Received April 18, 2016, accepted April 25, 2016, date of publication April 29, 2016, date of current version May 23, 2016.

*Digital Object Identifier 10.1109/ACCESS.2016.2561019*

# Insider Collusion Attack on Privacy-Preserving Kernel-Based Data Mining Systems

**PETER SHAOJUI WANG, FEIPEI LAI, (Senior Member, IEEE), HSU-CHUN HSIAO, AND JA-LING WU, (Fellow, IEEE)**

Department of Computer Science and Information Engineering, National Taiwan University, Taipei 10617, Taiwan

Corresponding author: P. S. Wang (shaojuiwang@gmail.com)

This work was supported by the Ministry of Science and Technology, Taiwan, under Grant 101-2221-E-002-203-MY3.

**ABSTRACT** In this paper, we consider a new insider threat for the privacy preserving work of distributed kernel-based data mining (DKBDM), such as distributed support vector machine. Among several known data breaching problems, those associated with insider attacks have been rising significantly, making this one of the fastest growing types of security breaches. Once considered a negligible concern, insider attacks have risen to be one of the top three central data violations. Insider-related research involving the distribution of kernel-based data mining is limited, resulting in substantial vulnerabilities in designing protection against collaborative organizations. Prior works often fall short by addressing a multifactorial model that is more limited in scope and implementation than addressing insiders within an organization colluding with outsiders. A faulty system allows collusion to go unnoticed when an insider shares data with an outsider, who can then recover the original data from message transmissions (intermediary kernel values) among organizations. This attack requires only accessibility to a few data entries within the organizations rather than requiring the encrypted administrative privileges typically found in the distribution of data mining scenarios. To the best of our knowledge, we are the first to explore this new insider threat in DKBDM. We also analytically demonstrate the minimum amount of insider data necessary to launch the insider attack. Finally, we follow up by introducing several proposed privacy-preserving schemes to counter the described attack.

**INDEX TERMS** Privacy preserving data mining, insider attack, data hiding, kernel.

## I. INTRODUCTION

Data-breaching problems related to *insider attacks* are one of the fastest growing attack types. According to the ‘‘2015 Verizon Data Breach Investigations Report,’’ [1] attacks from ‘‘insider misuse’’ have risen significantly, from 8% in 2013 to 20.6% in 2015. This near-triple rate of increase is astonishing when one considers that this rise has taken place over a span of only two years. As a result of this rapid increase, insider attacks are now among the top three types of data breaches [1]. Insider attacks arise not from system security errors but from staff inside the company's enterprise data security circles. Thus, insider attacks, because of this lack of technical barriers, are simple to carry out successfully. For example, in a single 10-minute phone call to an enterprise chain store, a nontechnical employee can provide enough data to a potential attacker for that attacker to execute a virtual attack—or worse—an impersonation. One call is all it takes for the system to crumble. A company may spend huge sums of hard-earned capital to find technical solutions to protect its perimeter yet still find it difficult to prevent an insider attack.

Many data mining applications store huge amounts of personal information; therefore, extensive research has primarily focused on dealing with potential privacy breaches [2]–[5]. One prime area of research in preserving privacy is the Support Vector Machine (SVM) [4], [5]. SVM is a very popular data mining methodology used mainly with the *kernel trick* to map data into a higher dimensional feature space as well as maintain archives with better mining precision results. With privacy protection in mind, Vaidya et al. provided a state-of-the-art privacy-preserving distributed SVM scheme to securely merge kernels [4], [5]. Their proposal encoded and hid the kernel values in a noisy mixture during transmission such that the original data cannot be recovered even if these distributed organizations colluded.

To the best of our knowledge, no prior work has considered a robust pragmatic model in which ‘‘insiders within organizations’’ collude with outsiders. Such a pragmatic model considers the insider as the key player in sharing data with an attacker, who can then recover the original data from the intermediary kernel values of the SVM model. This attack is more realistic because the attacker needs only to obtain a few data entries rather than the entire database from an organization to successfully recover the rest of the private data [6]. A comparison between our proposed insider collusion attack model and the present models in the privacy-preserving DKBDM area is shown in Fig. 1.

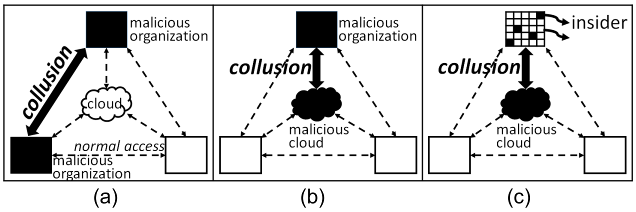


**FIGURE 1.** **Different attack models in DKBDM areas.**

In Fig. 1, a central cloud/transmission server and three participant organizations (represented by squares) cooperate with each other (represented by dashed double-headed arrows) to perform data mining. The black squares denote the malicious attackers, which collude (represented by the wide solid double-headed arrows) to deduce the private data kept by the other participants. Fig. 1 (a) shows some organizations colluding with each other; Fig. 1 (b) implies a malicious cloud colluding with some participant organizations; and Fig. 1 (c) denotes the malicious cloud colluding with insiders within organizations (insiders own only part of the data). Fig. 1 (c) is our proposed attack model. In this work, we first present a situational attack based on insider knowledge as an illustrative example. We then proceed to analyze the minimum amount of data required to launch an insider attack. The minimum number is then given a degree of capacity that characterizes a potential attacker. Finally, we describe several privacy-preserving schemes to deal with the above-mentioned attacks.

## II. RELATED WORK

Researchers whose focus is on security breaches have long considered insider threats as serious concerns that need to be curtailed. However, the approach through which this limitation ought to be implemented varies by institution [7]–[11]. Some important examples of insider threat schemes are mentioned by Claycomb and Nicoll [7] and include the following two cases:

(1) Administrators of a rogue cloud service provider.
(2) Employees in victims' organizations who exploit cloud weaknesses for unauthorized data access.

The first type of insider threat (1) has been addressed most extensively by research. The other case (2), which deals with an organizational insider who exploits vulnerabilities through cloud services to gain unauthorized access to organization systems and data, is the target we seek to tackle in this study.

Some research articles such as those of Lin and Chen [13] propose a random transformation scheme to preserve privacy. The process works by outsourcing the SVM training, thereby ensuring a lack of disclosure to the service provider of the actual content within the database. This privacy-preserving outsourcing method helps protect against the first type of insider threat model mentioned above (1).

Other articles such as those by Goryczka *et al.* [15] propose an ''m-privacy'' technique, which deals with the so-called ''m-adversary'' insider attack. This attack is carried out by an outsider who colludes with $m$ data providers (insiders), allowing inferences about other pieces of information from k-anonymized data. However, their method is limited to and works only for the k-anonymity privacy model [14]. The k-anonymity privacy model is a privacy-preserving data publishing scheme that works to reduce the granularity of data representation such that any given record becomes indistinguishable by mapping it to a number of records ('k') in the data. The ''m-privacy'' approach is orthogonal to our proposed method. Our method is a secure multiparty computation (SMC) scheme for distributed data mining. In this type of method, different parties cooperatively train the model for data mining tasks without revealing the actual data to each other. A distinction between prior work in this area and that of our model is that their insider collusion attack models [4], [5], [16] are defined either as collusion between collaborative organizations or collusion between the cloud and participant organizations, as shown in Fig. 1 (a) (b), while our model considers insiders within organizations as the key contributors for collusion with outsiders, as shown in Fig. 1 (c).

### REVIEW OF KERNEL-BASED DATA MINING METHOD

Here, we review the kernel-based data mining system. The implementation of kernel tricks brings huge improvements to statistical learning theory [17]. These kernel tricks incorporate the use of special functions (kernel functions) to map the original data input to a higher dimensional space, where the data clustering/classification job can be performed better, as shown in Fig. 2.

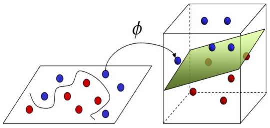


**FIGURE 2.** **After the kernel mapping, nonlinear separable data can become a linear classification case in a higher dimensional space.**[1]

There are several systems for implementing this scheme; however, our focus is on those systems that use kernel values (instead of original data) as the most basic computing unit. For example, Vaidya *et al.* propose a privacy-preserving distributed SVM method for securely merging kernels [4], [5].

[1] *Picture source: https://prateekvjoshi.files.wordpress.com/2012/08/2d-to-3d-projection.jpeg*

Navia-Vázquez *et al.* propose a distributed semiparametric support vector machine (DSSVM) using kernels as basic units to reduce the communication information [18]. Gönen *et al.* propose a multiple kernel machine learning model that describes the kernel as being a basic computing unit capable of possibly merging different types of kernel functions [19], [20]. We will provide more detail about these proposals in the next section.

One of the remarkable features of this type of system is that **data can be hidden in the kernel value safely**. It does this by first performing the kernel computation and then, transforming every pair of the original data (*two vectors*) into a kernel value (*a single number*). In this new format, no one can obtain the original data vectors of "a single number," e.g., for two data vectors $x_1 = [1, 3, 7]$ and $x_5 = [2, 5, 8]$, their linear kernel value $K_{15} = x_1 \cdot x_5 = [1, 3, 7] \cdot [2, 5, 8] = 1 * 2 + 3 * 5 + 7 * 8 = 73$. It is impossible for anyone who knows only the value of $K_{15}$ to obtain the content of $x_1$ and $x_5$.

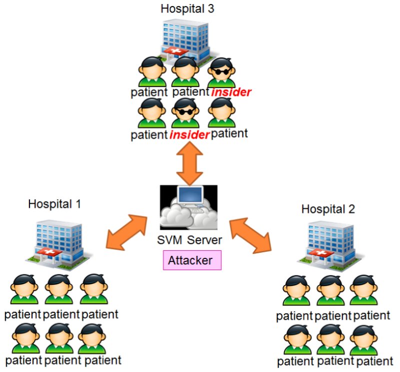


**FIGURE 3. The attacking scenario: Insiders within the hospitals help the outside attacker (the SVM Server) to launch attacks.**

## III. INSIDER COLLUSION ATTACK IN A KERNEL-BASED DATA MINING SYSTEM

First, we will illustrate in detail the privacy breach scenario through an example. As shown in Fig. 3, there are three roles: the SVM server, hospitals (organizations), and patients (members of organization). The *SVM server* provides the SVM service, which builds a global SVM model and performs classification. *Organizations* in this example are represented by three hospitals, which store their patient records separately. Organizations such as hospitals apply an SVM service to their data for data analysis. Some of the *members* are labeled as patients, while others consist of staff including doctors and nurses, each of whom has a ***part*** of the overall patient data records. Some members are **insiders**, who may then collude with outsiders to launch attacks. For example, a member within an organization such as an irresponsible doctor may sell patient records to outsiders. In such a situation, the semi-trusted SVM server acts as an outside attacker, attempting to acquire the entire private patient data with the help of the portion of the patients' records already obtained from the doctor.

### THREAT MODEL

There are three players in the investigated threat scenario:

1. *Data Owners—Organizations or Clients*: These organizations own the data and can be trusted. In a distributed computing environment, they may also participate in data mining tasks.
2. *Insiders*: Members within the data owner's organization are semi-trusted. They may leak their own data to outsiders. The insiders leak nothing but the data content. For example, the data indices do not need to be leaked.
3. *Outsider*: The entity does not belong to the data owner's organization. This group is semi-trusted and may collude with insiders. In a distributed environment, the data mining server, which coordinates sharing among the different subset of affiliate groups, may act as a potential outsider. This outsider (e.g. data mining server) knows the parameters of the mining data, but does not have access to the data content because that has been packed into a kernel format, as described in the following section.

### THE STATE-OF-THE-ART PRIVACY-PRESERVING COOPERATIVE SVM SYSTEM

For a simple explanation, we look to Vaidya's state-of-the-art privacy-preserving SVM System (PPSVM) [4]. In particular, we look at the modified version of Que *et al.* [5], which adds a coordinate server to the PPSVM system to illustrate the implementation of this type of cooperative SVM system. We also use this example system to introduce our proposed insider attacking scheme in Sections 3.1 to 3.3.

PPSVM has the following three procedures:

*Procedure 1 (Local Kernel Matrix Calculation):* Organizations use local data to build a global SVM model in the SVM server from vertically partitioned data. As shown in Fig. 4, from the high-level point of view, the entire set of vertically partitioned data records means that for the whole data matrix, with *m* features and *n* records, each vertical slice represents a different hospital's data. As shown in Fig. 5, Vaidya's local-kernel-merging theorem presents relationships in equations (1) and (2), where $x_i$ and $x_j$ denote two different patient data records, $x_i^r$ means the part of $x_i$ stored in the hospital $r$, $K_{ij}^r$ stands for the local kernel value computed by data vectors

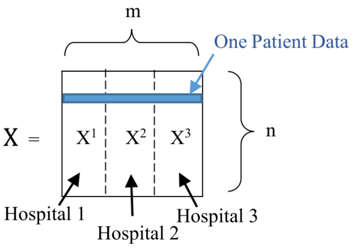


**FIGURE 4. Vertically distributed data with m dimensions (features) and *n* rows (records). Each vertical slice represents a different hospital's data. Here we assume that the total number of hospitals (*z*) is 3.**

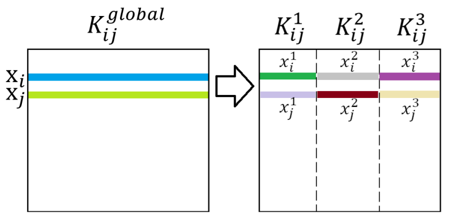


**FIGURE 5.** An illustration of the meaning of the local-kernel-merging theorem given in [4]. Here we also assume that $z = 3$.

$x_i^r$ and $x_j^r$ of the hospital $r$, as shown in equation (3), and $z$ is the total number of hospitals.

$$x_i^T x_j = x_i^{1^T} x_j^1 + x_i^{2^T} x_j^2 + \cdots + x_i^{z^T} x_j^z \quad (1)$$

$$K_{ij}^{global} = K_{ij}^1 + K_{ij}^2 + \cdots + K_{ij}^z \quad (2)$$

$$K_{ij}^r = x_i^{r^T} x_j^r \quad (3)$$

Equations (1) and (2) work for not only linear kernel but also polynomial kernel and RBF kernel. The reason is that both the polynomial kernel (c.f. equation (4)) and RBF kernel (c.f. equation (5)) can be represented as a dot product of data vectors. In equations (4) and (5), $b$ is the bias, $p$ is a power number and $g$ is a gamma parameter.

$$K_{ij} = [(x_i^T x_j) + b]^p \quad (4)$$

$$\begin{aligned} K_{ij} &= \exp\left(-\left\|x_i - x_j\right\|^2/g\right) \\ &= \exp(-(x_i^T x_i - 2x_i^T x_j + x_i^T x_j)/g) \end{aligned} \quad (5)$$

Each organization can compute its linear local kernel values by equation (3) and then, combine the local values to construct the linear global kernel values by equations (1) and (2). If either a polynomial or RBF kernel is needed, they (or a central cloud server) can further apply equation (4) or (5), on the basis of these *linear* global kernel values, to generate their polynomial or RBF kernel results.

In addition, each organization, $h$, holds a *kernel matrix*, $KM^h$. Each element $KM^h$(i, j) is the kernel value, $K_{ij}^h$, made by its own data pair $(x_i^h, x_j^h)$, as shown in Fig. 6.

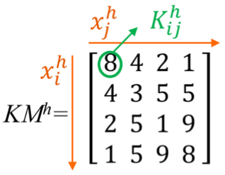


**FIGURE 6.** An example of a 4 × 4 kernel matrix.

*Procedure 2 (Transmitting the Local Kernel Matrix to the Server):* Each organization sends its partial kernel matrices to the SVM Server. The goal is to obtain a merged kernel matrix result used in SVM computation during the next procedure.

*Procedure 3 (Partial Weights Calculation):* The SVM server calculates the global SVM model parameters based on the merged kernel matrix and handles matrix multiplications of high-dimensional data. A local organization can obtain a global SVM model from calculations performed on the server.

In the above system (and other similar systems) because the original data have been packed (hidden) in the kernel values locally, the SVM server and other organizations cannot obtain the data content.

However, we observe that to complete the training/classification task, the system needs to compute every pair of kernel values $K_{ij}$. We also observe that every kernel value is composed of two data vectors. These observations motivate our proposed insider attack method. If there is a malicious insider in this system and the insider holds one of the two data vectors for a kernel value, he may possibly be able to recover the other one based on the known data vector. The proposed attacking steps are detailed as follows.

### A. ATTACK STEP 1: KERNEL COLLECTION

In this step, the malicious outsider collects kernel values from the victim system—as many as possible.

According to Que's modification version of Vaidya's state-of-the-art PPSVM system [4], [5], the server is able to collect "all" the private kernel matrices directly following procedure 2 of the system as described above.

There are many more ways to achieve the goal of collecting kernel values; we will not address them all in this paper. Another example is illustrated by Navia-Vázquez's distributed semiparametric support vector machine (DSSVM) [18]: in which kernels are used as basic units to reducing the amount of information required for communication. In [18], every client needs to send the pairs $\{R_m, r_m\}$ to the other clients for distributed learning, where $R_m$ and $r_m$ are two intermediate matrices consisting of kernel values to reduce the communication load. Note that $r_m = K_m^T W_m y_m$, $K_m$ is the kernel matrix of client $m$, and $W_m$ is a part of the client's entire weighting matrix, whereas $W$, and $y_m$ are the client part of the whole $y$-value matrix. The two parameters $W$ and $y$ are public to all clients. Thus, the client receiving $\{R_m, r_m\}$ could invert$r_m$ to obtain the kernel matrix $K_m$ based on the two public parameters $W$ and $y$. If one of the clients is an attacker, he can obtain all the values of the kernel matrix of the previous client.

### B. ATTACK STEP 2: KERNEL-AND-INSIDER-DATA (KID) LINKING

In this step, an insider colludes (shares his own data) with one of the outsiders, and then, the outsider searches for the kernel values composed from the data of the insider and his collection of kernel values.

The idea of the outsider's search stems from the fact that kernel values are composed on the basis of the insider's data and the other (non-insider) data, as shown in equation (6).

$$Kernel_{ij} = (InsiderData_i)^T (NotInsiderData_j) \quad (6)$$

In addition, this attacking scheme relies on using the kernel matrix, as shown in Fig. 6, which is very popular in

DKBDM area. We continue to use the above example of Vaidya's PPSVM [4], [5] to describe our idea as follows.

For a kernel matrix, if the outsider can obtain $i$ (the index of the insider's data $x_i$), he can easily find all possible generalized kernels, $K_{ij}$. Thus, the first job is to obtain the index value $i$, as shown in Fig. 7. Note that, according to our threat model assumption, the outsider has only the content of the insider's data, rather than its index.

The method for finding the index is explained as follows. As shown in Fig. 8, assume we know the content of data A, B, C and their kernel values. Additionally, assume that we know that *the kernel values of* (A, B) and (A, C) are both unique within in the kernel matrix. If we then carefully observe their indices in the kernel matrix, we could easily find the index of their common element, which would be the index of data A.

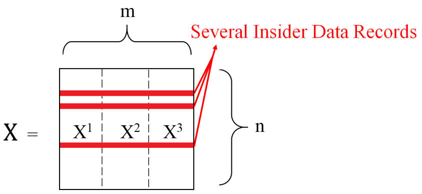


**FIGURE 7.** If the outsider knows the specific data index, he is then able to obtain several of the insider's data records.

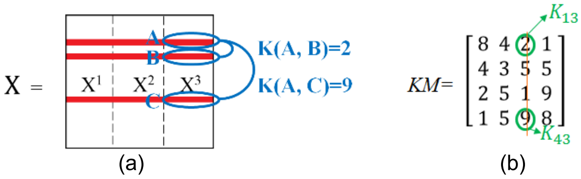


**FIGURE 8.** An example of deducing the index of an insider's data: The index of insider-data A should be 3.

For example, Fig. 8 (a) shows the three data records A, B, C *(represented by the blue ovals)* in Hospital 3's data and the kernel-computation-result values of (A, B) and (A, C) *(represented by the terms near the blue curve lines)*. Fig. 8 (b) presents the kernel matrix of Hospital 3's data, which has two values *(circled in green)* that are exactly equal to the kernel-computation-result values for (A, B) and (A, C) from Fig. 8 (a). We draw an orange linkage line between the two kernel values encircled in green, to present the index of their common elements, and this index of common elements would be the index of data A. The index of insider-data A should be 3.

However, in the real world, there are complex situations, such that the kernel values are "not" unique in the kernel matrix of this organization. Consequently, the likelihood of failure for the above method is high. An example is shown in Fig. 9. (For simplicity, we call the linkage line between the two kernel values a "*kernel line*", e.g., the line drawn between $K_{13}$ and $K_{43}$ in Fig. 8 (b).) To deal with these complex problems efficiently, the following principles are considered:

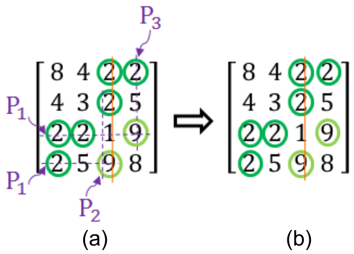


**FIGURE 9.** An example of applying Principles 1∼3. Similar to the example in Fig. 8 (b), however, here are more pairs of kernel values 2 and 9, which lead to more kernel lines.

- Principle 1: Because there is a symmetrical property in the kernel matrix, consider only vertical and horizontal kernel lines.
- Principle 2: Merge the kernel lines for the same axis of the index, because they all represent the same index.
- Principle 3: Remove the kernel lines representing the indices of the other insider's data. Some indices may have been labeled as the insiders' as a result of past search results; therefore, they should not be considered again.

Finally, if more than one kernel line still exists, the possibility of obtaining the correct index is 1/#undecided kernel lines—that is, they are chosen with equal probability.

For example, Fig. 9 shows more than one pair of kernel values 2 and 9, which leads to several kernel lines. The symbol "P$_i$" with an arrow pointing to a kernel line represents Principle $i$ applied to each possible kernel line. The dashed purple kernel lines would be deleted based on the principles; the orange solid line is the real kernel line. Fig. 9 (b) shows that there is only one real kernel line for all these potential kernel lines after applying Principles 1∼3 to Fig. 9 (a).

This Kernel-and-Insider-Data-Linking attack is implemented in Algorithm 1.

### C. ATTACK STEP 3: DATA RECOVERY

The outsider learned which kernel values were composed from which insider's data in step 2. In this step, the outsider recovers the remaining private data inside these kernels, which are composed of one insider's data and one private data. For example, in Fig. 8 (b), all the elements $K_{i3,i\neq3}$, which are $K_{13}$, $K_{23}$, and $K_{43}$ on the orange kernel line are composed of one insider data (insider-data A, whose index is 3) and one unknown private data (the data whose index is $i$). The question is: How can we retrieve the unknown data from the kernel value $K_{i3,i\neq3}$? This is the main focus in attack step 3. We will continue to use the same example of Vaidya's PPSVM system [4], [5] used above to help introduce our idea.

Suppose that in steps 1 and 2, the malicious outsider has successfully collected $n$ insider's data, S$_j$, where $j = 1 \sim n$, and has also collected the kernel values, $K_{ij}$, composed of the insider's data, S$_j$, and a non-insider's data, D$_i$. Assume that in total there are $p$ non-insider's data values in D$_i$, where

**Algorithm 1** Kernel-and-Insider-Data-Linking Attack

```
Require: $m \times m$ kernel matrix $KM$, total $m$ data records $x_1 \sim x_m$, and total $n$ insider's data $s_1 \sim s_n$
for $k = 1 \ldots n$ do
    {Compute $K1$ and $K2$, where $K1$ is the kernel value of $(s_k, s_{p,p \neq k, 1 \leqslant p \leqslant n})$, and $K2$ is the kernel value of $(s_k, s_{q,q \neq k || q \neq p, 1 \leqslant q \leqslant n})$}
    Let $KC1 = []$, $KC2 = []$, $l_1 = 0$, $l_2 = 0$, $IndexCand = []$, $Index = []$
    for for $i = 1 \ldots m$ do //Search for values equal to $K1$ and $K2$ in $KM$
        for $j = 1 \ldots m$ do
            if $KM(i, j) = K1$ then
                $KC1(l_1) = (i, j)$
            else if $KMi, j) = K2$ then
                $KC2(l_2) = (i, j)$
            end if
        end for
    end for
    for $u = 1 \ldots \max(l_1)$ do //Apply Principle 1 & 2 to kernel lines
        for $v = 1 \ldots max(l_2)$ do
            if $KC1(u)[1] \neq KC1(v)[1]$ & $KC1(u)[2] = KC1(v)[2]$ then
                if no element of the array $IndexCand(k) = KC1(u)[2]$ then
                    Insert the element $KC1(u)[2]$ into the array $IndexCand(k)$
                end if
            end if
        end for
    end for
end for
for $k = 1 \ldots n$ do //Apply Principle 3 to kernel lines
    if $\#element\ of\ IndexCand(k) = 1$ then
        $Index(k) = theelementofIndexCand(k)$
    end if
end for
for $k = 1 \ldots n$ do
    if $\#element\ of\ IndexCand(k) > 1$ then
        Delete all elements of $IndexCand(k)$ that has been assigned to the other $Index$
        $Index(k)$ = a randomly chosen element from the remaining elements of $IndexCand(k)$
    end if
end for
```

$i = 1 \sim p$. $S_j$ and $D_i$ are all vectors with $m$ elements, and $S_j(k)$ is the $k$-th element of $S_j$, where $k = 1 \sim m$. This is also true for $D_i(k)$. The goal of attack step 3 is to deduce all the non-insider's data, $D_1 \sim D_p$. First, the outsider uses the following method to deduce one non-insider's data value, $D_u$, and then proceeds to deduce all the other unknown non-insider's data values, one by one, in the same way.

In our illustration, we consider a data mining system operating on the **linear kernel** (the most basic kernel). The linear kernel equation was given above (3). For those linear kernel values composed of the insider's data $S_j$ and the non-insider's data $D_j$, their kernel computation can be expressed as in equation (7), where the value in the parentheses denotes the index of an element of a vector.

$$\begin{aligned} \boldsymbol{K_{ij}} = \mathbf{D}_i \cdot \mathbf{S_j} = \mathrm{D}_i(1) \times \mathrm{S}_j(1) + \mathrm{D}_i(2) \\ \times \mathrm{S}_j(2) + \cdots + \mathrm{D}_i(m) \times \mathrm{S}_j(m) \end{aligned} \tag{7}$$

The ultimate goal of the outsider is to deduce all unknown $D_i$ using the known $K_{ij}$ and $S_j$. To obtain $D_u$ in equation (8), as shown at the top of the next page, the outsider can list a set of $n$ data simultaneously as depicted in equation (9), as shown at the top of the next page. In this manner, he can then solve these $n$ simultaneous equations, which gives him the unique solution content for $D_u$. Subsequently, the attacker can continue to deduce all the other non-insider's data vectors $D_i$ by repeating the same process.

To simplify this concept, we provide an example in the form of vectors: Suppose the attacker wants to obtain the *unknown* data vector $D_2$. He has three known insider's data vectors of dimension **3**, which are,

$$S_1 = [2, 2, -1], S_3 = [-5, 1, 1], \quad \text{and} \quad S_4 = [10, 1, -2]$$

Assume he also has three known linear kernel values composed of the insider's and the *unknown* data vector:

$$K_{21} = 4, \quad K_{23} = 5, \quad \text{and} \quad K_{24} = 5$$

Assume the unknown data vector $D_2$ is

$$D_2 = [a, b, c]$$

Then, the attacker can list a set of simultaneous equations based on equation (7):

$$\begin{cases} 2a + 2b - 1c = 4 \\ -5a + 1b + 1c = 5 \\ 10a + 1b - 2c = 5 \end{cases}$$

The attacker solves these equations and obtains this unique solution:

$$D_2 = [a, b, c] = [2, 5, 10]$$

Then the attacker can continue to deduce all other unknown data vectors in the same way. ■

A DKBDM system, such as Vaidya's PPSVM system [4], [5], can also operate on polynomial or RBF kernels, either of which can be generated from the linear kernel values by equations (4) and (5). In our above scenario, a central cloud server is responsible for calculating equations (4) and (5) to transform linear kernels into polynomial or RBF kernels. Also, in this same example, we make the assumption that the outsider is this central cloud; thus, the cloud can readily obtain the original linear kernel values and then follow up by applying the attacking scheme described above to deduce all the hidden private data.

$$K_{uj} = \mathrm{D}_u \cdot \mathrm{S}_j, \quad j = 1 \sim \mathrm{n} \tag{8}$$

$$\begin{cases} \mathrm{D}_u(1) \times \mathrm{S}_1(1) + \mathrm{D}_u(2) \times \mathrm{S}_1(2) + \cdots + \mathrm{D}_u(m) \times \mathrm{S}_1(m) = K_{u1} \\ \mathrm{D}_u(1) \times \mathrm{S}_2(1) + \mathrm{D}_u(2) \times \mathrm{S}_2(2) + \cdots + \mathrm{D}_u(m) \times \mathrm{S}_2(m) = K_{u2} \\ . \\ . \\ . \\ \mathrm{D}_u(1) \times \mathrm{S}_n(1) + \mathrm{D}_u(2) \times \mathrm{S}_n(2) + \cdots + \mathrm{D}_u(m) \times \mathrm{S}_n(m) = K_{nu} \end{cases} \tag{9}$$

When the attacker is not this central cloud, but rather an individual, the following description shows how a similar scheme can be applied to polynomial or RBF-kernel based systems. The main idea behind this turnout is to mathematically reduce the polynomial/RBF-kernel format to a linear kernel format.

### 1) THE CASE OF POLYNOMIAL KERNEL

The polynomial kernel equation is given by equation (4). The kernel computation for those polynomial kernel values composed of the insider's data $\mathrm{S}_j$ and the non-insider's data $\mathrm{D}_i$ is given by equation (10) and can be rewritten into equation (11):

$$K_{ij} = [(\mathrm{D}_i \cdot \mathrm{S}_j) + b]^p \tag{10}$$

$$\begin{aligned} K'_{ij} &= \sqrt[p]{K_{ij}} - b \\ &= \mathrm{D}_i \cdot \mathrm{S}_j = \mathrm{D}_i(1) \times \mathrm{S}_j(1) + \cdots + \mathrm{D}_i(m) \times \mathrm{S}_j(m) \end{aligned} \tag{11}$$

Our threat model makes the assumption that the outsider knows the parameters $b$ and $p$ needed to carry out the data mining outcomes. It is rather easy for the outsider to transform the value $K_{ij}$ to $K'_{ij}$. Then, the outsider can list $n$ simultaneous equations based on equation (11), which is similar to equation (7); the only difference being that it replaces the kernel values $K'_{ij}$ of equation (11). Hence, the corresponding attacking process will be identical to that of linear kernel.

Furthermore, there are possibly multiple-root solutions to this situation. For polynomial kernel equation (4), if $p$ is even, there are two possible real-value root solutions, $\pm\sqrt[p]{K_{ij}}$. When the value of $K'_{ij}$ cannot be determined, the attack may fail. This experimental depiction is further elaborated in Section 5.3.

### 2) THE CASE OF RBF KERNEL

The RBF kernel mathematical formulation is shown in equation (5). The kernel computation for the RBF kernel values composed of the insider's data $\mathrm{S}_j$ and the non-insider's data $\mathrm{D}_i$ is given by equation (12). Per our threat model assumption, the outsider knows the parameter$g$, that is

$$\begin{aligned} K_{ij} &= \exp(-\left\|\mathrm{D}_i - \mathrm{S}_j\right\|^2/g) \\ &= \exp(-(\mathrm{D}_i \cdot \mathrm{D}_i - 2\mathrm{D}_i \cdot \mathrm{S}_j + \mathrm{S}_j{\cdot}\mathrm{S}_j)/g) \end{aligned} \tag{12}$$

However, unlike the above, the outsider may not easily solve a set of $n$ RBF equations. The reason for this is shown in equation (12), which can be rewritten as equation (13)—that is, in a quadratic form: only $\mathrm{D}_i$ is variable and $\mathrm{S}_j$ is known to the outsider.

$$K'_{ij} = -g \times log(K_{ij}) = (\mathrm{D}_i \cdot \mathrm{D}_i - 2\mathrm{D}_i \cdot \mathrm{S}_j + \mathrm{S}_j{\cdot}\mathrm{S}_j) \tag{13}$$

But there is a trick; we can eliminate the term $\mathrm{D}_i{\cdot}\mathrm{D}_i$ by subtracting the two quadratic components.

Assume the outsider has two kernel values, $K'_{ij}$ and $K'_{i,j-1}$ as denoted in equations (14) and (15).

$$K'_{ij} = \left(\mathrm{D}_i \cdot \mathrm{D}_i - 2\mathrm{D}_i \cdot \mathrm{S}_j + \mathrm{S}_j{\cdot}\mathrm{S}_j\right) \tag{14}$$

$$K'_{i,j-1} = (\mathrm{D}_i \cdot \mathrm{D}_i - 2\mathrm{D}_i \cdot \mathrm{S}_{j-1} + \mathrm{S}_{j-1}{\cdot}\mathrm{S}_{j-1}) \tag{15}$$

Then subtracting $K'_{i,j-1}$ from $K'_{ij}$ gives equation (16):

$$\begin{aligned} &\left(\mathrm{D}_i \cdot \mathrm{D}_i - 2\mathrm{D}_i \cdot \mathrm{S}_j + \mathrm{S}_j{\cdot}\mathrm{S}_j\right) \\ &- \left(\mathrm{D}_i \cdot \mathrm{D}_i - 2\mathrm{D}_i \cdot \mathrm{S}_{j-1} + \mathrm{S}_{j-1}{\cdot}\mathrm{S}_{j-1}\right) = K'_{ij} - K'_{i,j-1} \end{aligned} \tag{16}$$

After rearranging the above equations, he obtains the following equality:

$$-2\mathrm{D}_i \cdot \mathrm{S}_j + 2\mathrm{D}_i \cdot \mathrm{S}_{j-1} = K'_{ij} - K'_{i,j-1} - \mathrm{S}_j \cdot \mathrm{S}_j - \mathrm{S}_{j-1} \cdot \mathrm{S}_{j-1} \tag{17}$$

Because the insider's data vectors $\mathrm{S}_j$, $\mathrm{S}_{j-1}$ and kernel values $K'_{ij}, K'_{i,j-1}$ are known to the outsider, this leaves only the linear components $-2\mathrm{D}_i \cdot \mathrm{S}_j + 2\mathrm{D}_i \cdot \mathrm{S}_{j-1}$ as unknowns for the outsider. Furthermore, to simplify this equation, he reorganizes it as shown in equation (18).

$$\mathrm{D}_i{\cdot}(\mathrm{S}_j - \mathrm{S}_{j-1}) = (\mathrm{S}_j \cdot \mathrm{S}_j + \mathrm{S}_{j-1} \cdot \mathrm{S}_{j-1} + K'_{i,j-1} - K'_{ij})/2 \tag{18}$$

Defining $\mathrm{S}^*_j$ and $K^*_{ij}$ by equations (19) and (20), that is

$$\mathrm{S}^*_j = (\mathrm{S}_j - \mathrm{S}_{j-1}) \tag{19}$$

$$K^*_{ij} = (\mathrm{S}_j \cdot \mathrm{S}_j + \mathrm{S}_{j-1} \cdot \mathrm{S}_{j-1} + K'_{i,j-1} - K'_{ij})/2 \tag{20}$$

then we have

$$\mathbf{D_i{\cdot}S^*_j = K^*_{i,j}} \tag{21}$$

Equation (21) is very similar to the linear kernel in equation (7), with the difference that $\mathrm{S}_j$ and $K_{ij}$ are replaced by $\mathrm{S}^*_j$ and $K^*_{ij}$, respectively. However, the corresponding attacking process will be the same as that of the linear kernel. ■

In this example of PPSVM operating on vertically partitioned data, the attacker can recover one hospital's data by launching attack steps 1–3. Then he can repeat the same process to recover the data from all the other hospitals.

### D. HOW MUCH INSIDER DATA IS REQUIRED TO LAUNCH AN ATTACK?

The existence of a unique solution to the above-mentioned simultaneous equations implies that privacy data can be breached by a malicious outsider. However, a unique solution does not always exist for simultaneous equations. It only possibly exists in the following situation: when the number of equations is equal to or greater than the number of variables. In Section 3.3, the number of equations represents the number of insider, and the number of variables represents the number of data dimensions. In short, ***if the amount of insider data***[2] ***is equal to or larger than the dimensions of the dataset, the insider can breach the privacy.*** Furthermore, if we consider the linear dependence between the data vectors, the final privacy breach rule is the following:

***If the rank of the insider-data matrix***[3] ***is equal to or larger than the dimensions of the dataset, the privacy could be breached by the insider attack.***

Let us take the following dataset as an example. Suppose the malicious outsider wants to obtain an unknown data vector $\mathrm{D}_2$. The outsider knows three insider-data vectors consisting of three dimensions as described below; however, the data vectors $S_3$ and $S_4$ are linear dependent.

$$S_1 = [2, 2, -1], \quad S_3 = [-5, 1, 1], \quad \text{and} \quad S_4 = [-10, 2, 2]$$

The outsider also knows the linear kernel values made by the insider's data vectors and the unknown data vector $\mathrm{D}_2$:

$$K_{21} = 4, \quad K_{23} = 5, \quad \text{and} \quad K_{24} = 10.$$

The outsider assumes the unknown data vector $\mathrm{D}_2$ as follows: $\mathrm{D}_2 = [a, b, c]$.

Then the outsider can then list a set of the equations:

$$\begin{cases} 2a + 2b - 1c = 4 \\ -5a + 1b + 1c = 5 \\ -10a + 2b + 2c = 10 \end{cases}$$

The outsider will then attempt to solve the above equations. However, there is no unique solution to these equations because the second and third equations are the same. It is impossible to solve two equations with three unknown variables. The outsider, having tried to solve the equations, fails to obtain a unique solution. ■

## IV. THE PRIVACY-PRESERVING METHODS

The key to successfully countering the proposed attack with insider collusion lies in preventing the satisfaction of the privacy breach rule described in Section 3.4. In the following two sections we describe our two proposed methods that can effectively counter such attacks.

### A. REDUCING THE NUMBER OF THE INSIDERS

The first method involves reducing the total *number of insiders* until it becomes smaller than the number of data dimensions. The questions that then come to mind are: How can we evaluate the number of insiders? and How can we then identify the insiders to reduce their numbers?

There have been several related research discussions regarding this process [7], [8], [21]. We add our proposed method of identifying the insiders to these as described below.

*Catching Insiders based on Temporal Events*

First, we add an additional dimension of data in every data record. *This extradimensional data will be used in kernel computation*. We assume that this extradimensional data will not complicate the SVM classification result too much. The lack of complication is supposed because the extradimensional data are obtained from preclustering results performed by using clustering schemes such as a self-organizing map on the existing dataset. These schemes can map the original data to very low dimensions as a "thumbnail" of the dataset.

In this way, if there is an insider within an organization, he will need to access the extradimensional data to complete attack step 2 as described above: the kernel-and-insider-data (KID) linking.

Now, suppose that all successful accesses to system resources are logged, including any that access the extradimensional data. The monitoring system can then check if there has been any abnormal access to the extradimensional data based on temporal events.

Step 1: In general cases, in which the organization requests data-mining tasks, there ought to be an access record for the extradimensional data; otherwise, the extradimensional data themselves (which are also accessible) will "likely" be the insider action. For this reason, the data they are targeting for access should be marked as "possible insider's data". This can be performed by a data clustering algorithm, such as k-means, kNN (k Nearest Neighbor), or SVM.

Step 2: During a given time period, the monitoring system may detect this malicious activity, finding several data accesses marked as "possible insider's data." It should then raise an alarm and notify the organization administrators.

Step 3: The organization administrator can quickly check to determine whether those "possible insider's data" accesses are truly the results of malicious activity, or simply erroneous judgement. The administrator performs this check by comparing system logs of authorized access. He can then obtain an estimate of the number of possible insiders.

In Section 5.4, we will show the experiment results after reducing the number of insiders sufficiently (making it smaller than the number of data dimensions) to avoid the privacy breach problem.

[2] *In RBF kernel, the number of insider's data should be the number of the transformed insider's data,* $S_i^* = (S_i - S_{i-1})$*, as shown in equation (21).*

[3] *In RBF kernel, the rank of insider's data should be the rank of the transformed insider's data,* $S_i^* = (S_i - S_{i-1})$*, as shown in equation (21).*

### B. EXPANDING THE DIMENSION OF THE DATA

The second method to counter the attack is to expand the dimension of the dataset, making it larger than the number of insider.

Any method of data dimension expansion that does not involve loss of accuracy can be used here [12], [22]–[24]. In this section, we use our proposed novel geometric transform [25] because its privacy-preserving property enhances the complexity for malicious attackers.

The basic idea of the method is to map a line $x$ to a random curve $(x, y)$, that transforms the data mapped from a 1-dimensional space to a 2-dimensional space with some random perturbations. The mapping function is given in equation (22), where $D_j$ is a 1-dimensional vector, $D'_j$ is its transformed 2-dimensional vector, $D_j(i)$ is the $i$-th element of vector $D_j$, and $\bar{D}_j$ is the normalized result of $D_j$. Here, $k$ and $\theta$ are random parameters for preserving privacy.

$$D'_j(i) = (k \times cos((\bar{D}_j(i) \times 180 + \theta)/2), \times k \sin((\bar{D}_j(i) \times 180 + \theta)/2)) \tag{22}$$

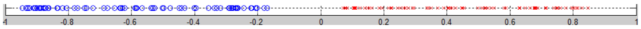


**FIGURE 10. The initial data (There are two clusters: we denote one as the blue dots "o" and another as the red dots "x").**

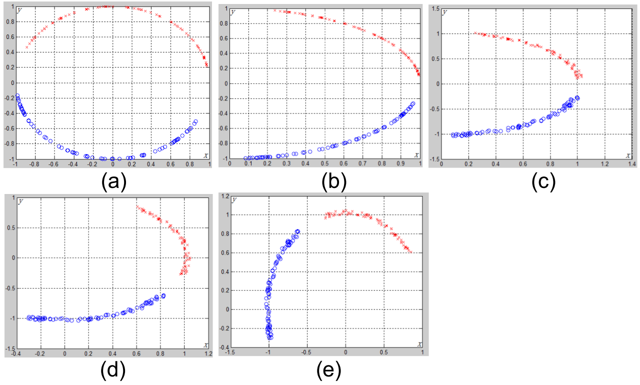


**FIGURE 11. Steps 2–6. (from the upper left to the lower right).**

An example is shown in Figs. 10 and 11. Initially there are two clusters of data in 1 dimension, where one cluster has values distributed between −0.15 and −0.95, denoted by "o" and colored in blue. The other cluster has values distributed between 0.05 and 0.85, denoted by "x" and colored in red. After mapping, the final mapped data distribution retains the clustering effect, but in the higher dimensional space, as shown in Fig. 11 (e). Let us illustrate the idea of the mapping function (22) as follows. Fig. 11 (a) is the result of mapping the normalized data *(1-dim.)* to a unit circle *(2-dim.)*. Fig. 11 (b) then maps the data to a curve as a better clustering result. Then we use some tricks for privacy protection as follows. Fig. 10 (c) adds a random radius $k$ to each node. Fig. 11 (d) adds a random angle$\theta$ for moving all nodes. Fig. 11 (e) randomly exchanges the dimensions of the entire data set.

In Section 5.4, we show the experimental results after applying this mapping method to avoid the privacy breach problem.

## V. EXPERIMENT ANALYSIS

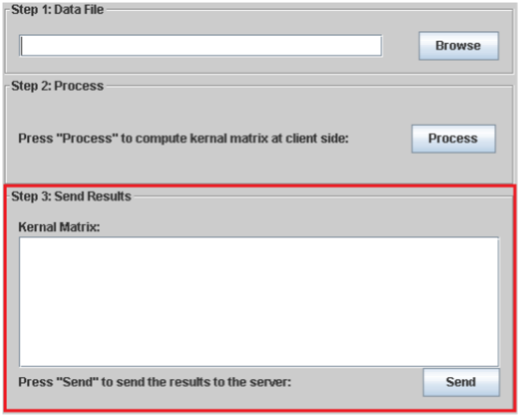
In our experiment, we examine the proposed insider attack scheme and the methods for preserving privacy. The datasets used in the experiment are available at the UCI machine learning repository [26]. We selected some medical and bank credit datasets because they usually have stronger privacy concerns. The selected medical datasets, containing medical records of patients, are Wisconsin breast cancer, Pima Indian diabetes, Statlog heart disease. The bank credit datasets, in which personal information of bank customers is included, are Australian credit and German credit numeric version. The dataset statistics are listed in Table 1, showing how many insiders are needed to execute the attack. In this experiment, we assume all these datasets have the same scenario as in Fig.3. We adopted LIBSVM [27] as the software tool and ran the simulation experiment on a computer with an Intel Core i7-4650U CPU running at 1.7 GHz with 8GB RAM.

**TABLE 1. Dataset statistics and the number of insiders.**

| Dataset | **#Data Dim.** | #Data Records | #Kernel Values | **#Insiders** |
|---|---|---|---|---|
| heart | ***13*** | 100 | 10100 | ***14 (Rank:13)*** |
| breast-cancer | ***10*** | 400 | 160400 | ***11 (Rank:10)*** |
| australian | ***14*** | 400 | 160400 | ***37 (Rank:14)*** |
| diabetes | ***8*** | 400 | 160400 | ***9 (Rank:8)*** |
| germannumer | ***24*** | 800 | 640800 | ***48 (Rank:24)*** |

### A. ATTACK STEP 1: COLLECTING KERNEL VALUES

We ran the simulated experiment on a real-world system, the DPP-SVM online web service [5], as shown in Fig. 12. According to the system's documentation [5], the server can obtain 100% of the kernel values; thus, this makes it the primary vulnerability in our attack model.


Step 1: Data File

Browse

Step 2: Process

Press "Process" to compute kernal matrix at client side:

Process

Step 3: Send Results

Kernal Matrix:

Press "Send" to send the results to the server:

Send

**FIGURE 12. The interface of the DPP-SVM Online Web Service [5]. The local client computes the kernel matrix and sends it to the server.**

### B. ATTACK STEP 2: KERNEL-AND-INSIDER-DATA (KID) LINKING

We ran a simulated KID-Linking attack on the datasets shown in Table 1. To evaluate the threat of the KID -Linking attack, we measure the ratio of the "recognized" insider's data to all insider's data, which we call the "KID-Linking Rate." A higher KID-Linking Rate signifies a greater possibility of data privacy losses. The result is shown in Table 2.

**TABLE 2. Linking the insider's data with their kernel values.**

| Dataset | #Insiders | KID-Linking Rate | | |
|---|---|---|---|---|
| | | Linear Kernel | Poly. Kernel | RBF Kernel |
| heart | 14 (14.00%) | **100%** | **100%** | **100%** |
| breast-cancer | 11 (2.75%) | **100%** | **100%** | **100%** |
| australian | 37 (9.25%) | **100%** | **100%** | **100%** |
| diabetes | 9 (2.25%) | **100%** | **100%** | **100%** |
| germannumer | 48 (6.00%) | **100%** | **97.92%** | **100%** |

*The ratio in (…) is the ratio of insiders to all data.*

### C. ATTACK STEP 3: DATA RECOVERY

In step 3 of the attack, our experiment mainly focuses on how the relationship of the number of insiders and the number of dimensions affects the data recovery rate. This result is shown in Table 3. Note that the data recovery rate for a Polynomial Kernel is lower than the others; this is due to the multi-root problem, as described in Section 3.3.

**TABLE 3. Deducing the private content of SVM training data.**

| Dataset | Rank of Insider's data | #Dim. | Data Recovery Rate | | |
|---|---|---|---|---|---|
| | | | Linear Kernel | Poly. Kernel | RBF Kernel |
| heart | ***13*** *(14)* | ***13*** | **100%** | **100%** | **100%** |
| breast-cancer | ***10*** *(11)* | ***10*** | **100%** | **100%** | **100%** |
| australian | ***14*** *(37)* | ***14*** | **100%** | **52.50%** | **100%** |
| diabetes | ***8*** *(9)* | ***8*** | **100%** | **100%** | **100%** |
| german-numer | ***24*** *(48)* | ***24*** | **100%** | **89.63%** | **100%** |

*The number in parentheses is the total number of insiders in this attack.*

### D. PRIVACY-PRESERVING METHOD 1: REDUCING THE NUMBER OF INSIDERS

The first privacy-preserving method is to reduce the number of insiders. In this simulated experiment, we assume that the organization administrator has applied the proposed method to catch insiders based on temporal events and, eventually, reduces the number of insiders to be smaller than the number of data dimensions. Under this assumption, the outsider tries to do the third attack step to recover data. The results of our experiments show that under this circumstance the data recovery rate is drastically decreased, from an average of 96% to 0%. The result is shown in Table 4

**TABLE 4. Reducing the number of insiders.**

| Dataset | Rank of Insider's Data | #Dim. | Data Recovery Rate | | |
|---|---|---|---|---|---|
| | | | Linear Kernel | Poly. Kernel | RBF Kernel |
| heart | ***10 (11)*** | ***13*** | 0% | 0% | 0% |
| breast-cancer | ***8 (9)*** | ***10*** | 0% | 0% | 0% |
| australian | ***11 (12)*** | ***14*** | 0% | 0% | 0% |
| diabetes | ***6 (7)*** | ***8*** | 0% | 0% | 0% |
| german-numer | ***21 (22)*** | ***24*** | 0% | 0% | 0% |

*The number in parentheses is the total number of insiders in this attack.*

### E. PRIVACY-PRESERVING METHOD 2: EXPANDING THE DIMENSION OF THE DATA

The second enhanced privacy-preserving method is to expand the data dimensions. Based on the method described in Section 4.2, the simulated experiment result shows that the data recovery rate again decreases from an average of 96% to 0%. The result is shown in Table 5. At the same time, the impact on data mining accuracy is limited, decreasing 14% on average in accuracy. Fig. 13 shows a comparison to the result of section 5.3 in privacy loss (data recovery rate).

**TABLE 5. Expanding the dimension of the data.**

| Dataset | Rank of Insider's Data | #Dim. | Data Recovery Rate | | |
|---|---|---|---|---|---|
| | | | Linear Kernel | Poly. Kernel | RBF Kernel |
| heart | ***13*** *(14)* | ***26*** | 0% | 0% | 0% |
| breast-cancer | ***10*** *(11)* | ***20*** | 0% | 0% | 0% |
| australian | ***14*** *(37)* | ***28*** | 0% | 0% | 0% |
| diabetes | ***8*** *(9)* | ***16*** | 0% | 0% | 0% |
| german-numer | ***24*** *(48)* | ***48*** | 0% | 0% | 0% |

*The number in parentheses is the total number of insiders in this attack.*

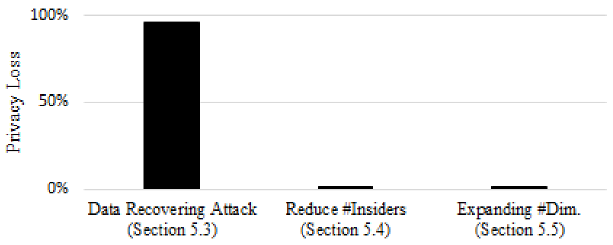


**FIGURE 13. Comparison with the Result of Section 5.3 in Privacy Loss (Data Recovery Rate).**

### F. CHECKING THE PRIVACY BREACH RULE

Finally, for checking the privacy breach rule described in Section 3.4, we repeated the experiments from Sections 5.3-5.5 with different settings for dimensions and

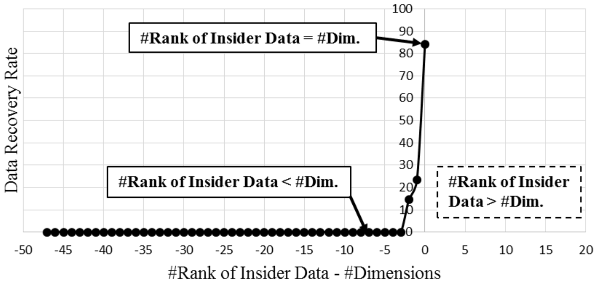


**FIGURE 14. The data recovery rates for different ranks of insider data and numbers of dimensions.**

insider ranks, to record all their average data recovery rates. The results plotted in Fig. 14 show that the privacy breach rule agrees with the experimental result. (Note that it is impossible to have an insider data rank larger than the number of dimensions, due to the rank property for a matrix.)

## VI. CONCLUSION

In this paper, we propose an insider collusion attack that is a threat to most data mining systems that operate on kernels and discuss how many insiders are sufficient to launch this type of attack. We also present two privacy-preserving methods to defend against the attack. Finally, experimental results are provided to prove the effectiveness of the proposed attack and defense schemes.

Note that our proposed attack scheme is not only applicable to the vertically partitioned data but also applicable to horizontally partitioned data and arbitrarily partitioned data [4]; as long as every kernel value is composed of two data vectors and stored in a kernel matrix, our proposed method can reverse those kernel values back to the original data. In fact, most data mining systems operating on kernel computation—especially those in a distributed environment—are potential victims of the proposed attack.

In the future work, we will discuss whether the privacy breach rule described in Section 3.4 can be relaxed, such that even though the exact recovery is not possible, but the attacker can identify the subspace of the private information (corresponding to many solutions to the set of linear equations). Under this situation, how to evaluate the security level of the system?

The scope of this paper is limited to non-homomorphic encryption methods. The main reason for this is that, until now, homomorphic encryption systems have been too slow to be practical. However, if we consider homomorphic encryption based systems such as [28], we believe that the proposed insider threats could lead to a known-plaintext attack, as described in [29]. Of course, we plan to address this issue in future work.

## REFERENCES

[1] *2015 Verizon Data Breach Investigations Report*, Verizon, Bedminster, NJ, USA, 2015.

[2] L. Xu, C. Jiang, J. Wang, J. Yuan, and Y. Ren, "Information security in big data: Privacy and data mining," *IEEE Access*, vol. 2, pp. 1149–1176, Oct. 2014.

[3] C. C. Aggarwal and P. S. Yu, "A general survey of privacy-preserving data mining models and algorithms," in *Privacy-Preserving Data Mining—Models and Algorithms*. USA: Springer, 2008, pp. 10–52.

[4] J. Vaidya, H. Yu, and X. Jiang, "Privacy-preserving SVM classification," *Knowl. Inf. Syst.*, vol. 14, no. 2, pp. 161–178, Feb. 2008.

[5] J. Que, X. Jiang, and L. Ohno-Machado, "A collaborative framework for distributed privacy-preserving support vector machine learning," in *Proc. AMIA Annu. Symp.*, 2012, pp. 1350–1359. [Online]. Available: http://privacy.ucsd.edu:8080/ppsvm/

[6] S. Hartley, *Over 20 Million Attempts to Hack into Health Database*. Auckland, New Zealand: The New Zealand Herald, 2014.

[7] W. R. Claycomb and A. Nicoll, "Insider threats to cloud computing: Directions for new research challenges," in *Proc. IEEE 36th Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Jul. 2012, pp. 387–394.

[8] P. Gaonjur and C. Bokhoree, "Risk of insider threats in information technology outsourcing: Can deceptive techniques be applied?" in *Proc. Int. Conf. Secur. Manage. (SAM)*, Las Vegas, NV, USA, Jun. 2006.

[9] S. Furnell and A. H. Phyo, "Considering the problem of insider IT misuse," *Austral. J. Inf. Syst.*, vol. 10, no. 2, pp. 134–138, 2003.

[10] G. B. Magklaras and S. M. Furnell, "The insider misuse threat survey: Investigating IT misuse from legitimate users," in *Proc. Austral. Inf. Warfare Secur. Conf.*, Perth, WA, Australia, 2004, pp. 1–9.

[11] Cloud Security Alliance (CSA). (2010). *Top Threats to Cloud Computing, Version 1.0*. [Online]. Available: https://cloudsecurityalliance.org/contact/

[12] K. Chen and L. Liu, "Geometric data perturbation for privacy preserving outsourced data mining," *Knowl. Inf. Syst.*, vol. 29, no. 3, pp. 657–695, Dec. 2011.

[13] K.-P. Lin and M.-S. Chen, "Privacy-preserving outsourcing support vector machines with random transformation,"in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 363–372.

[14] L. Sweeney, "Achieving $k$-anonymity privacy protection using generalization and suppression," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 10, no. 05, pp. 571–588, Oct. 2002.

[15] S. Goryczka, L. Xiong, and B. C. M. Fung, "$m$-privacy for collaborative data publishing," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 10, pp. 2520–2533, Oct. 2014.

[16] J. Zhan and S. Matwin, "A crypto-based approach to privacy-preserving collaborative data mining," in *Proc. 6th IEEE Int. Conf. Data Mining Workshops (ICDM)*, Dec. 2006, pp. 546–550.

[17] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[18] A. Navia-Vázquez, D. Gutiérrez-González, E. Parrado-Hernández, and J. J. Navarro-Abellán, "Distributed support vector machines," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 1091–1097, Jul. 2006.

[19] M. Gönen and E. Alpaydın, "Multiple kernel learning algorithms," *J. Mach. Learn. Res.*, vol. 12, pp. 2211–2268, Jul. 2011.

[20] A. Daemen *et al.*, "A kernel-based integration of genome-wide data for clinical decision support," *Genome Med.*, vol. 1, no. 4, p. 39, Apr. 2009.

[21] A. Sarkar, S. Köhler, S. Riddle, B. Ludaescher, and M. Bishop, "Insider attack identification and prevention using a declarative approach," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2014, pp. 265–276.

[22] K. Liu, H. Kargupta, and J. Ryan, "Random projection-based multiplicative data perturbation for privacy preserving distributed data mining," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 1, pp. 92–106, Jan. 2006.

[23] K. Chen, G. Sun, and L. Liu, "Towards attack-resilient geometric data perturbation," in *Proc. 7th SIAM Int. Conf. Data Mining*, 2007, pp. 1–12.

[24] S. R. M. Oliveira and O. R. Zaıane, "Privacy preserving clustering by data transformation," *J. Inf. Data Manage.*, vol. 1, no. 1, p. 37, 2010.

[25] P. S. Wang, S.-W. Chen, C.-H. Kuo, C.-M. Tu, and F. Lai, "An intelligent dietary planning mobile system with privacy-preserving mechanism," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jun. 2015, pp. 336–337.

[26] M. Lichman. (2013). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[27] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, Apr. 2011, Art. no. 27. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm

[28] F. Liu, W. K. Ng, and W. Zhang, "Encrypted SVM for outsourced data mining," in *Proc. IEEE Int. Conf. Cloud Comput. (CLOUD)*, Jun./Jul. 2015, pp. 1085–1092.

[29] D. Vizár and S. Vaudenay, "Cryptanalysis of chosen symmetric homomorphic schemes," *Studia Sci. Math. Hungarica*, vol. 52, no. 2, pp. 288–306, 2015.

**PETER SHAOJUI WANG** received the B.S.E.E. degree from National Taiwan Ocean University, in 2005, and the M.S. degree in computer science from National Central University, in 2007. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Information Engineering, National Taiwan University. He is also a Researcher with Chunghwa Telecom Laboratories, Taiwan. He plays a major role in establishing the Government Public Key Infrastructure in Taiwan. He had won the Outstanding Innovator Award in Chunghwa Telecom Laboratories. His research interests include machine learning, data mining, anonymity and privacy, and network security.

**FEIPEI LAI** (SM'94) received the B.S.E.E. degree from National Taiwan University, in 1980, and the M.S. and Ph.D. degrees in computer science from the University of Illinois at Urbana–Champaign, in 1984 and 1987, respectively. He is currently a Professor with the Graduate Institute of Biomedical Electronics and Bioinformatics, Department of Computer Science and Information Engineering, and the Department of Electrical Engineering, National Taiwan University. He was a Vice Superintendent of National Taiwan University Hospital, the Chairman of the Taiwan Network Information Center, and a Visiting Professor with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, USA. He was also a Guest Professor with the University of Dortmund, Germany, and a Visiting Senior Computer System Engineer with the Center for Supercomputing Research and Development, University of Illinois at Urbana–Champaign. He currently holds ten Taiwan patents and four U.S. patents.

**HSU-CHUN HSIAO** received the B.S. and M.S. degrees in electrical engineering from National Taiwan University, in 2006 and 2008, respectively, and the Ph.D. degree from Carnegie Mellon University, in 2014. He is an Assistant Professor with the Department of Computer Science and Information Engineering and the Graduate Institute of Networking and Multimedia, National Taiwan University, and also an Adjunct Assistant Researcher with the Research Center for Information Technology Innovation, Academia Sinica, Taiwan. Her research interests include network security, anonymity and privacy, and applied cryptography.

**JA-LING WU** (F'08) received the Ph.D. degree in electronics engineering from the Tatung Institute of Technology, Taiwan. He has been a Professor with the Department of Computer Science and Information Engineering, National Taiwan University (NTU), since 1996. From 2004 to 2007, he was appointed as the Head of the Graduate Institute of Networking and Multimedia with NTU. He was awarded one of the lifetime distinguished professors of NTU in 2006.