International Journal on Artificial Intelligence Tools
Vol. 23, No. 5 (2014) 1450007 (22 pages)

DOI: 10.1142/S0218213014500079

World Scientific
www.worldscientific.com

Privacy Preserving Data Mining Using Radial Basis Functions on Horizontally Partitioned Databases in the Malicious Model

Alexandros Panteli*, Manolis Maragoudakis† and Stefanos Gritzalis‡

Department of Information and Communication Systems Engineering
University of the Aegean, Karlovasi
Samos 83200, Greece
**alexander.panteli@gmail.com*
†*mmarag@aegean.gr*
‡*sgritz@aegean.gr*

Received 11 February 2013
Accepted 6 December 2013
Published 28 October 2014

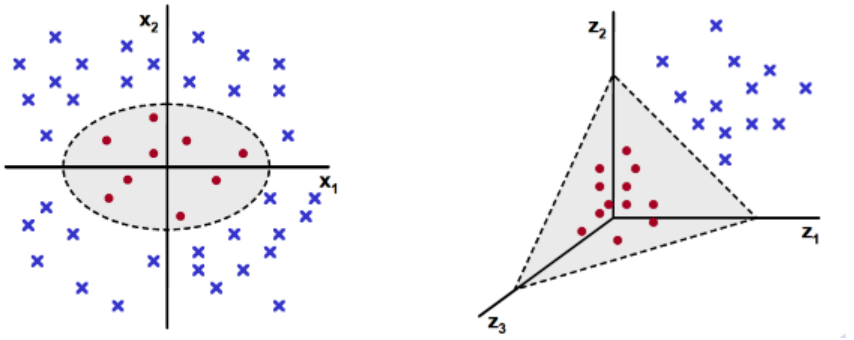
This paper presents a privacy preserving protocol for the computation of a Radial Basis Function (RBF) neural network model between *N* participants which share horizontally partitioned datasets. The RBF model is used for regression analysis tasks. The novel aspect of the proposed protocol lies to the fact that it assumes a malicious user model and does not use homomorphic cryptographic methods, which are inherently only suited for a semi-trusted user environment. The performance analysis shows that the communication overhead is low enough to warranty its use while the computational complexity is identical in most cases with the centralized computation scenario (e.g. a trusted third party). The accuracy of the output model is only marginally subpar to a centralized computation on the union of all datasets.

Keywords: Privacy preserving data mining; radial basis function neural networks.

1. Introduction

Classification and *Regression* belong to a fundamental analytical modeling family of Machine Learning concepts, aiming at predicting the value of a single nominal (for Classification) or continuous (for Regression) attribute, based on the values of other, known, variables. Examples of the aforementioned tasks include the probability estimation of a new, unseen email to belong to the spam category, the forecasting of the price of a stock market, a diagnosis of a certain disease based on the values of a medical test result, etc.

A certain sub-category of Classification and Regression deal with problems that are not linearly separated, thus more advanced practices are required in order capture nonlinear patterns within data. One of the most common practices towards this direction is the introduction of kernel methods, in which data are mapped into higher dimensions, anticipating that there they will exhibit linear patterns. Mapping is a common expression denoting the changing of feature representation.



(a) (b)

Fig. 1. (Color online) A sample dataset where classes are: (a) non linearly separable and (b) after a kernel trick they become linearly separable in a new feature dimensionality.

Consider, for example, the dataset of Fig. 1(a). Each example is defined by a two-feature vector $x = \{x_1, x_2\}$ and a class label of either blue x or red circle. Clearly, there is no linear separator exists for this data. If one introduces a kernel function that maps each initial example $x = \{x_1, x_2\}$ to $z = \left\{x_1^2, \sqrt{2x_1x_2}, x_2^2\right\}$, then data becomes linearly separable in the new representation (Fig. 1(b)).

One of the most known kernel functions is the radial basis function (RBF), which is commonly used in a variety of classifiers, including the homonym RBF neural network as well as Support Vector Machines (SVM). RBF neural networks have a straightforward topology, there are easy to implement and have proven to be robust in noisy, high-dimensional data.[1]

Even though in the majority of situations, classification is performed by a single organization which holds all data by itself, there exists frequent cases where correlated data is collected by different stakeholders. For instance, stores such as supermarkets may hold transactions information for their customers among different branches, hospitals and health centers may collect data from their patients' medical examinations, etc.

This is the case where similar attributes are being collected over different instances, known as *horizontally-partitioned data*. There exists another situation, in which different attributes are collected among the same instances. For example, consider a bank that owns financial data about a certain customer and a real estate company that collects information about the properties of the same customer which can be jointly linked for serving a supposing scenario of a fraud identification task. The latter case is named as *vertically-partitioned data*. Nevertheless, in both cases, mining of joint datasets can lead to more accurate results and is more appropriate than single-source mining.

Privacy is a high priority concern in modern day data processing and analysis. Even though privacy can be preserved when the processing occurs on a centralized dataset belonging to a single organization, this is not the case when multiple datasets are involved. To tackle this problem, numerous approaches have been proposed, which can be roughly divided into two categories.[2] The former category tries to deal with this problem by randomization, perturbation or some other transformation of the data that

minimizes the loss of information, (e.g. through the ability to reconstruct some initial distribution). Some examples are the work by Zahidul *et al.*,[3] Agrawal *et al.*[4] and Zhang *et al*.[5] In the case of the latter approach, the malicious adversary model is adopted.

The latter stream of thought focuses on cryptographic or cryptographic-like (algebraic) methods which, by principle, are trying to solve the secure multi-party computation problem. The work presented in this paper is an approach belonging to this category. Depending on the method used, with the assumption of a semi-trusted user model, there exist a plethora of privacy preserving methods, in order for a selected data mining algorithm to be used. Even though a semi-trusted model is often a realistic one, it does not provide sufficient security when dealing with high risk data such as medical or financial private data. The problem that needs to be solved is the computation of a data mining model using the union of the horizontally-partitioned datasets of N users without compromising the privacy of each user's data.

Formally, we can say that we want to compute a function $f(D_1, D_2, ..., D_n)$ where D_i denotes the dataset of user i with $i = 1, 2, \ldots, N$. Each user should only have knowledge of his/her own data and additionally any knowledge that can be inferred from the result of $f(D_1, D_2, ..., D_n)$ for the data of any other user. The latter definition is also called the *secure multi-party computation* problem. A different way of categorization, as denoted by literature,[4] is the distinction between a system where data miners also possess data and where data processing/mining is done on a central node (trusted party) that does not have any data of its own. The method proposed in this work assumes the first scenario, which resembles a peer to peer distributed system. The present work presents a privacy preserving RBF neural network model, applied on horizontally-partitioned dataset with classification and regression capabilities.

This paper is structured as follows: Section 2 discusses related work on the field and Section 3 introduces the basic notions on the RBF neural network model. Section 4 deals with the description of the proposed protocol, a simple implementation scenario based on the known XOR classification problem, a nonlinearly separated example where RBF is known for its effectiveness plus some systematic evaluation of certain aspects of it such as complexity, security robustness, classification and regression performance, etc. Finally, Section 5 contains some concluding remarks.

2. Related Research

A solution to the generalized problem was given by Yao.[6] In this paper, two assumptions were considered; the function has input from only two users and that a semi trusted model is adopted. Later work by Goldreicht *et al.*,[7] expanded the previous method to incorporate application by N users. Even though the aforementioned methods can privately compute any function f, the computational complexity is highly depended on the complexity of the function and the size of the datasets (in terms of instances and attributes). For a modern day application involving vast databases and complex data mining algorithms (i.e. functions in some sense) these approaches cannot be used effectively.

To alleviate this issue, various privacy preserving versions of existing data mining algorithms have been proposed, some of them using perturbation/randomization instead of trying to solve the generalized problem. The following approaches deal with a large number of existing data mining algorithms from the subdomains of classification, clustering and association discovery. Such examples are privacy preserving Decision Trees,[8] privacy preserving Naïve Bayes,[9] privacy preserving *k*-means clustering,[10] privacy preserving Support Vector Machines (SVM) classification,[11] privacy preserving Bayesian Networks and privacy preserving *A-priori* association rule discovery.[12,13] The common characteristic of the above methods is that they assume a semi-honest user model as mentioned earlier.

This paper presents a privacy preserving method for the computation of the RBF neural network classification/regression model by assuming a malicious user model. The protocol proposed is for horizontally partitioned data and can be used by *N* users.

More recent studies demonstrate a way to privately compute set functions (which can be applied for private calculation of the RBF model) assuming malicious user model.[14] Nevertheless, the drawback of such methods is that they can be only applied for the computation between two users.

3. Radial Basis Function (RBF) Networks

RBF networks are artificial neural networks with one hidden neuron layer.[15] The topology is depicted in Fig. 2. We can interpret this topology as the output of a neural network with one hidden layer of RBF activation functions and a linear output model. It is important to stress the main difference between RBF and multilayer perceptrons. The activation responses of the nodes in RBF are of a local nature (i.e. the output of each RBF node is the same for all points having the same Euclidean distance from their respective center and decreases exponentially (using the *Gaussian* kernel) with the distance). In multilayer perceptrons these responses are of a global nature, meaning that the output of each neuron is the same for all points on a hyperplane. This intrinsic difference has important effects on both convergence and generalization performance.

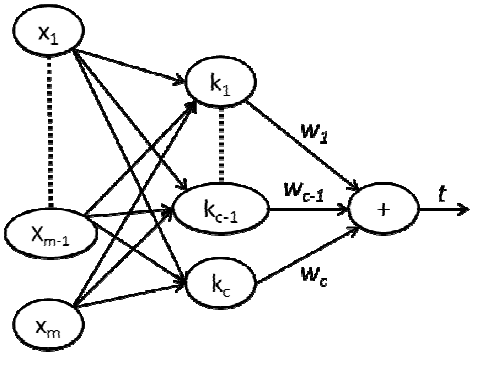


Fig. 2. Architecture of a radial basis function network: An input vector x is used as input to all radial basis functions. The output of the network is a linear combination of the outputs from the radial basis function.

RBF are considered to be fast learners while they require a sufficient number of centers in order to generalize effectively.[16]

Considering the architecture of an RBF neural network again, the kernel function is a function of the Euclidean distance between points. Given a set of m known records (in a form of a vector of n attributes), a number c of them are selected as radial basis functions (or centers k_c), where $c \leq m$. The output t_j, $j = 1, 2, \ldots, m$ is the known class label for input vector s_j (of dimensionality n). It is a linear combination of the weighted outputs from the centers. The exact formula can be given as:

$$t_j = \sum_{i=1}^{c} w_i \rho(\|s_j - k_i\|) \tag{1}$$

where:

- t_j is the known class for vector s_j
- c is the number of selected centers
- k_i denotes the ith center
- ρ is the chosen radial basis function. Usually, the Gaussian kernel, expressed as:

$$\rho(\|s_j - k_i\|) = e^{-\frac{\|s_j - k_i\|^2}{2\sigma^2}} \tag{2}$$

- w_i is the ith weight, corresponding to the k_i center.

The above equation can also be written in matrix form as $\Phi w = t$, where

$$\Phi_{ij} = \rho(\|s_i - k_j\|)$$

In a classification task (note: we consider regression as similar to classification), which belongs to supervised learning processes, we are given a training set of input vectors with the class being annotated (i.e. known).

From the above equation, during training time, we need to solve for w. Since usually we select fewer centers than instances ($c \leq m$), matrix Φ is not square, thus the solution of $\Phi w = t$ is given by: $w = inv(\Phi^T \Phi) \times \Phi^T t$, where function $inv(X)$ denotes the inverse matrix of matrix X if it exists, or a pseudo-inverse matrix of X otherwise.

Then, the calculated weights w can then be used to classify new instances (denoted by s) as such:

$$f(s) = \sum_{i=1}^{c} w_i \rho(\|s - k_i\|) \tag{3}$$

In practical applications, the number of chosen centers c plays an important role for the performance of the classifier. Usually, a clustering algorithm is performed before training in order to eliminate distant (or *noisy*) data instances from being selected as potential centers. Note that a plethora of available kernel functions exists and the proposed protocol is independent of the choice of it. For a thorough insight on RBF theory, please refer to Liu and Bozdogan.[17]

4. Proposed Approach

In order to help technical topics to be explained in a more understandable fashion, the common archetypal characters of Alice and Bob will be adopted. Let us assume that a privacy preserving method for calculating the RBF is available, we also assume that Alice and Bob (the two users) each have a set of data points (with λ and β their respective sizes) and their selected center points. Using this method, both Alice and Bob can privately calculate each cell of the matrix Φ that will be used for the calculations of the classification model.

If the resulting matrix is examined we note that each user can infer the distance of an unknown data point (a data point of the other user) from each of his known centers. If the number of centers is greater than the dimensionality of the data points, all unknown data points can be calculated from this matrix. Therefore, a method for the private computation of the RBF $\rho(x, y)$ is not a solution.

As presented in the preliminaries, the calculation of a vector of weights that fit the current selection of centers/data points and that are used for the prediction of the unknown function can be found by solving the system $\Phi w = t$ or by calculating

$$w = inv(\Phi^T \Phi) \times \Phi^T t \tag{4}$$

The elements of the matrices $\Phi^T\Phi$ and $\Phi^T t$ are by definition:

$$(\Phi^T \Phi)_{i,j} = \sum_{k=1}^{m} \phi_{ki} \phi_{kj} \tag{5a}$$

$$(\Phi^T t)_i = \sum_{j=1}^{m} \phi_{ji} t_j \tag{5b}$$

The above equations show that each cell of the matrices $\Phi^T\Phi$ and $\Phi^T t$ and is calculated using elements of the same row of Φ, therefore since each row of matrix Φ concerns a single data point (belonging to either Alice or Bob) the above can be written as:

$$(\Phi^T \Phi)_{i,j} = \sum_{k \in A} \phi_{ki} \phi_{kj} + \sum_{l \in B} \phi_{li} \phi_{lj} \tag{6a}$$

$$(\Phi^T t)_i = \sum_{k \in A} \phi_{ki} t_k + \sum_{l \in B} \phi_{li} t_l \tag{6b}$$

With the assumption that the centers used are known by both Alice and Bob, Alice can calculate:

$$(\Phi^T \Phi)_{A_{i,j}} = \sum_{k \in A} \phi_{ki} \phi_{kj} \tag{7a}$$

$$(\Phi^T t)_{A_i} = \sum_{k \in A} \phi_{ki} t_k \tag{7b}$$

And similarly Bob calculates

$$(\Phi^T\Phi)_{B_{i,j}} = \sum_{k\in B}\phi_{ki}\phi_{kj} \tag{8a}$$

$$(\Phi^T t)_{B_i} = \sum_{k\in B}\phi_{ki}t_k \tag{8b}$$

They exchange these matrices, sum them, and compute w to obtain the RBF classification model.

To calculate the unknown data points each user must solve a system of $2c^2 + 2c$ nonlinear equations with $\lambda(n + 1)$ unknowns (for Bob) and $\beta(n + 1)$ unknowns (for Alice). If the number of centers c satisfies both, then we have:

$$\lambda(n+1) > 2c^2 + 2c \tag{9a}$$

$$\beta(n+1) > 2c^2 + 2c \tag{9b}$$

Therefore the resulting system is underdetermined. As it will be shown in a subsequent paragraph, the number of unknowns (number of data points of each user multiplied by the dimensionality) can remain private; therefore even the computation of a local solution is infeasible. For simplicity, the condition

$$c < \min(\sqrt{\lambda}, \sqrt{\beta}) \tag{10}$$

is used as the point where the system becomes underdetermined (which satisfies both inequalities given above).

Let u_i denote the ith user, as before the matrices $\Phi^T\Phi$ and $\Phi^T t$ can be written as

$$(\Phi^T\Phi)_{i,j} = \sum_{k\in u_1}\phi_{ki}\phi_{kj} + \sum_{l\in u_2}\phi_{li}\phi_{lj} + \ldots + \sum_{z\in u_N}\phi_{zi}\phi_{zj} \tag{11a}$$

$$(\Phi^T t)_i = \sum_{k\in u_1}\phi_{ki}t_k + \sum_{l\in u_2}\phi_{li}t_l + \ldots + \sum_{z\in u_N}\phi_{zi}t_z \tag{11b}$$

This leads to the conclusion that the method for the private computation of the RBF classification model between two users described above can be extended for the use by N users. Similarly, each user computes the matrices that correspond to his data points and sends it to all other users, w can be calculated from the sums of these matrices.

4.1. *Center selection sub-protocol*

To be able to implement the method described, the two (or N) users have to come to agreement as to how many and what centers to be used (and these have to be known to all participants). The answer to the second question is given by the clustering algorithm or other center selection method used. Since the number of the chosen centers c has to satisfy that

$$c < \min(\sqrt{\lambda_i}), \quad i = 1,2,\ldots,N$$

where λ_i denotes the number of data points owned by user i, a sub-protocol to calculate this number is needed. The above condition is an extension of the condition defined in 6 to be used by N users.

One solution is of the following form:

1. Each user (i) creates a random number r_i, sufficiently large such as it is impossible for any user to have Nr_i records. r_i is chosen in such way so that this value gives no information regarding the number of points held by user i (e.g. knowing that user i has less than one quadrillion client records gives no information since there do not exist one quadrillion people on earth). This number is announced to all participants. At the end of this step all users have a table of the r_i values for $i = 1, 2, \ldots, N$.
2. The user with the smallest value, for example i, r_i reduces the r-values of all users in such manner to satisfy that

$$\sum_{j=1}^{N} r_{nj} < \sqrt{\lambda_i} \qquad \text{and} \qquad \frac{r_{nj}}{r_{ni}} = \frac{r_j}{r_i}, \text{ for each pair } i, j$$

   where r_{nj} is the new value of r_j as is selected by user i. User i is tagged as having executed step 2.
3. Step 2 is repeated until all users have altered the table.
4. The number of centers c is equal to

$$\sum_{j=1}^{N} r_j \tag{12}$$

   and each user i will contribute r_i centers.

4.1.1. *Robustness under statistical attacks*

Since each user has control over the table with the r-values, each user has control over c, and since each user must lower these values, the result will satisfy the inequality given. Obviously, this protocol can be used with the assumption of malicious participants since the only non detectable active attack is the use of a crafted initial value (which does not pose a security risk), the order the users will change the r-values does not matter though. Beyond the satisfaction of the inequality given so that the RBF model can be calculated privately it is important that the number of data points owned by each user is kept private.

Using the above protocol as is, it is straightforward for each adversary to infer which users have reduced the number of centers because the current number of centers did not satisfy the inequality given for them. Therefore they can estimate their number of data points. In order to secure the number of data points of each user, step 2 of the above protocol must use a stochastic model in case that no reduction is necessary. In particular if we assume the number of data points of each user as a set of *iid* discrete random variables, the probability of user i having inadequate number of records (and a reduction of the number of centers is needed) is

$$\left(\frac{m-1}{2m}\right)^{i-1} \tag{13}$$

where m denotes the number of possible values the random variable "number of data points" can have. Simulating the above model with possible values $1000, \ldots, 10\,000$ as the number of data points of each user ($m = 9001$) we observed that on average the number of centers is reduced by 16% if there is a reduction. Therefore we modify step 2 of the aforementioned protocol as such:

2. The user with the smallest value (for example i) r_i.
   If $\sum_{j=1}^{N} r_{nj} \geq \sqrt{\lambda_i}$, reduces the r-values of all users in such manner to satisfy that

$$\sum_{j=1}^{N} r_{nj} < \sqrt{\lambda_i} \quad \text{and} \quad \frac{r_{nj}}{r_{ni}} = \frac{r_j}{r_i} \text{, for each pair } i, j$$

   Else, with probability $[(m - 1)/2m]^{k-1}$ (approximated by $\left(\frac{1}{2}\right)^{k-1}$), where k indicates at what position the ith user is — e.g. fifth so far —, each r_j is reduced by a percentage drawn from a uniform distribution with expected value 16%. r_{nj} is the new value of r_j as is selected by user i. User i is tagged as having executed step 2.

The main drawback of the above sub-protocol is that the final number of centers could be too low, affecting the RBF generalization performance. The two factors that determine the behavior of the protocol are:

- Variation of the database size of each user
- The number of users

Results of the simulation of the above protocol (for the above parameters) are given in Fig. 3, in sub-plots (a), (b) and (c). Each illustration represents a different experiment, where the maximum variation between sizes of users' databases is different. For example, in the first graph the user with the fewest points cannot have less than 1/3 of the points of the user with the most points. The solid black line indicates the maximum number of centers that can be chosen (the number of centers is reduced only if necessary) and the dotted line indicates the number of centers chosen when using the modified protocol.

To summarize, center selection is accomplished by following these steps:

1. All users execute the protocol outined above (modified to protect the number of records of each user) to determine the number of centers each user will contribute (r-values). Simulations show that given a sufficient database size discrepancy the tradeoff between accuracy (number of centers centers) and privacy is not limiting.
2. Each user computes his share of centers using his chosen method (e.g. clustering). Note that center points do not have to be actual data points necessarily.
3. Each user sends these neurons to all other users (round-robin methods for the propagation are susceptible to malicious nodes that do not participate).

Now that all the center neurons are known, the proposed protocol described at the beginning of this section can be used.

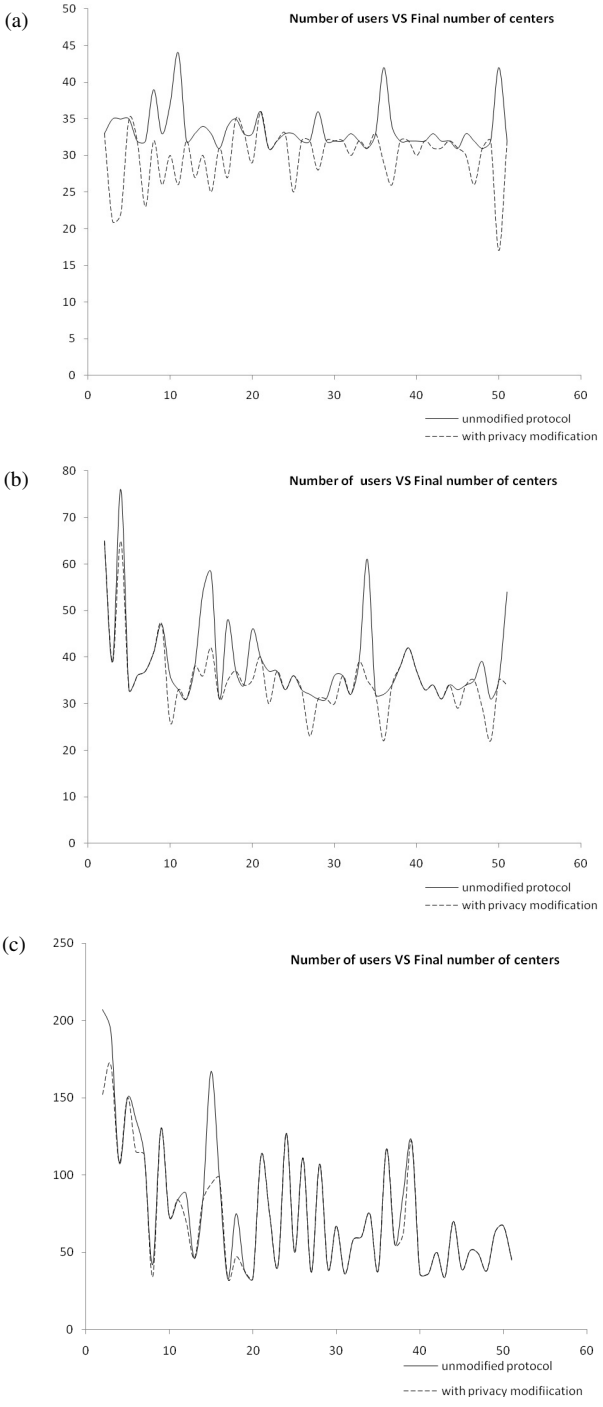


Fig. 3. Final number of centers using both versions of the protocol for various *N*. (a) 300% maximum discrepancy between database sizes, (b) 1 order of magnitude (1000%) and (c) 10 000%.

A subtle detail is that the users must also know the order in which to use the chosen centers (so that the sum of the computed matrices is the RBF model matrix they are trying to compute as described earlier, i.e. use the same row/column ordering when summing matrices). Total ordering is easily achievable by sorting the selected centers by their Euclidean distance from the zero point. The sorting method (increasing or decreasing) as well as the way equal distances are handled are global and predetermined configuration parameters.

4.2. *Additional security considerations*

As mentioned in the previous paragraphs, using the following inequality

$$c < \min(\sqrt{\lambda_i}), \quad i = 1,2,...,N$$

guarantees that the resulting system of equations to be transmitted (by user i) will be underdetermined. To further examine the security provided by the scheme we consider the system of equations received by an adversary from user N, defined by:

$$(\Phi^T \Phi)_{i,j} = \sum_{k \in u_N} \phi_{ki} \phi_{kj}$$

$$(\Phi^T t)_i = \sum_{k \in u_N} \phi_{ki} t_k$$

It is obvious that the terms of each equation is a sum of functions of **all** the unknown values (the u_N set of points held by the user in question). This is due to the fact that the quantity ϕ_{ki} is the result of the kernel function (2) for point k (unknown to adversary) and center i (known to all participants). Therefore any sub-set of equations of this system will have the same number of unknowns and will be underdetermined.

Since the state space is known to all participants, and it could be quite small (e.g. binary columns) it is possible that the true solution can be inferred from the resulting infinite set of solutions given by the underdetermined system.[18–20] Fortunately, the prerequisites for such analysis do not hold, specifically:

- The resulting system is non-linear: Even using a linear kernel function (typically Gaussian though) leads to sums of products of the unknown for most equations. (c^2 out of $c^2 + c$ will be non-linear). Non-linearity can be guaranteed if using a non-linear kernel function.
- The number of unknowns is not known: The sub-protocol presented keeps the number of records of each user private, therefore only a lower bound of c^2 exists. The higher bound is practically infinite as it can be set arbitrarily high for use by the center set size selection sub-protocol.

A short recap tackling the main attack schemes when adopting a malicious model is given below:

- A non-participating party has no effect on honest users, the computation result will be the same as if the malicious user never showed interest in participating. This is obvious by examining Equations (11a) and (11b).

- Departure of a party can be detected using a timeout and by the above point, has no impact on honest users.
- Forging the number of centers used by sending more that allocated is easily detectable.
- Using dummy data and centers only impacts the accuracy of the model and not the security of the data held by other parties. The range of kernel functions is positive therefore the complexity of the system of equations cannot be reduced by specific center neurons.

For the center selection sub-protocol in particular:

- A forged initial r_i to gain priority or be the last user to execute the protocol has no security impact on honest users. The malicious participant could however impact the accuracy of the resulting model if deliberately choosing a new very low value for c. A global "minimum" number of center neurons can be used to alleviate this.
- Each user has control over the number of centers that will be used (and it can only be further lowered). Therefore the inequality $c < \min(\sqrt{\lambda_i}), i = 1, 2, ..., N$ can always be satisfied for honest users.
- The number of records of each user will remain private. The only information given for the number of records of each user i (denoted by λ_i) is that $c < \sqrt{\lambda}_i < Nr_i$, where the upper bound can be arbitrarily high. These boundaries cannot be weakened since the protocol employs a stochastic model for users that did not need to reduce the number of centers to sometimes deliberately do so (in expense of model accuracy).

4.3. *Application example*

Using a well-known, characteristic example of nonlinearly separable data (i.e. the *XOR* function problem illustrated below), a demonstration of the proposed approach is provided.

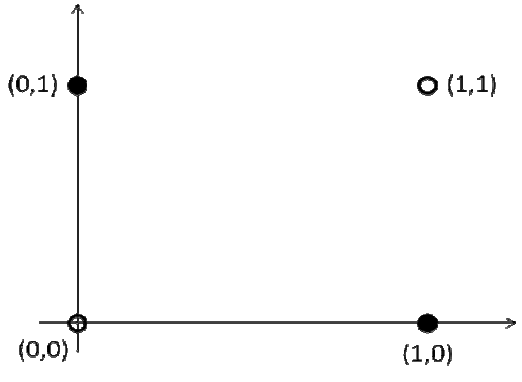


Fig. 4. Data points of the binary XOR problem.

For the XOR problem there exist only four points for classification, these being (0, 0), (0, 1), (1, 0) and (1, 1) respectively. The known classes for these points are 0, 1, 1, 0

(or false, true, true, false) correspondingly. Let us assume that Alice owns (0, 0), (0, 1) and their class and the rest are owned by Bob. The goal is the private computation of the RBF model on the union of these data.

Let us consider first the centralized computation of the RBF classification model for the above problem. Arbitrarily choosing (0, 0) and (1, 1) as the known centers, using Gaussian RBF and with $\sigma = 1$ we can calculate the following values of the radial basis function:

$$e^{\frac{-(\|x_1-c_1\|)^2}{2\sigma^2}} = e^{\frac{-(\|(0,0)-(0,0)\|)^2}{2\sigma^2}} = 1 \qquad e^{\frac{-(\|x_1-c_2\|)^2}{2\sigma^2}} = e^{\frac{-(\|(0,0)-(1,1)\|)^2}{2\sigma^2}} = e^{-1}$$

$$e^{\frac{-(\|x_2-c_1\|)^2}{2\sigma^2}} = e^{\frac{-(\|(0,1)-(0,0)\|)^2}{2\sigma^2}} = e^{-\frac{1}{2}} \qquad e^{\frac{-(\|x_2-c_2\|)^2}{2\sigma^2}} = e^{\frac{-(\|(0,1)-(1,1)\|)^2}{2\sigma^2}} = e^{-\frac{1}{2}}$$

$$e^{\frac{-(\|x_3-c_1\|)^2}{2\sigma^2}} = e^{\frac{-(\|(1,0)-(0,0)\|)^2}{2\sigma^2}} = e^{-\frac{1}{2}} \qquad e^{\frac{-(\|x_3-c_2\|)^2}{2\sigma^2}} = e^{\frac{-(\|(1,0)-(1,1)\|)^2}{2\sigma^2}} = e^{-\frac{1}{2}}$$

$$e^{\frac{-(\|x_4-c_1\|)^2}{2\sigma^2}} = e^{\frac{-(\|(1,1)-(0,0)\|)^2}{2\sigma^2}} = e^{-1} \qquad e^{\frac{-(\|x_4-c_2\|)^2}{2\sigma^2}} = e^{\frac{-(\|(1,1)-(1,1)\|)^2}{2\sigma^2}} = 1$$

Therefore matrix Φ is given by

$$\Phi = \begin{pmatrix} 1 & e^{-1} \\ e^{-\frac{1}{2}} & e^{-\frac{1}{2}} \\ e^{-\frac{1}{2}} & e^{-\frac{1}{2}} \\ e^{-1} & 1 \end{pmatrix}$$

and the linear system $\Phi w = t$ is

$$\begin{pmatrix} 1 & e^{-1} \\ e^{-\frac{1}{2}} & e^{-\frac{1}{2}} \\ e^{-\frac{1}{2}} & e^{-\frac{1}{2}} \\ e^{-1} & 1 \end{pmatrix} \times \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

It follows that

$$\Phi^T\Phi = \begin{pmatrix} 1 & e^{-\frac{1}{2}} & e^{-\frac{1}{2}} & e^{-1} \\ e^{-1} & e^{-\frac{1}{2}} & e^{-\frac{1}{2}} & 1 \end{pmatrix} \times \begin{pmatrix} 1 & e^{-1} \\ e^{-\frac{1}{2}} & e^{-\frac{1}{2}} \\ e^{-\frac{1}{2}} & e^{-\frac{1}{2}} \\ e^{-1} & 1 \end{pmatrix} = \begin{pmatrix} 1+e^{-2}+2e^{-1} & 4e^{-1} \\ 4e^{-1} & 1+e^{-2}+2e^{-1} \end{pmatrix}$$

and

$$\Phi^T t = \begin{pmatrix} 1 & e^{-\frac{1}{2}} & e^{-\frac{1}{2}} & e^{-1} \\ e^{-1} & e^{-\frac{1}{2}} & e^{-\frac{1}{2}} & 1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 2e^{-\frac{1}{2}} \\ 2e^{-\frac{1}{2}} \end{pmatrix}$$

The proposed approach for the computation of the RBF model should produce the same matrices as above. Since only half of the points are known by each user (i.e. rows of Φ), Φ_A which is known to Alice will be

$$\Phi_A = \begin{pmatrix} 1 & e^{-1} \\ e^{-\frac{1}{2}} & e^{-\frac{1}{2}} \end{pmatrix}$$

Therefore we compute $(\Phi^T\Phi)_A$ as

$$\left(\Phi^T\Phi\right)_A = \Phi_A^T\Phi_A = \begin{pmatrix} 1 & e^{-\frac{1}{2}} \\ e^{-1} & e^{-\frac{1}{2}} \end{pmatrix} \times \begin{pmatrix} 1 & e^{-1} \\ e^{-\frac{1}{2}} & e^{-\frac{1}{2}} \end{pmatrix} = \begin{pmatrix} 1+e^{-1} & 2e^{-1} \\ 2e^{-1} & e^{-2}+e^{-1} \end{pmatrix}$$

and $(\Phi^T t)_A$ as

$$\left(\Phi^T t\right)_A = \Phi_A^T t_A = \begin{pmatrix} 1 & e^{-\frac{1}{2}} \\ e^{-1} & e^{-\frac{1}{2}} \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} e^{-\frac{1}{2}} \\ e^{-\frac{1}{2}} \end{pmatrix}$$

(where t_A is the vector for the classes of the points owned by Alice, which is part of t). Similarly, Bob computes

$$\Phi_B = \begin{pmatrix} e^{-\frac{1}{2}} & e^{-\frac{1}{2}} \\ e^{-1} & 1 \end{pmatrix} \Rightarrow$$

$$\left(\Phi^T\Phi\right)_B = \Phi_B^T\Phi_B = \begin{pmatrix} e^{-\frac{1}{2}} & e^{-1} \\ e^{-\frac{1}{2}} & 1 \end{pmatrix} \times \begin{pmatrix} e^{-\frac{1}{2}} & e^{-\frac{1}{2}} \\ e^{-1} & 1 \end{pmatrix} = \begin{pmatrix} e^{-2}+e^{-1} & 2e^{-1} \\ 2e^{-1} & 1+e^{-1} \end{pmatrix}$$

and

$$\left(\Phi^T t\right)_B = \Phi_B^T t_B = \begin{pmatrix} e^{-\frac{1}{2}} & e^{-1} \\ e^{-\frac{1}{2}} & 1 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} e^{-\frac{1}{2}} \\ e^{-\frac{1}{2}} \end{pmatrix}$$

Following the proposed protocol, the users will exchange these matrices and add them, arriving to the same result, given by

$$\left(\Phi^T\Phi\right)_A + \left(\Phi^T\Phi\right)_B = \Phi^T\Phi \qquad \text{and} \qquad \left(\Phi^T t\right)_A + \left(\Phi^T t\right)_B = \Phi^T t \ .$$

It should be noted that in this example the chosen centers were actual data points which is not a pre-requisite for the application of the proposed protocol.

4.4. *Computational complexity*

The computational complexity of the construction of the RBF classification/regression model is defined by the calculation of $\Phi^T\Phi$. Since Φ is not necessarily a square matrix, fast multiplication algorithms like *Strassen* or *Coppersmith* cannot be used, thus the computational complexity of calculating $\Phi^T\Phi$ is $O(mcm)$.

Using the proposed method $\Phi^T\Phi$ and $\Phi^T t$ are computed in a distributed fashion (with a computational cost of $O(mcm)$), the computational overhead is the calculation of $w = inv(\Phi^T\Phi) \times \Phi^T t$ by each user which means N inversions of a cxc matrix and N multiplications of matrices of sizes cxc and $cx1$. This gives a computational complexity of $O(Nc^{2.376})$ using the *Coppersmith-Winograd* algorithm, which, depending on N, can be lower than $O(mcm)$. In this case, the computational complexity is the same for both methods.

4.5. *Communication overhead*

The total communication cost of the proposed protocol is

$$N(N-1)(c^2+c)+(N-1)cn \ \text{ or } \ O(N^2c^2) \tag{14}$$

The first term is the cost of sending the two sub-matrices of all users to all users and the second is the communication cost so that the chosen centers get known to all users. The above do not take into account the center selection sub-protocol that has a communication complexity of $O(N)$, which does not affect the asymptotic complexity.

The non privacy preserving calculation has a cost of $\Omega(Ncn)$ since all users must learn the vector w. Since n and N are comparable quantities a safe conclusion is that the communication overhead is a factor c at most.

4.6. *Security issues within the malicious model*

If the malicious user model is adopted the following general attacks must be considered:

1. Substitution of input
2. Premature protocol abortion
3. Deviation from the protocol
4. No participation

For point 1, it is clear that the substitution of input from some users will lead to an erroneous result but the privacy of the data of honest users is not at risk. This is because in order for these data points to be computed an underdetermined system must be solved, this system of nonlinear equations is guaranteed to be underdetermined since each user

chose so during the center selection. The deviation, abortion, etc., of the protocol are all detectable. This is apparent if we consider a simplified version of the protocol:

1. Determine centers
2. Send data

The center selection protocol presented earlier is secure within the malicious user model parameters. An characteristic of the proposed method is that it is "online" in the sense that if data is available only from, for example, 2 out of 5 users then the result is the same as if the protocol was used from only those two users.

Therefore the abortion of the protocol is not an issue. The only case in which the abortion of the protocol impacts honest users is if this happened after the selection of the number of centers that will be used but before their transmission (detectable using a timeout). In that case the result will be based on fewer centers than the number that could have been used (which implies that the model will be less accurate).

4.7. *Experimental evaluation*

In this section, experiments using the proposed protocol against centralized (local) computation are presented, in both classification and regression modeling tasks. More specifically, datasets from the UCI Machine Learning repository (http://archive.ics.uci.edu/ml/) were used in order to measure the performance of the proposed privacy-preserving RBF methodology. A supplementary experimental round using synthetic data was also performed. In all experiments, the standard, centralized RBF model was compared against the privacy-preserving one, simulating three parties for the protocol. Each party was allocated to a different physical machine, i.e. a Quad-core Intel personal computer with 3,2 GHz CPU and 6 GB of RAM. The local network was set up using a router to connect each machine at 100 Mbit Ethernet connection speed.

The primary goal of the experimental evaluation is to compare the performance between the distributed, privacy-preserving RBF (PP-RBF) method against the local one using:

1. The number of centers that were computed by the proposed protocol.
2. Multiples of these numbers (only for the local computation, in order to demonstrate that it does not affect the performance significantly).

A secondary aim of this study is to measure time in all computations. Since computational complexity has been previously discussed, it is worth mentioning the empirical outcomes in each dataset.

The procedural development of the experiments is outlined below:

1. Data from each set are shared to each machine in an arbitrary distribution, manually chosen to be 15%, 35% and 50% respectively. Recording of total execution time and total number of centers that were computed was followed.
2. Conduct a performance evaluation, using precision and recall for classification and root mean squared error for regression. (Please see below for further details).

3. Conduct a performance evaluation using the aforementioned metrics, on a single machine, using the centralized RBF method with (a) the same number of centers chosen by the privacy-preserving method, (b) twice this number and finally (c) quadruple this number.

4.7.1. *Classification*

Table 1 tabulates the information about each dataset used for classification.

Table 1. Characteristics of the datasets used for classification.

Source	Dataset	Instances	Attributes	Class Values
UCI	Shuttle StatLog	43500	9	7
UCI	Dermatology	366	35	6
UCI	Votes	435	17	2
Synthetic (Matlab)	Quadratic Discriminant Analysis	10000	50	2

As previously described, performance metrics for classification tasks include *Precision* (P) and *Recall* (R), defined as $P = TP/(TP + FP)$ and $R = TP/(TP + FN)$ respectively. The following *confusion matrix* explains the above parameters:

Confusion Matrix			
		Actual Class	
		True	***False***
Predicted class	***True***	TP	FP
	False	FN	TN

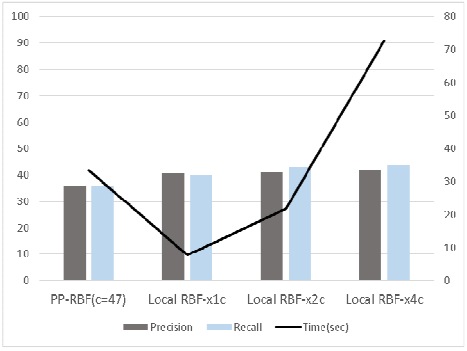
The following figure depict the outcomes of the evaluation process for each dataset.

The left vertical axis measures precision and recall while the right vertical axis measures execution time in seconds. The labels of each columns correspond to the privacy-preserving RBF model (PP-RBF), followed by the number of centers chosen, whereas local execution is denoted as "Local RBF", followed by the multiplier coefficient for the number of centers. For example, in the upper left part of Fig. 5, one can observe that the PP-RBF method has determined that 47 centers is adequate, thus the local execution uses 47, 94 and 188 centers respectively.

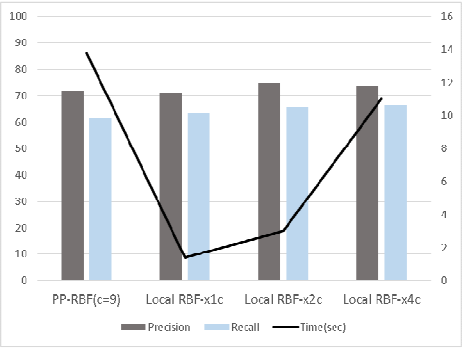
The above results suggest that the privacy-preserving protocol is significantly slower that the centralized implementation, however, as both precision and recall metrics suggest, the overall performance is very similar to the centralized implementation, even when the latter uses significantly larger amount of centers.

4.7.2. *Regression*

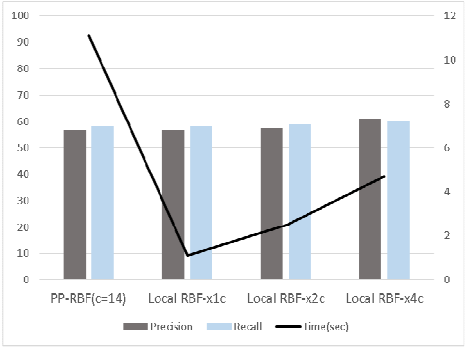
Similarly, the following table presents information on the datasets used for regression analysis. The experimental set was again according to the description of the



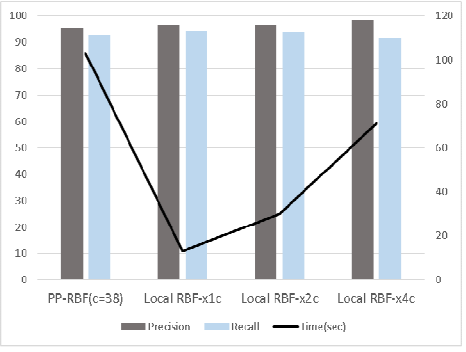
Average Precision/Recall for Shuttle database. Number of centers selected by the PP-RBF, $c = 47$



Average Precision/Recall for Dermatology. Number of centers selected by the PP-RBF, $c = 9$



Average Precision/Recall for Votes database. Number of centers selected by the PP-RBF, $c = 14$



Average Precision/Recall for Synthetic database. Number of centers selected by the PP-RBF, $c = 38$

Fig. 5. (Color online) Experimental results in term of Precision/Recall of the proposed approach against local RBF classification, using variable number of centers, for each dataset.

Table 2. Characteristics of the datasets used for regression.

Source	Dataset	Instances	Attributes
UCI	Forest Fire	517	13
UCI	Automobile	205	26
Synthetic (Matlab)	Sinus	1000	2
Synthetic (Matlab)	Non-linear regression	10000	50

previous section. In this case, the performance metric was chosen to be the *Root Mean Squared Error* (E_i), defined by $E_i = \sqrt{1/n}\sum_{j=1}^{n}(P_{(ij)} - T_j)^2$, where $P_{(ij)}$ is the value that algorithm i predicted for the sample j (from a set of examples) and T_j is the value of the "target value" for the jth example.

Figures 6 and 7 depict the evaluation outcome for the regression tasks as expressed by the error rate and execution time respectively. As one can observe, the behavior of the

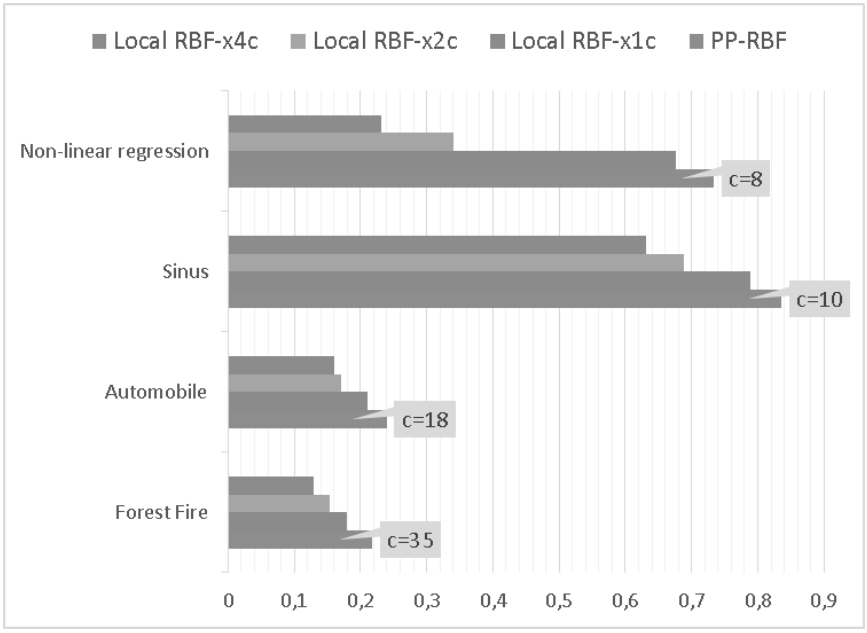


Fig. 6. Experimental results in terms of Root Mean Squared Error of the proposed approach (PP-RBF) against local RBF regression, using variable number of centers, for each dataset. Inside the callout, the number of selected centers by the PP-RBF method.

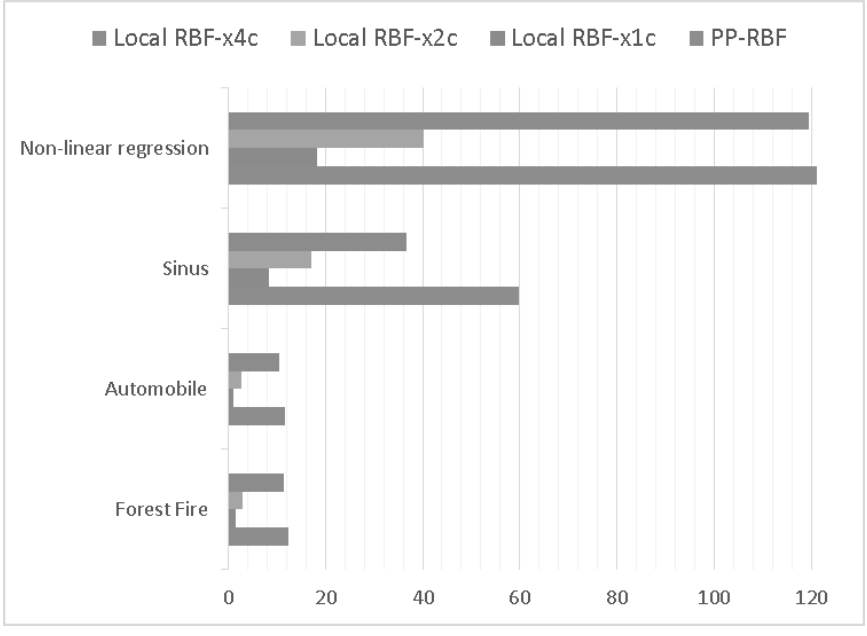


Fig. 7. Experimental results in terms of execution time (in seconds) of the proposed approach (PP-RBF) against local RBF regression, using variable number of centers, for each dataset.

proposed approach is analogous to the classification case, meaning that the computational effort of the proposed method is, as expected, much higher when compared to the centralized one. Nevertheless, the error rate is comparable, particularly for the centralized case that exploits the same number of centers to the privacy-preserving model. The difference between the performance increases when using a larger number of centers. In the case where the local execution incorporates four times the number of centers chosen by PP-RPF, it significantly outperforms the prediction ability of the privacy-preserving protocol. Arguments for a possible elucidation of this occurrence are discussed in the succeeding paragraph.

4.7.3. *Parameter size*

A direct consequence from the parameters set, so that the proposed protocol protects the privacy of each user's data points, is that there is a limitation to the number of centers used. This does not pose a significant problem for discrete target variable problems (classification) since usually the number of data points used is orders of magnitude larger than the number of possible classes, this makes the limitation that $c < \min(\sqrt{\lambda_i})$, $i = 1, 2, \ldots, N$ a very relaxed one.

On the other hand, in regression problems, the number of centers may be directly analogous to the performance (precision, recall) of the model. Nevertheless, studies by Mao *et al.*,[21,22] show that even in regression problems, with the correct selection of neurons/centers the number needed is a very small percentage of the total available data points.

Using the proposed method, the clustering/neuron selection is done on subsets of the available data (each user's data points) which will affect performance. This can be averted by the use of privacy preserving distributed clustering algorithms, for example the method described in Vaidya and Clifton.[10]

5. Conclusions

The present article described a privacy-preserving RBF classification or regression modeling protocol, applied to horizontally-partitioned datasets. As shown above, this method can be used while adopting the malicious user model, since it avoids the pitfall of using homomorphic cryptographic schemes. In order to do so, the neurons used should be known to all users and the number of these neurons is limited by the number of data points the smallest subset has. Points that need further study include a method for proven privacy of the number of data points each user has (for use by the neuron selection sub-protocol) and the further exploitation of the impact on performance the proposed method has. Experimental evaluation in both classification and regression tasks using standard datasets as well as synthetic ones suggest that the private distributed computation of the classification model has comparable results to the centralized one in terms of prediction performance, while its execution time is comparable to the execution time of local computation when the latter is using a significantly larger number of centers.

References

1. S. Haykin, *Neural Networks: A Comprehensive Foundation* (Prentice Hall, 1999).
2. V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis, State-of-the-art in privacy preserving data mining, in *SIGMOD Rec*. **33**(1) (March 2004) 50–57. doi:10.1145/974121.974131
3. Md. Z. Islam and L. Brankovic, A framework for privacy preserving classification in data mining , in *Proc. of the Second Workshop on Australasian Information Security, Data Mining and Web Intelligence, and Software Internationalisation* (ACSW Frontiers '04), Vol. 32 (Australian Computer Society, Inc., Darlinghurst, Australia, 2004), pp. 163–168.
4. R. Agrawal and R. Srikant, Privacy-preserving data mining, *ACM SIGMOD Record* **29**(2) (2000) 439–450.
5. N. Zhang, S. Wang and W. Zhao, A new scheme on privacy-preserving data classification, in *Proc. of the 11th ACM SIGKDD Int. Conf. on Knowledge Discovery in Data Mining* (*KDD '05*), (ACM, New York, NY, USA, 2005), pp. 374–383, doi:10.1145/1081870.1081913.
6. A. Yao, How to generate and exchange secrets, in *27th FOCS* (1986).
7. O. Goldreich, S. Micali and A. Wigderso, How to play any mental game — A completeness theorem for protocols with honest majority, in *Proc. of the 19th Ann. Symp. on the Theory of Computing* (*STOC*) (ACM, 1987), pp. 218–229.
8. Y. Lindell and B. Pinkas, Privacy preserving data mining, in *Advances in Cryptology – CRYPTO 2000* (Springer-Verlag, August 20–24, 2000).
9. J. Vaidya and C. Clifton, Privacy preserving naive bayes classifier for vertically partitioned data, in *SIAM Int. Conf. on Data Mining* (2004).
10. J. Vaidya and C. Clifton, Privacy-preserving k-means clustering over vertically partitioned data, in *The Ninth ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining,* (Washington, DC, August 24–27, 2003).
11. J. Vaidya, H. Yu and X. Jiang, Privacy preserving SVM classification, in *Knowledge and Information Systems* **14**(2) (2008) 161–178.
12. Z. Yang, Privacy-preserving computation of Bayesian networks on vertically partitioned data, in *IEEE Transactions on Data Knowledge Engineering* (*TKDE*) **18**(9) (2006). Earlier versions of parts of the work appeared in *Proc. of the 10th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining* (*KDD*) (2004) and in *Proc. of the Int. Workshop on Privacy Data Management* (2005).
13. Y. Duan, J. Canny and J. Zhan, Efficient privacy-preserving association rule mining: P4P style, in *IEEE Symp. on Computational Intelligence and Data Mining* (*CIDM 2007*), pp. 654–660.
14. Y. Sang and H. Shen, Efficient and secure protocols for privacy-preserving set operations, in *ACM Transactions on Information and System Security* (*TISSEC*) **13**(1) (2009).
15. T. Poggio and F. Girosi, Networks for approximation and learning, in *Proc. of the IEEE* **78**(9) (1990) 1484–1487.
16. S. Theodoridis and K. Koutroumbas, *Pattern Recognition* (Academic Press, 2003).
17. Z. Liu and H. Bozdogan, RBF neural networks for classification using new kernel functions, in *Neural, Parallel & Scientific Computations – Special issue: Advances in Intelligent Systems and Applications* **11** (2003) 41–52.
18. D. Donoho, H. Kakavand and J. Mammen, The simplest solution to an underdetermined system of linear equations, *IEEE Int. Symp. on Information Theory* (2006).
19. B. J. Kubica, Interval methods for solving underdetermined nonlinear equations systems — *SCAN* 2008.
20. C. C. Paige and M. A. Saunders, Solution of sparse indefinite systems of linear equations, *SIAM Journal of Numerical Analysis* **12**(4) (September 1975).

21. K. Z. Mao, RBF neural network center selection based on fisher ratio class separability measure, in *IEEE Transactions on Neural Networks* **13**(5) (September 2002).
22. K. Z. Mao and G.-B. Huang, Neuron selection for RBF neural network classifier based on data structure preserving criterion, in *IEEE Transactions on Neural Networks* **16**(6) (November 2005).