \ldots, G_{S2}$.
3. *S*2 then computes the partial aggregate of the related semi-variance using an HE scheme and finds sp_{ij} for $j = i + 1, i + 2, \ldots, G_{S2}$, where $sp_{ij} = \xi_{KS1}\left[\frac{P_j^2}{2} + (-P_i)P_j\right]$. Similar argument defined in the previous step is also true for the related measurements. Therefore, *S*2 masks such partial aggregate

similarly. To mask them, *S*2 utilizes HE scheme with *KS*1 and randomization; and finds encrypted disguised partial sum as follows: $\xi_{KS1}(sp_{ij}) = \xi_{KS1}(sp_{ij} + u_{ij})$. Recall that u represents uniformly generated random numbers with zero mean. *S*2 then sends such encrypted masked values to *S*1.

4. Since *S*1 knows the related private key, it decrypts $\xi_{KS1}(dp_{ij})$ and $\xi_{KS1}(sp_{ij})$; and obtains $dp_{ij} = dp_{ij} + rv_{ij}$ and $sp_{ij} = sp_{ij} + u_{ij}$ partial aggregates.
5. *S*1 can now compute masked distances and semi-variances because it knows the other required partial values as follows: $\overline{d_{ij}} = \sqrt{(x_i^2 + y_i^2) + dp_{ij} + rv_{ij}} = d_{ij} + U_{ij} = d_{ij} + R_{ij}$ and and $\overline{s_{ij}} = \left[\frac{P_i^2}{2} + sp_{ij} + u\right]$. Now, *S*1 has all necessary distances and semi-variances in perturbed form and the ones estimated by it.

*S*1 and *S*2 now switch their roles and perform the above steps for $j = 1, 2, \ldots, G_{S2}-1$. Thus, at the end of such steps, *S*2 now gets all necessary distances and semi-variances in masked form and the ones estimated by it.

B. Binning and Model Generation: The servers follow the following steps to bin the distances and create the model:

1. They decide the binning methodology. One possible methodology is to use equal width bins.
2. Each server first clusters the distances and the related semi-variances found in ***Case I*** by each server. They then found partial aggregates and counts for each bin and exchange such partial aggregates. Recall that each server cannot learn coordinates and related measurements from such aggregates even if there is only one distance and related measurements fall into a single bin.
3. They then cluster the remaining distances; and find average distances and related average semi-variances for each bin. Since most of the distances and the related semi-variances are masked by random numbers, with increasing number of items fall into the same bin, the effects of random numbers become smaller. Note that the random numbers are generated using a random number distribution with zero mean, the expected value of the arithmetic mean of the random numbers will be zero.

4. Next, each server plots average semi-variances versus average distances; and finds the formula to estimate semi-variance at any given distance. Notice that the servers might not end up with the same formula due to differently masked data items. Thus, $S1$ and $S2$ might end up with $Semi\text{-}variance_{S1} = f_{S1}$ (*distance*) and $Semi\text{-}variance_{S2} = f_{S2}$ (*distance*), respectively.
5. Each server then can estimate the semi-variances between any two locations using the obtained function; and $S1$ and $S2$ creates $\boldsymbol{\Gamma_{S1}}$ and $\boldsymbol{\Gamma_{S2}}$ matrices, which are $(G + 1) \times (G + 1)$ symmetric matrices including the related semi-variances. Notice that the last row and correspondingly the last column are filled with 1s, except the diagonal entry, which is set to 0. Also note that the related semi-variances for those distances estimated in the ***Case I*** are exchanged. Since the parties do not know the semi-variance functions held by each other, they cannot learn the related distances from such exchanged data.
6. They finally generate the models, $\boldsymbol{\Gamma^{-1}}$ matrices. Hence, $S1$ and $S2$ finds $\boldsymbol{\Gamma_{S1}^{-1}}$ and $\boldsymbol{\Gamma_{S2}^{-1}}$ matrices, respectively.

Online Phase: This phase includes weight and prediction estimations. In this phase, the client C is supposed to send the required data (coordinates of the location for which it is looking for a prediction) to the master server *MS*. However, such data are also considered confidential. Therefore, the servers and the client perform the followings to estimate prediction without jeopardizing their privacy.

A. Estimating Weights: The parties and the client can estimate the weights collaboratively as follows:

1. In order to hide its true location q for which C is looking for a prediction, it creates n-1 bogus locations and masks q among them. It then sends all n locations to the *MS*.
2. The *MS* forwards all these n locations to the $S2$. Recall that $S1$ acts as *MS*.
3. For each location $j = 1, 2, \ldots, n$, the servers then perform the followings:
   a. Each server estimates the distances between j and each location they hold using Eq. (2.2).
   b. $S2$ encrypts each distance using an HE scheme with its public key $KS2$; and sends them to the *MS*.

c. Using an HE method, the *MS* finds the related semi-variances in the encrypted form for those distances received from *S*2 utilizing *Semi-variance*$_{S1}$ = f_{S1} (*distance*).

d. In the meantime, the *MS* also finds the corresponding semi-variances for those distances estimated by it. It then encrypts them using HE method with *KS*2.

e. The *MS* then creates the matrix **g**, which is a (G + 1) × 1 matrix including the semi variances in encrypted form estimated between q and each measured location.

f. Since the *MS* has the matrix $\mathbf{\Gamma_{S1}^{-1}}$ and the matrix **g** (including encrypted values), it can estimate the kriging weights (**λ** matrix) using Eq. (2.5) by employing an HE scheme. Notice that **λ** is a (G + 1) × 1 matrix including encrypted weights, which are encrypted with *KS*2.

B. Prediction Estimation: For each location j = 1, 2, …, n, the servers then perform the followings:

1. The *MS* sends the required encrypted weights to *S*2. Since *S*2 knows the corresponding private key, it decrypts such encrypted weights and obtains λ_{S2j} values.

2. It then finds partial aggregate of the prediction for location z (referred to as PP_{S2z}) by performing a scalar dot product, as defined in Eq. (2.6). Notice that PP_{S2z} can be estimated as follows:

$$PP_{S2z} = \boldsymbol{\lambda}_{S2j} \cdot \mathbf{P}_{S2j} = \sum_{j=1}^{G_{S2}} \boldsymbol{\lambda}_{S2j} \times \mathbf{P}_{S2j}.$$

3. Similarly, the *MS* also finds partial aggregate of the prediction for location z (referred to as PP_{S1z}) in encrypted form by performing a scalar dot product, as defined in Eq. (2.6) using an HE method with *KC*. Notice that PP_{S1z} can be estimated as follows:

$$\xi_{KC}(PP_{S1z}) = \xi_{KC}(\boldsymbol{\lambda}_{S1j} \cdot \mathbf{P}_{S1j}) = \xi_{KC}\left[\sum_{j=1}^{G_{S1}} \boldsymbol{\lambda}_{S1j} \times \mathbf{P}_{S1j}\right] \quad (6.3)$$

4. The *MS* then sends it to *S*2, which then compute $\xi_{KC}(P_j) = \xi_{KC}(PP_{S2z}) + \xi_{KC}(PP_{S1z})$ in encrypted form using the homomorphic encryption property.

5. *S*2 then sends it to the *MS*.

6. Once the *MS* obtains the predictions for all n locations, the *C* utilizes OT to get the prediction for its real location q. It finally decrypts it and obtains P_q.

6.2. Performance and Privacy Analysis

In this section, the protocol is investigated with respect to overall performance (supplementary workloads) and privacy. Additional off-line and on line costs are inevitable because privacy and performance conflict with each other. Hence, the recommended scheme is analyzed in terms of such costs. Since a privacy-preserving method is proposed, it is shown if it protects privacy or not.

6.2.1. Overall performance analysis

Additional costs like storage, communication, and computation costs caused by privacy-preserving measures are interested in. Therefore, the protocol is analyzed with respect to supplementary costs. Although it consists of off-line and online phases and off-line costs are not that critical for the overall performance, online requirements are not that rigid in kriging interpolations like in some real time applications. In the following, the suggested method is scrutinized in terms of off-line and online extra workloads.

6.2.2. Storage costs analysis

Privacy-preserving measures conducted by the PKPD protocol cause some supplementary storage costs. Dominant extra storage costs occur due to the random numbers used to mask distances and semi-variances and the additional model created by the servers. Additional storage costs because of random numbers are in the order of O (G^2). Similarly, they are in the order of O (G^2) for additional model.

6.2.3. Communication costs analysis

In a traditional central server-based kriging interpolation, number of communications is *two* only. The client sends the coordinates of the location to the server, which sends a prediction back to the client. In the proposed scheme, however, number of communications increases due to privacy concerns. In the off-

line phase, number of communications performed between the servers is $2(G_{S1} + G_{S2}) = 2G$. In other words, the number of communications conducted off-line is in the order of $O(G)$. During online phase, the client performs one OT, which could be achieved with poly-logarithmic (in n) communication time, stated before. Moreover, the servers and the client conduct six more communications. Although number of communications increases during both off-line and online computations, they are considered acceptable due to soft real time requirements of kriging interpolation.

6.2.4. Computation costs analysis

The suggested protocol is also investigated with respect to extra off-line and on line workloads. Overwhelming additional computation workloads occur due to cryptographic computations (encryption and decryption) and multiplications. The costs due to random number generation, additions, and subtractions are omitted. In the off-line phase, number of encryptions is in the order of $O(G_{S1}G_{S2})$. Similarly, number of decryptions is also in the order of $O(G_{S1}G_{S2})$. In addition to encryptions and decryption, extra computations occur because of the additional model. Notice that in a traditional kriging interpolation, only one model is created. However, the servers create two models in the proposed method. Thus, computation costs increase by two times due to model generation. Number of encryptions is in the order of $O(nG^2)$ in the online phase. Likewise, number of decryptions is in the order of $O(nG)$. Moreover, number of multiplications increases by $O(n)$ times in the protocol because predictions are estimated for n locations rather than one. And finally, one OT is conducted. However, its computation cost can be considered negligible compared to workloads caused by cryptographic computations.

6.2.5. Privacy analysis

The proposed protocol should be able to protect the participating parties' (the servers and the client) privacy. Confidential data items are measurements and their related locations held by the servers and the estimated prediction and the location

for which a prediction is sought. Thus, it is shown that the protocol protects private data while providing predictions on partitioned data. It is assumed that the servers and the client are semi-honest. They follow the protocol steps yet they try to learn as much information as possible about each other's private data.

The client *C* wants to hide the coordinates of the location for which she is looking for a prediction. To do so, she hides the true location's coordinates among the randomly generated *n* - 1 bogus locations and utilizes OT. Therefore, the probability for the servers to guess the true location is 1 out of *n*. With increasing value of *n*, such probability becomes smaller; however, performance might get worse. The parties can decide the value of *n* according to how much privacy and performance they want. The estimated prediction is also regarded as confidential. As described previously, such prediction is protected using encryption. The servers cannot decrypt the encrypted value, because they do not know the related private key, which is held by the client. Moreover, it is hidden among *n* - 1 predictions for bogus locations. Thus, the servers cannot learn the estimated prediction.

The measurements and their corresponding coordinates are private for the servers. They hide them from the client and each other. The client cannot derive information about them, because she receives a final aggregate from the *MS*. It is not possible to learn such data from an aggregate. In order to protect such confidential data from each other, the servers utilize random perturbation, encryption, and aggregation. Random data perturbation method helps the servers change the coordinate values in each distance calculation.

Uniformly randomly generated random numbers over a range are added to coordinates. They are independently created for each distance calculation. Due to randomization and masked coordinates, the servers cannot derive information about the real coordinates from distances. For example, adding a random number 0.10 to coordinates of a location moves the real point 18 kilometers away in any direction (the direction of the movement is arbitrary), where the data are from the U. S. National Geochemical Survey Database. Moreover, due to encryption, it is not possible to learn truthful information from encrypted data without knowing the decryption key. Finally, aggregate results also prevent the servers from deriving useful information about each other's data.

As discussed previously, HE and OT are secure and they protect confidential data. HE allows the parties to calculate the required kriging functions using encrypted data. Therefore, it prevents malicious parties from deriving information about each other's data. Similarly, OT prevents the parties from learning private data values.

6.2.6. Experiments: Accuracy analysis

Different sets of experiments are conducted using two real data sets collected for geo-statistical purposes to evaluate the proposed solution with respect to accuracy. (*i*) Since it is hypothesized that collaboration is expected to improve accuracy, experiments are run to show how cooperation between two parties affects the accuracy of the predictions. (*ii*) Recall that randomization, HE, and the OT are utilized to achieve privacy. Encryption and the OT do not affect accuracy. However, randomization might because it adds randomness to real data. Also note that coordinates and semi-variances are masked. Hence, another set of experiments are performed to show how disguising coordinates with randomization only affects the quality of the predictions, where varying ranges of random numbers are used. (*iii*) Third, trials are done to demonstrate accuracy changes due to semi-variance masking only, where again the ranges of random numbers are varied. (*iv*) Finally, a set of experiments are performed to show the joint effects of the randomization.

6.2.7. Experiments and empirical outcomes

6.2.7.1. Experiment I: Effects of collaboration

It is hypothesized that collaboration between two servers improves accuracy. In other words, predictions estimated from integrated data are more accurate than ones estimated from split data only. Therefore, a set of experiments are run to show how collaboration affects accuracy. In a given region, predictions can be estimated from *G* sample points and their related measurements. Recall that the partitioned data-based scenario assumes that such measurements are split between two servers.

Because data might be unevenly partitioned, it is assumed that β percent of the measurements are held by $S1$ while the remaining 100 - β percent are held by $S2$. When β is 50, it means that data are evenly partitioned between the servers $S1$ and $S2$. Similarly, β being 100 means that the entire measurements are held by a single party. In other words, the results for β being 100 represent the outcomes based on integrated data (the results after collaboration). If β is chosen as 20, it means that one server holds 20 percent of the data and the other holds the remaining 80 percent of the data.

Due to uneven partitioning, accuracy gains because of collaboration are expected to be different. Moreover, accuracy is dependent on the number of measurements G. Hence, experiments are conducted by varying the β values from 20 to 100 and the G values from 5 to 50. Given G measurements, it is assumed that $S1$ holds β percent of them. Thus, uniformly randomly selected β percent of the G measurements are used to estimate predictions. Then, the remaining ones are utilized. Due to randomness, the trials are repeated 100 times. The RMSE and MAE values are computed for both cases. In the case of collaboration, predictions are estimated from all G measurements. In other words, the servers provide predictions on their integrated data collaboratively. Hence, the entire measurements (β = 100) are used to estimate predictions. The RMSE and MAE values are computed for integrated data. Finally, the outcomes with respect to RMSE and MAE are displayed in Table 6.1 and Table 6.2, respectively, for both data sets.

Table 6.1. Effects of collaboration on RMSE with varying β and G values

	Illinois Data Set				***Colorado Data Set***			
β*(%)/*G	**5**	**15**	**30**	**50**	**5**	**15**	**30**	**50**
20	0.1416	0.1357	0.1345	0.1352	0.3479	0.3354	0.3368	0.3418
35	0.1365	0.1333	0.1335	0.1338	0.3346	0.3276	0.3279	0.3293
50	0.1304	0.1269	0.1273	0.1280	0.3237	0.3192	0.3194	0.3199
65	0.1284	0.1251	0.1261	0.1272	0.3224	0.3141	0.3144	0.3159
80	0.1247	0.1225	0.1228	0.1244	0.3198	0.3128	0.3132	0.3135
100	0.1212	0.1192	0.1190	0.1215	0.3165	0.3092	0.3098	0.3113

Table 6.2. Effects of collaboration on MAE with varying β and G values

	Illinois Data Set				*Colorado Data Set*			
β*(%)/*G	**5**	**15**	**30**	**50**	**5**	**15**	**30**	**50**
20	0.1063	0.1014	0.1012	0.1017	0.2528	0.2452	0.2471	0.2517
35	0.1005	0.0982	0.0986	0.0991	0.2398	0.2337	0.2344	0.2360
50	0.0964	0.0939	0.0946	0.0954	0.2328	0.2284	0.2287	0.2291
65	0.0953	0.0920	0.0931	0.0944	0.2312	0.2242	0.2247	0.2269
80	0.0924	0.0905	0.0909	0.0923	0.2290	0.2230	0.2232	0.2235
100	0.0905	0.0885	0.0882	0.0905	0.2260	0.2201	0.2205	0.2207

Notice that the goal is to verify the hypothesis, which states that accuracy improves due to collaboration. As observed from Table 6.1 and Table 6.2, collaboration between two servers makes accuracy better. When G is 30 in the Illinois data set, the accuracy improves from 0.1345 to 0.1190 for an $S1$ that holds 20 percent of the measurements. Similarly, the precision increases from 0.1228 to 0.1190 for an $S2$ that holds 80 percent of the measurements. Accuracy gains due to collaboration are higher for the server holding less data. Similar trends are observed in the Colorado data set. If G is 15, accuracy improves from 0.3354 to 0.3092 for $S1$ and from 0.3128 to 0.3092 for $S2$. When data are evenly partitioned between two servers for a G of 30 and 15, the RMSE values are approximately 0.1273 and 0.3192 for the Illinois and Colorado data sets, respectively. As observed from Table 6.2, the outcomes with respect to the MAE values also confirm the hypothesis. With decreasing β values, data holders benefit more from collaboration for both data sets. The best outcomes are observed when G is 30 and 15 for the Illinois and Colorado data sets, respectively, in terms of both the RMSE and MAE values. In summary, the servers holding partitioned data are able to provide more accurate predictions if they cooperate.

6.2.7.2. Experiment II: Effects of disguising coordinates only

To hide true coordinates, data owners add random numbers to them. Hence, in the second set of experiments, how adding random numbers to coordinate values only affects the accuracy of the model is investigated. Notice that because privacy and accuracy are conflicting goals, randomization might make accuracy worse. It

is assumed that the servers generate random numbers using a uniform distribution over the range [-α; α]. Experiments are conducted while varying α values from 0.05 to 0.25, where G is set to 30 and 15 for the Illinois and Colorado data sets, respectively, because they provide the most accurate predictions, as shown previously. After running the trials 100 times, the overall averages are displayed in Table 6.3 for both data sets.

Table 6.3. Effects of disguising coordinates only on accuracy with varying α values

	Illinois Data Set			***Colorado Data Set***		
α	**0.05**	**0.15**	**0.25**	**0.05**	**0.15**	**0.25**
RMSE	0.1205	0.1245	0.1279	0.3093	0.3198	0.3264
MAE	0.0896	0.0936	0.0970	0.2203	0.2288	0.2356

Recall that if the servers decide to collaborate, the RMSE and MAE values are 0.1190 and 0.0882, respectively, for the Illinois data set; they are 0.3092 and 0.2201, respectively, for the Colorado data set, as shown in Table 6.1 and Table 6.2. Because privacy and accuracy are conflicting goals, accuracy is expected to decrease when masking coordinates. As observed from Table 6.3, the quality of the predictions in terms of both RMSE and MAE decreases. With increasing α values, accuracy losses become larger due to increased randomness. However, the outcomes for smaller α values (less than 0.25) are very close to the results obtained after collaboration without privacy concerns. Although privacy-preserving measures make precision worse, the results of the scheme are still promising because the accuracy gains due to collaboration are larger than the accuracy losses due to privacy concerns.

6.2.7.3. Experiment III: Effects of disguising measurements only

In addition to protecting coordinates using randomization, the related measurements are also masked via randomized perturbation. Due to data perturbation, the accuracy is expected to decrease. To show how masking measurements affect precision, another set of trials are performed using real data sets. It is again assumed that the servers generate random numbers using a uniform

distribution over the range [- ρ; ρ]. Hence, ρ values are varied from 0.05 to 0.25 to show the effects of different amounts of randomness, where G is set to 30 and 15 for the Illinois and Colorado data sets, respectively. The overall averages of 100 trials are shown in Table 6.4 for both data sets.

Table 6.4. Effects of disguising measurements only on accuracy with varying ρ values

	Illinois Data Set			***Colorado Data Set***		
ρ	**0.05**	**0.15**	**0.25**	**0.05**	**0.15**	**0.25**
RMSE	0.1204	0.1299	0.1541	0.3092	0.3152	0.3313
MAE	0.0894	0.0986	0.1221	0.2208	0.2255	0.2401

As expected and observed from Table 6.4, the quality of the predictions decreases due to masking measurements because data perturbation has negative effects on accuracy. With increasing ρ values, the amount of randomness increases, thus, causing larger accuracy losses. Although loss of accuracy is inevitable, for smaller ρ values, it is still possible to obtain promising outcomes.

6.2.7.4. Experiment IV: Overall performance of the proposed scheme

A set of experiments is finally conducted to show the joint effects of the data disguising schemes. In these trials, the coordinates in distance estimations and the measurements in semi-variance computations are both masked, as explained previously. Random numbers are generated using a uniform distribution over the same range [- δ; δ]. The δ values are changed from 0.05 to0.25 and the trials are repeated 100 times. The optimum values of G are used for both data sets. The overall averages are demonstrated in Table 6.5. Notice again that the RMSE and MAE values for the kriging model on integrated data without privacy concerns are 0.1190 and 0.0882 for the Illinois data set, respectively. Similarly, they are 0.3092 and 0.2201, respectively, for the Colorado data set.

Table 6.5. Overall performance of the protocol with varying δ values

	Illinois Data Set			***Colorado Data Set***		
δ	**0.05**	**0.15**	**0.25**	**0.05**	**0.15**	**0.25**
RMSE	0.1198	0.1420	0.629	0.3092	0.3228	0.3376
MAE	0.0899	0.1098	0.1262	0.2205	0.2339	0.2487

The outcomes presented in Table 6.5 demonstrate that privacy-preserving measures decrease precision due to randomness. The quality of the predictions decreases with augmented δ values due to increasing randomness. However, for smaller δ values, it is still possible to obtain promising results. For example, the outcomes for a δ of 0.05 in the scheme are better than the outcomes for split data only. Hence, the parties are able to provide better outcomes in terms of accuracy and more dependable predictions (outcomes are generated from a greater number of measurements) using the proposed method if they use smaller δ values.

Finally, the empirical outcomes are presented in Table 6.6 in order to depict the overall picture, where G is set to 30 and 15 for the Illinois and Colorado data sets, respectively and δ is selected as 0.05. The outcomes demonstrate the outcomes of kriging on integrated data without privacy (KID), kriging on partitioned data without privacy (KPD) and the proposed privacy-preserving method or PKPD. Without collaboration, the servers may not provide accurate and dependable predictions from insufficient data. On the one hand, when collaborating without privacy concerns, accuracy improves with respect to RMSE and MAE for both data sets. Collaboration definitely improves overall performance. To enhance both precision and trustworthiness, data owners can collaboratively generate a kriging model and offer predictions on integrated data. The privacy-preserving measures, on the other hand, make accuracy worse because privacy and precision are conflicting goals. With increasing randomness, privacy improves while accuracy degrades. However, for a small amount of randomness, as shown with empirical results in Table 6.6, the method is still able to provide more accurate and dependable predictions than the ones estimated from split data only. Accuracy losses due to randomness are small, as observed from Table 6.6.

Table 6.6. Comparison of the PKPD with kriging without privacy

	Illinois Data Set			*Colorado Data Set*		
	KID	**KPD**	**PKPD**	**KID**	**KPD**	**PKPD**
RMSE	0.1190	0.1273	0.1198	0.3092	0.3192	0.3092
MAE	0.0882	0.0946	0.0899	0.2201	0.2284	0.2205

6.3. Private Kriging on Distributed Data

In a traditional kriging interpolation, there is one server that holds coordinate and measurement information. The client asks a prediction for a specific location and the server does all the required calculations and returns an estimated value as a prediction. In such scheme, there is no privacy. The client does not will to disclose the coordinate, where it is interested in. In addition, in some region, there might be more than two servers that collect data for kriging purposes. As mentioned before, accuracy of geo-statistical methods depends on the number of sample points. Therefore, the servers may want to collaborate to provide better services. This scheme is called distributed data-based method (PKDD), where there are M servers $S_1, S_2, \ldots, S_M$ holds private data. The servers do not want to share their private data in order to survive in the future. The proposed solution gives servers an opportunity to come together and create more accurate geo-statistical models.

In the following section, the steps of the protocol are described. The computations can be grouped as off-line and online computations, where online performance is not that critical.

Off-line Phase: Calculations like distance, semi-variance, binning, and creating kriging model are performed in this phase.

A. Distance and semi-variance calculation: In order to create a kriging model, all distance and semi-variance values for locations, the where servers have measurements have to be calculated. In the proposed protocol, there is a total number of G ($\sum_{i=1}^{n} Gsi$) sample points in region A. The servers $S_1, S_2, \ldots, S_M$ hold $G_{S1}, G_{S2}, \ldots, G_{SM}$ sample points, respectively. There are two cases; points i and j are held by the same server and points i and j are held by two different servers.

I. Case I: Any two points i and j are held by same server: The servers do

not need to collaborate with others to calculate distance and semi-variance values for points, which are stored in its database. Therefore, each server calculates distance and semi-variance values using Eq. (2.2) and Eq. (2.3).

1. Each server calculates distances between points i and j, where $i = 1, 2, \ldots, G_{Sk} - 1$, $j = i + 1, i + 2, \ldots, G_{Sk}$ and $k = 1, 2, \ldots, M$.
2. Each server calculates semi-variance values for points i and j, where $i = 1, 2, \ldots, G_{Sk} - 1$, $j = i + 1, i + 2, \ldots, G_{Sk}$ and $k = 1, 2, \ldots, M$.

II: Case II: Each of any two locations i and j is held by different server: The servers need to collaborate with the other servers to find distance and semi-variance values for points, which are held by different servers. The distance equation, Eq. (2.2), can be expanded as follows:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

$$= \sqrt{x_i^2 + y_i^2 + x_j^2 + y_j^2 + (-2)x_i x_j + (-2)y_i y_j}$$

The server, which possesses point (x_i, y_i) calculates the values x_i^2 and y_i^2 by itself. The corresponding server, which holds point (x_j, y_j) calculates x_j^2 and y_j^2. However, they need to collaborate to calculate values $(-2)x_i x_j$ and $(-2)y_i y_j$. Similarly, they can calculate semi-variance value. The Eq. (2.3) can be expanded as follows:

$$s_{ij} = \frac{1}{2} \times [P_i - P_j]^2 = \frac{P_i^2}{2} + \frac{P_j^2}{2} + (-1)P_i P_j$$

The server, which possesses point (x_i, y_i) calculates value $\frac{P_i^2}{2}$ by itself. The corresponding server, which holds point (x_j, y_j) calculates $\frac{P_j^2}{2}$. However, they need to collaborate to calculate the values $(-1)P_i P_j$.

The steps of how to calculate distance and semi-variance values are explained in terms of S_k. The corresponding server will be called as S_m ($m = 1, 2, \ldots, k\text{-}1, k+1, \ldots, n$).

1. S_k calculates the values $\xi_{KSk}(-2x_i)$, $\xi_{KSk}(-2y_i)$, and $\xi_{KSk}(-P_i)$ using an HE method and its public key KSk. S_k sends all encrypted values to S_m.
2. S_m masks its coordinates using randomization technique in order to prevent S_k from learning S_k's private data. To do so, S_m creates two arrays (r and v) of

random number, which has uniform distribution with zero mean and σ. In each array, there are G_{Sm} numbers of random numbers. Then, S_m adds random numbers to its coordinates values to find $x_j + r_j$ and $y_j + v_j$. After calculating perturbed coordinates, S_m finds $\xi_{KSk}(\overline{dp_{ij}}) = \xi_{KSk}[(x_j^2 + y_j^2) + (-2x_i)x_j + (-2y_i)y_j + u_{ij}] = \xi_{KSk}(dp_{ij} + u_{ij})$ for $j = i + 1, i + 2, \ldots, G_{Sm}$.

3. Similarly, S_m produces a random array (u) using uniform distribution with zero mean and σ. To hide its measurement, S_m adds these random numbers. S_m finds $\xi_{KSk}(\overline{sp_{ij}}) = \xi_{KSk}\left[\frac{P_j^2}{2} + (-P_i)P_j + z_{ij}\right] = \xi_{KSk}(sp_{ij} + z_{ij})$ and sends to S_k.
4. S_k decrypt $\xi_{KSk}(dp_{ij} + u_{ij})$ and $\xi_{KSk}(sp_{ij} + z_{ij})$ values using its private key and gets $dp_{ij} + u_{ij}$ and $sp_{ij} + z_{ij}$.
5. S_k has all necessary values to compute distances and semi-variance value for points, which are held by S_m.
6. S_k repeats these process described above with other servers.
7. The servers change their roles and all servers get distance and semi-variance values for all points in the region A to create a kriging model.

B. Binning and Model Creation: After obtaining all required distance and semi-variance values to create a kriging model, the servers execute the following steps:

1. They agree on the binning methodology.
2. Next, they calculate average distance and semi-variance values for each bin.
3. Each server then plots average distance and semi-variance using one of the functions used in kriging models. They come up with a formula to describe the relationship between distance and semi-variance. Due to random data, which are used to hide coordinate and measurement values, each server might end up with a slightly different formula. *Semi-variance*$_{S1}$ = f_{S1} (*distance*), *Semi-variance*$_{S2}$ = f_{S2} (*distance*), …, *Semi-variance*$_{Sn}$ = f_{Sn} (*distance*), respectively.
4. Each server calculates **Γ** matrices using the semi-variance formula and distance values for all points ($G = \sum_{i=1}^{n} Gsi$). **Γ** is a (G +1) × (G +1) symmetric matrix. The last row and column are filled with 1's except the

diagonal element. The value of diagonal element is 0.

5. They finally find $\mathbf{\Gamma^{-1}}$ matrices to use in estimation phase, as explained below.

Online Phase: In this phase, the client *C* sends the coordinate for which it needs a prediction to the *MS* (It is assumed that S_1 acts as *MS*, but any server may act as *MS*). The coordinate value is considered as private data of the client. The servers, which have measurement values in the same region calculate kriging weights and produce the final estimation collaboratively.

A. Estimating Weights: The servers and the client *C* calculates weights, as described below:

1. The *C* utilizes OT to hides its true location *q*. Therefore, it produces *n*-1 bogus coordinate values and stores true location in *n*-1 bogus coordinate. Then, it permutes *n* location and sends them to the *MS*.
2. The *MS* forwards all *n* coordinate values to other servers.
3. For each location $z = 1, 2, \ldots, n$, the servers perform the followings:
   a. Each server finds distances between *z* and each coordinate in their database using Eq. (2.2).
   b. The servers $S_1, S_2, \ldots, S_n$ encrypt each distance using an HE scheme with their public key *KSi* and send them to other servers.
   c. The *MS* finds the related semi-variance values for distances received from servers $S_2, S_3, \ldots, S_n$ utilizing $Semi\text{-}variance_{S1} = f_{S1}(distance)$ in the encrypted form using an HE scheme.
   d. The *MS* then creates the matrix $\mathbf{g}$, which is a $(G + 1) \times 1$ matrix including the semi variances in encrypted form estimated between *z* and each measured location.
   e. Since the *MS* has the matrix $\mathbf{\Gamma_{S1}^{-1}}$ and the matrix $\mathbf{g}$ (including encrypted values), it can estimate the kriging weights ($\boldsymbol{\lambda}$ matrix) using Eq. (2.5) by employing an HE scheme. Notice that $\boldsymbol{\lambda}$ is a $(G + 1) \times 1$ matrix including encrypted weights, which are encrypted with public keys of server *KS*2, *KS*3, …, *KSn*.

B. Prediction Estimation: For each location $z = 1, 2, \ldots, n$, the servers perform the followings:

1. The *MS* sends corresponding weights, which are encrypted using public keys to servers S_2, S_3, …, S_n. The servers decrypt encrypted values using their private key and gets λ_{Smj}.
2. They perform scalar dot products, defined in Eq. (2.6), to calculate partial aggregate of the prediction for location *z* (referred to as PP_{Smz}). PP_{Smz} can be calculated as follows:

$$PP_{Smz} = \boldsymbol{\lambda}_{Sm} \cdot \mathbf{P}_{Sm} = \sum_{j=1}^{G_{Sm}} \boldsymbol{\lambda}_{Smj} \times \mathbf{P}_{Smj}$$

3. Then, they compute $\xi_{KC}(P_{Smz})$ using client's public key *KC* and send to the *MS*.
4. The *MS* similarly calculates partial aggregate of the prediction for location *z* (referred to as PP_{S1z}) in encrypted form by performing a scalar dot product using an HE scheme with client's public key *KC*. PP_{S1z} can be calculated as follows:

$$\xi_{KC}(PP_{S1z}) = \xi_{KC}(\boldsymbol{\lambda}_{S1} \cdot \mathbf{P}_{S1}) = \xi_{KC}\left[\sum_{j=1}^{G_{S1}} \boldsymbol{\lambda}_{S1j} \times \mathbf{P}_{S1j}\right]$$

5. The *MS* computes $\xi_{KC}(PP_z) = \xi_{KC}(PP_{S1z}) + \xi_{KC}(PP_{S2z}) + \ldots + \xi_{KC}(PP_{Snz})$ when it gets all prediction values from all servers.
6. The *C* gets the final prediction value for its real location *q* using OT. It decrypts using its private key and obtains prediction *Pq*.

6.4. Performance and Privacy Analysis

In this section, PKDD scheme is analyzed with respect to performance and privacy. PKDD scheme brings extra storage, communication, and computation cost to assure privacy between collaborating parties. These costs can be divided into two parts: off-line and online. In real life scenarios, off-line costs are less crucial than online costs. Moreover, these online constraints in kriging are not too rigid as compared to other online protocols such as recommender systems. In the last subsection, PKDD is analyzed in terms of privacy.

6.4.1. Storage cost analysis

PKDD scheme uses randomization, OT, and encryption to provide privacy. Randomization method increases storage cost. Parties use random numbers to disguise coordinate and measurement values. Each party creates three arrays storing random numbers for x_i, y_i, and P_i. Each array has a length of G_{Si}, where $i = 1, 2, \ldots, m$. In total, randomization requires $3 \times \sum_{i=1}^{m} Gsi = 3 \times G$ extra storage area. In other words, storage costs for random numbers are in the order of $O(G)$. In addition to randomization, parties need to store kriging model parameters. For $\boldsymbol{\Gamma}$ matrix, they need $m \times (G + 1) \times (G + 1)$ and for $\boldsymbol{\lambda}$ vector, they need $m \times (G + 1) \times 1$ storage area. In total, extra storage requirement is $3 \times G + \mathrm{m} \times (G + 1) \times (G + 1) + m \times (G + 1) \times 1$, which is in order of $O(mG^2)$, where m is much smaller than G.

6.4.2. Communication cost analysis

In real life applications, kriging requires *two* communications between client and server. The client sends the coordinate, where it needs a prediction and the server does all the required calculations and sends back a prediction value to the client. If privacy is necessary between collaborating parties, the servers are supposed to establish extra communications. During off-line phase, they have to establish $2 \times (m-1) \times \sum_{i=1}^{n} Gsi = 2 \times (m-1) \times G$, which is in order of $O(mG)$ communications between each other. In online phase, the client performs OT, which can be conducted in poly-logarithmic time (in n). Therefore, the servers have to make $\times\, n + 2n \times (m-1) \times G + n \times G$, which is in order of $O(mnG)$. In total, number of communications in PKDD are in the order of $O(mnG)$. G is always much bigger than both m and n. Therefore, it requires $O(G)$ extra communications.

6.4.3. Computation cost analysis

The proposed scheme has two phases: off-line and online. Accordingly, it is scrutinized with respect to two phases. The costs of some operations such as random number generation, subtractions, and additions are negligible. In the off-line phase,

order of O ($G_{S1}G_{S2}$, …,G_{Sn}) encryption should be conducted. In the similar manner, the decryption is in the order of O ($G_{S1}G_{S2}$, …, G_{Sn}). Moreover, additional computations occur due to model creation for each server. If privacy is not a concern, a model creation is adequate to provide a prediction; however, the proposed method requires n different model creations for n parties. Thus, computation costs increase by n times.

The online phase demands order of O (nG^2) encryptions. Likewise, number of decryptions is in the order of O (nG^2). Additionally, number of multiplications increases by O (n) times because predictions are estimated for n bogus locations rather than just one. In the final step of the protocol, an OT is conducted. However, computation cost of OT can be omitted.

6.4.4. Privacy analysis

The proposed scheme should verify that the private data of both client and the servers should be kept secret. Locations and corresponding measurements held by the servers and the coordinate for which the client asks prediction and result of interpolation method are assumed as confidential data. It is also assumed that the servers and the client are semi-honest. They follow the protocol steps; however, they try to learn as much information as possible about each other's private data.

The client C composes n-1 bogus coordinates and hides its true location among them and uses OT. The probability of guessing its true coordinate is 1 out of n. If n is chosen a bigger value, it increases privacy, but causes more computation cost. A reliable n value can be chosen to provide adequate privacy. The final result is also private data of the client. Therefore, the servers should not have an idea of such value. Since encryption enforces private key must be known by the client only, the servers cannot decrypt the cipher text, which is encrypted by client's public key. Hence, the servers cannot learn the final prediction value. The servers should hide their coordinates and measurements from others servers and the client. The client gets an aggregate value, which does not give information about the servers' private data. The servers use random perturbation and encryption methods to hide confidential data. Random perturbation methods change the coordinate value for

distance calculation. Random values generated from a uniform distribution over a reliable range are added to coordinate and measurements. The servers cannot find the exact coordinate and measurements of other servers. Finally, aggregate results also prevent the servers from deriving useful information about each other's data.

6.4.5. Experiments and accuracy analysis

6.4.5.1. Experiment I: Effects of collaboration

It is hypothesized that collaboration between parties provide more accurate and dependable prediction. In order to prove this hypothesis, different sets of experiments are conducted using real data. It is also assumed that there are one, three, four, or five parties. In one party case, all measurements are held by one party. In three-party case, there are three parties that hold one third of all measurements. The distribution is held randomly. The same methodology is followed for the four and five-party case. For all cases, *G* is varied from 5 to 50. The results of RMSE and MAE values are presented for the Illinois data set in Table 6.7 and 6.8. Table 6.9 and 6.10 show the RMSE and the MAE values for the Colorado data set.

Table 6.7. Effects of collaboration on RMSE with varying *G* values (Illinois data set)

G	**5**	**10**	**15**	**20**	**30**	**40**	**50**
Integrated	0.1207	0.1181	0.1185	0.1181	0.1183	0.1199	0.1208
3-Party	0.1353	0.1330	0.1321	0.1321	0.1323	0.1323	0.1326
4-Party	0.1380	0.1340	0.1333	0.1334	0.1329	0.1331	0.1336
5-Party	0.1404	0.1357	0.1345	0.1337	0.1333	0.1337	0.1340

Table 6.8. Effects of collaboration on MAE with varying *G* values (Illinois data set)

G	**5**	**10**	**15**	**20**	**30**	**40**	**50**
Integrated	0.0905	0.0889	0.0885	0.0877	0.0882	0.0898	0.0905
3-Party	0.1005	0.0984	0.0982	0.0984	0.0986	0.0988	0.0991
4-Party	0.1036	0.1005	0.1006	0.1008	0.1007	0.1008	0.1012
5-Party	0.1063	0.1022	0.1014	0.1013	0.1012	0.1014	0.1017

Table 6.9. Effects of collaboration on RMSE with varying *G* values (Colorado data set)

G	**5**	**10**	**15**	**20**	**30**	**40**	**50**
Integrated	0.3165	0.3096	0.3092	0.3096	0.3098	0.3103	0.3113
3-Party	0.3382	0.3317	0.3306	0.3306	0.3307	0.3317	0.3317
4-Party	0.3476	0.3368	0.3344	0.3350	0.3368	0.3381	0.3385
5-Party	0.3499	0.3378	0.3362	0.3363	0.3394	0.3427	0.3456

Table 6.10. Effects of collaboration on MAE with varying *G* values (Colorado data set)

G	**5**	**10**	**15**	**20**	**30**	**40**	**50**
Integrated	0.2260	0.2205	0.2201	0.2201	0.2205	0.2206	0.2207
3-Party	0.2419	0.2365	0.2359	0.2360	0.2362	0.2369	0.2369
4-Party	0.2505	0.2427	0.2426	0.2434	0.2449	0.2458	0.2464
5-Party	0.2528	0.2464	0.2462	0.2465	0.2488	0.2517	0.2544

The results shown in the tables verify the hypothesis. Collaboration definitely provides more accurate predictions. In each column, accuracy is getting worse for varying *G* values. Therefore, if geo-statistics methods are based on more measurements, they produce better results with respect to accuracy. The RMSE value improves 13% and the MAE value increases 15% for the Illinois data set when *G* is 20.

6.4.5.2. Experiment II: Overall performance of the proposed scheme

The second sets of experiments analyze overall effects of randomization method on both coordinates and measurement values. PKDD scheme utilizes randomization method to disguise coordinate values. However, randomization may affect accuracy due to adding random numbers. On the contrary, HE and OT do not change accuracy.

Table 6.11. Overall performance with varying δ values for the Illinois data set (RMSE)

	0.05	**0.15**	**0.25**
Integrated	0.1200	0.1424	0.1633
3-Party	0.1313	0.1527	0.1828
4-Party	0.1328	0.1529	0.1859
5-Party	0.1334	0.1568	0.1899

Table 6.12. Overall performance with varying δ values for the Illinois data set (MAE)

	0.05	**0.15**	**0.25**
Integrated	0.0901	0.1102	0.1266
3-Party	0.0986	0.1174	0.1432
4-Party	0.1018	0.1194	0.1470
5-Party	0.1022	0.1224	0.1503

Table 6.13. Overall performance with varying δ values for the Colorado data set (RMSE)

	0.05	**0.15**	**0.25**
Integrated	0.3092	0.3228	0.3376
3-Party	0.3369	0.3502	0.3615
4-Party	0.3380	0.3527	0.3639
5-Party	0.3394	0.3531	0.3647

Table 6.14. Overall performance with varying δ values for the Colorado data set (MAE)

	0.05	**0.15**	**0.25**
Integrated	0.2205	0.2339	0.2487
3-Party	0.2414	0.2543	0.2656
4-Party	0.2470	0.2567	0.2685
5-Party	0.2487	0.2583	0.2699

As expected, randomization makes accuracy worse. Adding a random number over the range [-0.05; 0.05] does not change the RMSE and the MAE values for the Colorado data set when *G* is 15. If the range is chosen over the range [-0.25; 0.25], accuracy decreases 9% for the same *G* value. However, the RMSE value is less than

5-party and the MAE value is equal to 5-party case. Therefore, if the range is chosen less than 0.25, PKDD method will produce better result than 5-party case. In conclusion, adding random numbers, if they are chosen from an acceptable range, does not significantly affect accuracy.

6.5. Conclusion

To sum up, insufficient measurements lead to inaccurate model creation. If the kriging model generated using less data, as understood from the results of the experiments, accuracy of the model is not reasonable. Therefore, companies should be encouraged to collaborate to join their data. However, their financial futures depend on such valuable data. The proposed solutions fill this gap. They enable companies to create a better kriging model based on measurements of two or more parties without revealing their private data. The privacy of the client is also taken into consideration. The coordinate and final prediction values are accepted as private data of the client. The servers cannot learn the coordinate for which the client needs prediction.

7. CONCLUSIONS AND FUTURE WORK

Without privacy concerns, it is relatively simple task to offer predictions using either kriging or inverse distance weighting interpolation techniques. In traditional inverse distance weighting and kriging methods, there is no privacy concern. However, coordinate values, related measurements, estimated predictions, and their locations are usually considered confidential. Revealing such data might cause privacy violations and financial losses. Therefore, involving parties do not want to disclose their private and valuable data to each other. It then becomes imperative to provide predictions using interpolation methods while preserving confidentiality.

In this dissertation, privacy-preserving schemes are applied to kriging and inverse distance weighting interpolation techniques in order to estimate recommendations without deeply jeopardizing the involving parties' privacy. Homomorphic encryption, 1-out-of-*n* oblivious transfer protocol, and randomization are used to achieve confidentiality. The proposed methods protect the clients' and the servers' privacy. Locations and their related measurements held by the servers and estimated prediction and its coordinate held by the client are considered as confidential. Each scheme is analyzed in terms of privacy, supplementary costs, and accuracy. Experiments on real data are conducted to show whether the proposed methods provide accurate predictions or not.

The proposed schemes should be able to provide accurate predictions efficiently while preserving confidentiality. In other words, there are basically three main goals that should be provided by the proposed methods. They are known as privacy, accuracy, and efficiency. Efficiency or performance requirements in interpolation methods are not rigid. Such methods usually do not offer recommendations with rigid limitations. Hence, unlike online recommendation schemes like collaborative filtering, efficiency or online performance is not that critical for interpolation techniques. The proposed methods should not cause significant accuracy losses due to privacy-preserving measures. Notice that privacy concerns usually make accuracy worse. The proposed methods are supposed to protect privacy without causing significant accuracy losses. Thus, the proposed

methods should estimate recommendations with decent accuracy while preserving confidentiality. Finally, the methods should not cause privacy violations. They are expected to protect privacy of each participating companies.

In geo-statistical applications, partitioned data might occur due to limited resources like budget, time, and so on. Data collectors might not gather enough measurements when they have not sufficient budget and time. In case of split data, it is challenging for data owners to provide geo-statistics methods because they do not want to share their confidential data. The proposed methods overcome this obstacle by allowing involving parties to estimate predictions without jeopardizing privacy. The recommended solutions are explained for providing partitioned and distributed data-based schemes with privacy. It is shown that the methods are secure and supplementary costs due to privacy measures do not significantly affect overall performance. Empirical outcomes show that the methods can produce promising predictions.

The proposed techniques are analyzed with respect to additional costs like storage, communication, and computation costs in order to show their efficiency. As shown by such analyses, the methods are able to efficiently provide predictions. In addition to performance analysis, they are also investigate in terms of privacy. Privacy analyses demonstrate that the schemes are secure; and they do not violate privacy. Finally, real data-based empirical outcomes show that the proposed methods provide accurate recommendations. Even if some privacy-preserving measures cause accuracy losses, the methods are still able to offer predictions with decent accuracy.

Kriging method generally produces more accurate predictions than inverse distance weighting. To protect confidential data of both client and servers, uniformly randomly chosen random numbers are added to power values in inverse distance weighting solutions. Similarly, noise data are added to coordinate values in order to hide servers' private data. As observed from the experiments, kriging results are worse for the same range of random numbers. However, if random numbers are chosen from a smaller range for kriging method, accuracy will be enhanced. A reasonable distribution range could be used without revealing confidential data.

There are several geo-statistics methods described in the literature. Future direction is to explore each geo-statistics methods as done for IDW and kriging. Therefore, there remains work to propose solutions for central-, partitioned- and distributed data-based schemes for each one of the geo-statistics methods. The proposed solutions for kriging and inverse distance weighting in this dissertation and future studies that will be presented in the future allow organizations to compute geo-statistics methods in a private way so that their outputs will be more accurate and dependable.

REFERENCES

Aggarwal, C.C. and Yu, P.S. (2008), "Privacy-preserving Data Mining: A Survey," *Handbook of Database Security* (Ed: Gertz, M. and Jajodia, S.), Springer US, New York, NY, USA, 431-460.

Agrawal, R. and Srikant, R. (2000), "Privacy-preserving data mining," *ACM SIGMOD Record*, **29 (2)**, 439-450.

Aguilar-Melchor, C., Fau, S., Fontaine, C. and Gogniat, G. (2013), "Recent advances in homomorphic encryption: A possible future for signal processing in the encrypted domain," *Signal Processing Magazine*, **30 (2)**, 108-117.

Ali, M., Goovaerts, P., Nazia, N., Haq, M.Z., Yunus, M. and Emch, M. (2006), "Application of Poisson kriging to the mapping of cholera and dysentery incidence in an endemic area of Bangladesh," *International Journal of Heatlh Geographics*, **5 (1)**.

Armstrong, M. (1998), *Basic Linear Geostatistics*, Springer, Berlin, Germany.

Asharov, G., Lindell, Y., Schneider, T. and Zohner, M. (2013), "More efficient oblivious transfer and extensions for faster secure computation," *The 2013 ACM SIGSAC Conference on Computer & Communications Security*, Berlin, Germany, 535-548.

Baboulin, M., Dongarra, J., Herrmann, J. and Tomov, S. (2013), "Accelerating linear system solutions using randomization techniques," *ACM Transactions on Mathematical Software*, **39 (2)**, 1-13.

Bansal, A., Chen, T. and Zhong, S. (2011), "Privacy preserving Back-propagation neural network learning over arbitrarily partitioned data," *Neural Computing and Applications*, **20 (1)**, 143-150.

Bartier, P.M. and Keller, C.P. (1996), "Multivariate interpolation to incorporate thematic surface data using inverse distance weighting (IDW)," *Computers & Geosciences*, **22 (7)**, 795-799.

Benaloh, J. (1994), "Dense probabilistic encryption," *The Workshop on Selected Areas of Cryptography*, Kingston, Canada, 120-128.

Bertino, E., Lin, D. and Jiang, W. (2008), "A Survey of Quantification of Privacy Preserving Data Mining Algorithms," *Privacy-Preserving Data Mining* (Ed: Aggarwal, C.C. and Yu, P.S.), Springer US, New York, NY, USA, 183-205.

Brassard, G., Crepeau, C. and Robert, J.-M. (1987), "All-or-nothing disclosure of secrets," *Lecture Notes in Computer Science*, **263**, 234-238.

Bringer, J., Chabanne, H. and Patey, A. (2013), "Privacy-preserving biometric identification using secure multiparty computation: An overview and recent trends," *Signal Processing Magazine*, **30 (2)**, 42-52.

Cachin, C., Micali, S. and Stadler, M. (1999), "Computationally private information retrieval with polylogarithmic communication," *Lecture Notes in Computer Science*, **1592**, 402-414.

Canny, J. (2002a), "Collaborative filtering with privacy," *The 2002 IEEE Symposium on Security and Privacy*, Oakland, CA, USA, 45-57.

Canny, J. (2002b), "Collaborative filtering with privacy via factor analysis," *The 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Tampere, Finland, 238-245.

Chen, K. and Liu, L. (2011), "Geometric data perturbation for privacy preserving outsourced data mining," *Knowledge and Information Systems*, **29 (3)**, 657-695.

Cheon, J.H., Coron, J.-S., Kim, J., Lee, M.S., Lepoint, T., Tibouchi, M. and Yun, A. (2013), "Batch fully homomorphic encryption over the integers," *Lecture Notes in Computer Science*, **7881**, 315-335.

Choi, M.-J., Kim, H.-S. and Moon, Y.-S. (2012), "Publishing sensitive time-series data under preservation of privacy and distance orders," *International Journal of Innovative Computing, Information and Control*, **8 (5)**, 3619-3638.

Clifton, C., Kantarcioglu, M., Vaidya, J., Lin, X. and Zhu, M.Y. (2002), "Tools for privacy preserving distributed data mining," *ACM SIGKDD Explorations Newsletter*, **4 (2)**, 28-34.

Cranor, L.F., Reagle, J. and Ackerman, M.S. (2000), *Beyond concern: Understanding net users' attitudes about online privacy*, AT&T Labs-Research Technical Report TR 99.4.3, Cambridge, MA, USA.

Duan, Y. and Canny, J. (2008), "Practical private computation and zero-knowledge tools for privacy-preserving distributed data mining," *The SIAM International Conference on Data Mining*, Atlanta, GA, USA, 265-276.

Even, S., Goldreich, O. and Lempel, A. (1985), "A randomized protocol for signing contracts," *Communications of the ACM*, **28 (6)**, 637-647.

Evfimievski, A. (2002), "Randomization in privacy preserving data mining," *ACM SIGKDD Explorations Newsletter*, **4 (2)**, 43-48.

Exeter, D.J., Rodgers, S. and Sabel, C.E. (2014), ""Whose data is it anyway?" The implications of putting small area-level health and social data online," *Health Policy*, **114 (1)**, 88-96.

Fritz, J., Neuweiler, I. and Nowak, W. (2009), "Application of FFT-based algorithms for large-scale universal kriging problems," *Mathematical Geosciences*, **41 (5)**, 509-533.

Gambs, S., Killijian, M.-O., Moise, I. and del Prado Cortez, M.N. (2013), "MapReducing GEPETO or towards conducting a privacy analysis on millions of mobility traces," *The 2013 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum*, Cambridge, MA, USA, 1937-1946.

Gordon, S.D., Malkin, T., Rosulek, M. and Wee, H. (2013), "Multi-party computation of polynomials and branching programs without simultaneous interaction," *Lecture Notes in Computer Science*, **7881**, 575-591.

Guo, C., Chang, C.-C. and Wang, Z.-H. (2012), "A new data hiding scheme based on DNA sequence," *International Journal of Innovative Computing Information and Control*, **8 (1)**, 139-149.

Han, S. and Ng, W.K. (2007a), "Privacy-preserving genetic algorithms for rule discovery," *Lecture Notes in Computer Science*, **4654**, 407-417.

Han, S. and Ng, W.K. (2007b), "Multi-party privacy-preserving decision trees for arbitrarily partitioned data," *International Journal of Intelligent Control and Systems*, **12 (4)**, 351-358.

Henecka, W. and Schneider, T. (2013), "Faster secure two-party computation with less memory," *The 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, Hangzhou, China, 437-446.

Huang, S.C.-H., Lin, Q.-W. and Chang, C.-K. (2013), "Secure homomorphic and searchable encryption in ad hoc networks," *The 42nd International Conference on Parallel Processing*, Lyon, France, 937-942.

Jagannathan, G. and Wright, R.N. (2005), "Privacy-preserving distributed *k*-means clustering over arbitrarily partitioned data," *The 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, Chicago, IL, USA, 593-599.

Jang, M.-H. (2012), "Three-dimensional visualization of an emotional map with geographical information systems: A case study of historical and cultural heritage in the Yeongsan River Basin, Korea," *International Journal of Geographical Information Science*, **26 (8)**, 1393-1413.

Johnston, K., Ver Hoef, J.M., Krivoruchko, K. and Lucas, N. (2001), *Using ArcGIS geostatistical analyst*. http://dusk2.geo.orst.edu/gis/geostat_analyst.pdf

Joseph, J., Sharif, H.O., Sunil, T. and Alamgir, H. (2013), "Application of validation data for assessing spatial interpolation methods for 8-*h* ozone or other sparsely monitored constituents," *Environmental Pollution*, **178 (0)**, 411-418.

Kaleli, C. and Polat, H. (2010), "P2P collaborative filtering with privacy," *Turkish Journal of Electric Electrical Engineering and Computer Sciences*, **8 (1)**, 101-116.

Kalivas, D.P., Kollias, V.J. and Apostolidis, E.H. (2013), "Evaluation of three spatial interpolation methods to estimate forest volume in the municipal forest of the Greek island Skyros," *Geo-spatial Information Science*, **16 (2)**, 100-112.

Kaymaz, I. (2005), "Application of kriging method to structural reliability problems," *Structural Safety*, **27 (2)**, 133-151.

Kerschbaum, F., Schneider, T. and Schröpfer, A. (2013), "Automatic protocol selection in secure two-party computations," *The 20th Network & Distributed System Security Symposium*, San Diego, CA, USA.

Kleijnen, J.P.C. (2009), "Kriging metamodeling in simulation: A review," *European Journal of Operational Research*, **192 (3)**, 707-716.

Kolesnikov, V. and Kumaresan, R. (2013), "Improved OT extension for transferring short secrets," *Lecture Notes in Computer Science*, **8043**, 54-70.

Krige, D.G. (1951), "A statistical approach to some basic mine valuation problems on the Witwatersrand," *Journal of the Chemical, Metallurgical and Mining Society of South Africa*, **52 (6)**, 119-139.

Krasilnikov, P., Carré, F. and Montanarella, L. (2008), *Soil geography and geostatistics-Concepts and applications*, European Communities, Luxembourg.

Kwan, M.-P, Casas, I. and Schmitz, B.C. (2004), "Protection of geoprivacy and accuracy of spatial information: How effective are geographical masks?," *Cartographica: The International Journal for Geographic Information and Geovisualization*, **39 (2)**, 15-28.

Largueche, F.-Z.B. (2006), "Estimating soil contamination with kriging interpolation method," *American Journal of Applied Sciences*, **3 (6)**, 1894-1898.

Leitner, M. and Curtis, A. (2006), "A first step towards a framework for presenting the location of confidential point data on maps—results of an empirical perceptual study," *International Journal of Geographical Information Science*, **20 (7)**, 813-822.

Li, G. and Wang, Y. (2012), "A privacy-preserving classification method based on singular value decomposition," *The International Arab Journal of Information Technology*, **9 (6)**, 529-534.

Li, J. (2008), *A Review of Spatial Interpolation Methods for Environmental Scientists*, Geoscience Australia, Canberra, Australia.

Li, L. and Goodchild, M.F. (2013), "Is privacy still an issue in the era of big data?—Location disclosure in spatial footprints," *The 2013 21st International Conference on Geoinformatics*, Kaifeng, Henan, China, 1-4.

Li, L., Huang, L. and Yang, W. (2011), "Privacy-preserving outlier detection over arbitrarily partitioned data," *The 3rd International Symposium on Information Engineering and Electronic Commerce*, Huangshi, China, 103-106.

Li, X.-B. and Sarkar, S. (2006), "Privacy protection in data mining: A perturbation approach for categorical data," *Information Systems Research*, **17 (3)**, 254-270.

Li, Z., Li, X., Li, C. and Cao, Z. (2010), "Improvement on inverse distance weighted interpolation for ore reserve estimation," *The 2010 7th International Conference on Fuzzy Systems and Knowledge Discovery*, Beijing, China, 1703-1706.

Lindell, Y. and Pinkas, B. (2000), "Privacy preserving data mining," *The 20th Annual International Cryptology Conference on Advances in Cryptology*, Santa Barbara, CA, USA, 36-54.

Lu, G.Y. and Wong, D.W. (2008), "An adaptive inverse-distance weighting spatial interpolation technique," *Computers & Geosciences*, **34 (9)**, 1044-1055.

Ly, S., Charles, C. and Degré, A. (2011), "Geostatistical interpolation of daily rainfall at catchment scale: the use of several variogram models in the Ourthe and Ambleve catchments, Belgium," *Hydrology & Earth System Sciences*, **15 (7)**, 2259-2274.

Meng, Q., Liu, Z. and Borders, B.E. (2013), "Assessment of regression kriging for spatial interpolation – comparisons of seven GIS interpolation methods," *Cartography and Geographic Information Science*, **40 (1)**, 28-39.

Meskine, F. and Bahloul, S.N. (2012), "Privacy preserving *k*-means clustering: A survey research," *The International Arab Journal of Information Technology*, **9 (2)**, 194-200.

Naoum, S. and Tsanis, I.K. (2004), "Ranking spatial interpolation techniques using a GIS-based DSS," *Global Nest: the Intenational Journal*, **6 (1)**, 1-20.

Nayak, G. and Devi, S. (2011), "A survey on privacy preserving data mining: Approaches and techniques," *International Journal of Engineering Science and Technology*, **3 (3)**, 2117-2133.

Niu, Y., Tan, X., Chen, S., Wang, H., Yu, K. and Bu, Z. (2013), "A security privacy protection scheme for data collection of smart meters based on homomorphic encryption," *The IEEE Eurocon 2013*, Zagreb, Croatia, 1401-1405.

Noar, M. and Pinkas, B. (1999), "Oblivious transfer and polynomial evaluation," *The 31st Annual ACM Symposium on Theory of Computing*, Atlanta, GA, USA, 245-254.

Obviex, (2011), *How to calculate the size of encrypted data?* http://www.obviex.com/Articles/CiphertextSize.aspx.

Paillier, P. (1999), "Public-key cryptosystems based on composite degree residuosity classes," *Lecture Notes in Computer Science*, **1592**, 223-238.

Polat, H. and Du, W. (2005), "Privacy-preserving collaborative filtering," *International Journal of Electronic Commerce*, **9 (4)**, 9-35.

Polat, H. and Du, W. (2006), "Achieving private recommendations using randomized response techniques," *Lecture Notes in Computer Science*, **3918**, 637-646.

Prabhakaran, M.M. and Sahai, A. (2013), *Secure Multi-party Computation*, IOS Press, The Netherlands.

Prasad, P.K. and Rangan, C.P. (2007), "Privacy preserving BIRCH algorithm for clustering over arbitrarily partitioned databases," *Lecture Notes in Computer Science*, **4632**, 146-157.

Rivoirard, J. and Romary, T. (2011), "Continuity for kriging with moving neighborhood," *Mathematical Geosciences*, **43 (4)**, 469-481.

Rojas-Avellaneda, D. and Silván-Cárdenas, J.L. (2006), "Performance of geostatistical interpolation methods for modeling sampled data with non-stationary mean," *Stochastic Environmental Research and Risk Assessment*, **20 (6)**, 455-467.

Sachan, A., Roy, D. and Arun, P.V. (2013), "An analysis of privacy preservation techniques in data mining," *Advances in Intelligent Systems and Computing*, **178**, 119-128.

Senyurek, E. and Yakut, I. (2013), "A brief survey on electronic voting," *The 1st International Symposium on Digital Forensics and Security*, Elaziğ, Turkey, 76-79.

Sepehri, M., Cimato, S. and Damiani, E. (2013), "A scalable multi-party protocol for privacy-preserving equality test," *Lecture Notes in Business Information Processing*, **148**, 466-477.

Shad, R., Mesgari, M.S., Abkar, A. and Shad, A. (2009), "Predicting air pollution using fuzzy genetic linear membership kriging in GIS," *Computers, Environment and Urban Systems*, **33 (6)**, 472-481.

Shahbeik, S., Afzal, P., Moarefvand, P. and Qumarsy, M. (2013), "Comparison between ordinary kriging (OK) and inverse distance weighted (IDW) based on estimation error. Case study: Dardevey iron ore deposit, NE Iran," *Arabian Journal of Geosciences*, DOI:10.1007/s12517-013-0978-2.

Shepard, D. (1968), "A two-dimensional interpolation function for irregularly-spaced data," *The 1968 23rd ACM National Conference*, 517-524.

Sun, W., Minasny, B. and McBratney, A. (2012), "Analysis and prediction of soil properties using local regression-kriging," *Geoderma*, **171-172 (0)**, 16-23.

Taur, J.-S., Lin, H.-Y., Lee, H.-L. and Tao, C.-W. (2012), "Data hiding in DNA sequences based on table lookup substitution," *International Journal of Innovative Computing Information and Control*, **8 (10A)**, 6585-6598.

Tobler, W.R. (1979), "Smooth pycnophylactic interpolation for geographical regions," *Journal of the American Statistical Association*, **74 (367)**, 519-530.

Triki, I., Trabelsi, N., Zairi, M. and Dhia, H.B. (2013), "Multivariate statistical and geostatistical techniques for assessing groundwater salinization in Sfax, a coastal region of eastern Tunisia," *Desalination and Water Treatment*, **51 (13-15)**, 2609-2616.

Tugrul, B. and Polat, H. (2013a), "Estimating kriging-based predictions with privacy," *International Journal of Innovative Computing, Information and Control*, **9 (8)**, 3197-3209.

Tugrul, B. and Polat, H. (2013b), "Privacy-preserving inverse distance weighted interpolation," *The Arabian Journal for Science and Engineering*. DOI: 10.1007/s13369-013-0887-4.

Tzeng, W.-G. (2002), "Efficient 1-out-*n* oblivious transfer schemes," *Lecture Notes in Computer Science*, **2274**, 159-171.

Young, C., Martin, D. and Skinner, C. (2009), "Geographically intelligent disclosure control for flexible aggregation of census data," *International Journal of Geographical Information Science*, **23 (4)**, 457-482.

Zhang, L., Li, X.-Y., Liu, Y. and Jung, T. (2013), "Verifiable private multi-party computation: Ranging and ranking," *The 32nd IEEE International Conference on Computer Communications*, Turin, Italy, 605-609.

Zhang, S., Ford, J. and Makedon, F. (2006), "A privacy-preserving collaborative filtering scheme with two-way communication," *The 7th ACM Conference on Electronic Commerce*, Ann Arbor, MI, USA, 316-323.