Privacy-Preserving Clinical Decision Support System using Gaussian Kernel based Classification

Yogachandran Rahulamathavan, *Member, IEEE*, Suresh Veluru, Raphael C.-W Phan, Jonathon A. Chambers, *Fellow, IEEE,* and Muttukrishnan Rajarajan, *Senior Member, IEEE*,

***Abstract*— A clinical decision support system forms a critical capability to link health observations with health knowledge to influence choices by clinicians for improved healthcare. Recent trends towards remote outsourcing can be exploited to provide efficient and accurate clinical decision support in healthcare. In this scenario, clinicians can use the health knowledge located in remote servers via the Internet to diagnose their patients. However, the fact that these servers are third party and therefore potentially not fully trusted raises possible privacy concerns. In this paper, we propose a novel privacy-preserving protocol for a clinical decision support system where the patients' data always remain in encrypted form during the diagnosis process. Hence the server involved in the diagnosis process is not able to learn any extra knowledge about the patient data and results. Our experimental results on popular medical data sets from UCI database demonstrate that the accuracy of the proposed protocol is up to 97.21% and the privacy of patient data is not compromised.**

***Index Terms*— Pricacy, clinical decision support, encryption, classification, support vector machine.**

I. INTRODUCTION

A clinical decision support system is a computerized medical diagnosis process for enhancing health-related decisions and actions with pertinent, organized healthcare knowledge and patient data to improve health and healthcare delivery [1]. Artificial intelligence in machine learning together with biomedical engineering revamp the available clinical data set into healthcare knowledge to build the clinical decision support system [2]–[5]. The current approach uses locally available clinical data sets to build a clinical decision support system. However, the accuracy of the system depends on the availability of sufficient valid clinical data sets but these are not always accessible. As an example, a particular general practitioner (GP) surgery does not generally have sufficient number of samples for all the diseases. Hence, making a correct diagnosis using limited samples is unlikely to be successful.

The recent advances in remote outsourcing techniques (i.e. cloud computing) can be exploited in healthcare to provide efficient and accurate decision support as a service. This service could be utilized by any clinicians in a flexible manner such as on-demand or pay-per use [6]. Within this context, let us consider the following scenario: a third party server builds a clinical decision support system using the existing clinical data set (i.e. assume that the server has rich clinical data set for a particular disease). Now clinicians, who want to verify whether their patients are affected by that particular disease, could send the patient data to the server via the Internet to perform diagnosis based on the healthcare knowledge at the server. This new notion overcomes the difficulties that would be faced by the clinicians such as having to collect a large number of samples (i.e. rich clinical data set), and requiring high computational and storage resources to build their own decision support system.

However, there is now a risk that the third party servers are potentially untrusted servers. Hence, releasing the patient data samples owned by the clinician or revealing the decision to the untrusted server raises privacy concerns. This drawback can affect the adoption of outsourcing techniques in healthcare [7], [8]. Furthermore, the server may not wish to disclose the features of the clinical decision support system even if it offers the service to the clinicians. Hence, in this paper we propose a privacy preserving clinical decision support system which preserves the privacy of the patient data, the decision and the server side clinical decision support system parameters, so that the benefits of the emerging outsourcing technology can also be enjoyed in healthcare sector.

In particular, we consider a decision support system developed using support vector machine (SVM), which is one of the machine learning tools which has been widely used to predict various diseases in biomedical engineering [9]–[11]. Typically, using a SVM consists of two different phases namely training and testing. During the training phase, a classifier will be trained using features of the training data set belonging to different classes. In the testing phase, any unlabeled data sample can be classified and labeled to the corresponding matched class using the trained classifier. In the current setting, the available clinical data set can be used to train a classifier and the trained classifier can be used as a clinical decision support system during the testing phase to make the decision for the patient data.

Depending on the separability of the available training data, the SVM uses particular kernel functions such as linear and non-linear kernels. If the number of features is larger than the number of instances, it is not necessary to map the data into higher dimensional space. It is because non-linear mapping does not improve the performance. Since medical data sets, in general, have less number of features than the number

Yogachandran Rahulamathavan, Suresh Veluru and Muttukrishnan Rajarajan are with the School of Engineering and Mathematical Science, City University London, London, U.K. (e-mail: Yogachandran.Rahulamathavan.1@city.ac.uk; Suresh.Veluru.1@city.ac.uk; R.Muttukrishnan@city.ac.uk).

Raphael C.-W Phan is with Faculty of Engineering, Multimedia University, 63100 Cyberjaya, Malaysia. (e-mail: raphael@mmu.edu.my)

Jonathan A. Chambers is with the Advanced Signal Processing Group, School of Electronic, Electrical and Systems Engineering, Loughborough University, LE11 3TU, Leicestershire, U.K. (e-mail: J.A.Chambers@lboro.ac.uk).

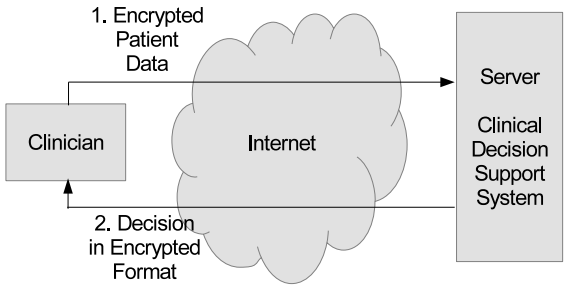

Fig. 1. The overview of a privacy-preserving clinical decision support system.

of instances it is possible to get better classification results with non-linear kernel based SVM. Polynomial and Gaussian kernels are non-linear kernels. The polynomial kernel based model is parametric while the Gaussian kernel based model is non-parametric. In a way a non-parametric model means that the complexity of the model is infinite, its complexity grows with number of instances. In contrast a parametric model's size is fixed, so after a certain point, the model become saturated, and giving more and more instances will not help. It means that the accuracy is dependent on the chosen degree of the polynomial. However, Gaussian kernel finds the best polynomial function in the infinite dimension for the given data set. Hence, we consider Gaussian kernel based classification in this paper.

To the best of our knowledge, we present the first known privacy-preserving clinical decision support system for a Gaussian kernel based SVM. In order to preserve privacy, we re-design the conventional Gaussian kernel based SVM algorithm as an encrypted-domain algorithm using the Paillier homomorphic encryption technique as one of its building blocks [14]. Since Paillier encryption supports only integers and the system variables are continuous and the Gaussian kernel involves exponentiation of negative values, crucially we develop a novel technique to scale the variables, which overcomes these barriers without deteriorating the privacy and performance.

In the system, as shown in Fig. 1, a clinician sends the patient data sample in the encrypted format to the server over the Internet. Then the server exploits the Paillier homomorphic encryption properties to perform the operations directly on the encrypted data, or if there are any operations that cannot be handled by homomorphic properties, then there will be a limited amount of interaction between the clinician and the server based on two-party secure computation protocols [15]. We assume that both the parties will execute the protocol correctly to maintain their reputation, hence we assume that they will behave in a semi-honest manner, i.e. they are honest but curious so privacy is a real issue.

The rest of this paper is organized as follows: In Section II, we describe the conventional SVM, i.e. the steps involved in training the SVM and classification in the plain-domain. Particular focus is placed on the Gaussian kernel method. In Section III we first briefly describe one of the building blocks i.e. homomorphic encryption, and show how SVM classification can be extended to work in the encrypted-domain. Hence, the patient data can remain encrypted even when it is being processed by the server. In particular, the novel technique for scaling variables without deteriorating the performance and privacy is described in Section III.B. We analyze the performance of thus encrypted-domain method in Section IV. We review related works in Section V. Conclusions are discussed in Section VI.

Notation. We use boldface upper and lower case letters for matrices and vectors, respectively; $(.)'$ denotes the transpose operator; $\|.\|_2$ the Euclidean norm; $[\![m]\!]$ the encryption of message m; and $sign(m)$ denotes sign of the number m. The modular reduction operator is denoted by mod.

II. Support Vector Machine

SVMs have been widely used in machine learning for data classification [16], [17]. They have high generalization ability which provides high reliability in real-world applications such as image processing, computer vision, text mining, natural language processing, biomedical engineering and many more [18]–[21]. The goal of a SVM is to separate classes by a classification function, which is obtained by training with the data samples. We describe the classification function of a SVM in the following subsection. This classification function is crucial to derive the privacy-preserving decision support system proposed in Section III.

A. In Plain-Domain

We start with a training set of samples $\tilde{\mathbf{x}}_i \in \mathbb{R}^n$, $i = 1, \ldots, N$ where each sample $\tilde{\mathbf{x}}_i$ belongs to one of the two classes denoted by a label $y_i \in \{-1, +1\}$, $i = 1, \ldots, N$. Using these training data samples we can train a SVM to classify an unlabelled test sample. Before training a SVM, the training data need to be normalized. Normalization keeps the numeric values of training samples on the same scale and prevents samples with a large original scale from biasing the solution. Let us denote the normalized training data samples as $\mathbf{x}_i \in \mathbb{R}^n$, $i = 1, \ldots, N$ where,

$$\mathbf{x}_i = \frac{\tilde{\mathbf{x}}_i - \bar{\mathbf{x}}}{\sigma^2}, \forall i, \tag{1}$$

where $\bar{\mathbf{x}}$ and σ are denote mean and standard deviation of the training data samples. Depending on the separability of the training data, this problem is further divided into either a linear classification problem or a non-linear classification problem.

1) Linear classification problem: The goal of linear classification is to obtain two parallel hyperplanes as shown in Fig. 2, $\mathbf{w}'\mathbf{x} + b = -1$ and $\mathbf{w}'\mathbf{x} + b = +1$, where $\mathbf{w}$ and b are classification parameters obtained during the training process. Both hyperplanes separate the training data of the two classes such that the distance between those hyperplanes is maximized.

After the training stage we can classify an unlabelled test sample, $\tilde{\mathbf{t}} \in \mathbb{R}^n$. Before the classification, the test sample is normalized similar to (1) as

$$\mathbf{t} = \frac{\tilde{\mathbf{t}} - \bar{\mathbf{x}}}{\sigma^2}. \tag{2}$$

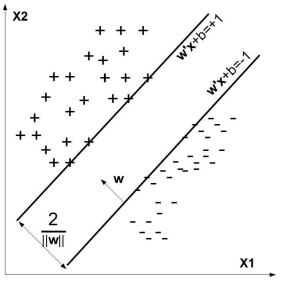

Fig. 2. Training data samples for two different classes are denoted by $+$ and $-$ signs.

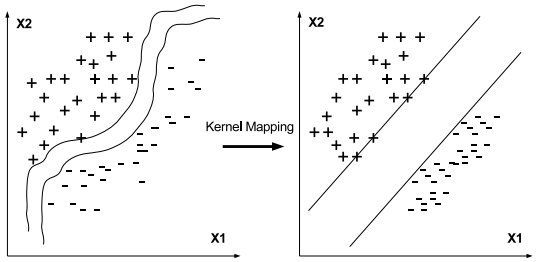

Fig. 3. Non-linear classification problem converted into linear classification problem after the kernel mapping.

Now the normalized test sample, $\mathbf{t}$, can be substituted into the following classification function

$$f(\mathbf{t}) = sign(\mathbf{w}'\mathbf{t} + b) = sign\left(\sum_{s\in S} \alpha_s y_s \mathbf{x}_s'\mathbf{t} + b\right), \quad (3)$$

where $f(\mathbf{t}) \in \{-1, +1\}$, α_i, $i = 1, \ldots, N$ are Lagrangian variables [22] and $\mathbf{x}_s$, $s = 1, \ldots, N$ are support vectors. If $f(\mathbf{t}) = +1$ then the test sample $\tilde{\mathbf{t}}$ belongs to the $+ve$ class else it belongs to the $-ve$ class. Please note that a decision function $d(\mathbf{t})$ can be extracted from (3) as

$$d(\mathbf{t}) = \mathbf{w}'\mathbf{t} + b = \sum_{s\in S} \alpha_s y_s \mathbf{x}_s'\mathbf{t} + b, \quad (4)$$

where $\mathbf{w}'\mathbf{x} + b = 0$ denotes the decision-hyperplane which lies between the two hyperplanes (i.e. $\mathbf{w}'\mathbf{x} + b = -1$ and $\mathbf{w}'\mathbf{x} + b = +1$)

2) Non-linear classification problem: In the previous section, we discussed the classification problem where the training data samples were linearly separable. However, it has been proven in the literature that a similar approach can be used for a non-linear classification problem using kernel methods [23]. Hence, the non-linear classification algorithm is formally similar to the linear classification algorithms except that the dot product between the data samples (i.e. $\mathbf{x}_i'\mathbf{x}_j$) is replaced by various non-linear kernel functions. These kernel functions transform the non-linear classification problem into a linear classification problem by mapping data samples into a higher dimensional feature space (see Fig. 3). In this work we consider only a Gaussian function as kernel, where the dot product between the data samples $\mathbf{x}_s$ and $\mathbf{t}$ in (3) and (4) can be replaced as

$$\mathbf{x}_s'\mathbf{t} \Rightarrow K(\mathbf{x}_s, \mathbf{t}) = e^{-\gamma||\mathbf{x}_s - \mathbf{t}||_2^2}, \quad (5)$$

where $\gamma > 0$. Hence, the classification function in (3) can be modified as

$$f(\mathbf{t}) \quad = \quad sign\underbrace{\left(\sum_{s\in S} \alpha_s y_s e^{-\gamma||\mathbf{x}_s - \mathbf{t}||_2^2} + b\right)}_{\text{decision function}}. \quad (6)$$

Without encryption the server would use (6) to make a decision on the basis of the patient data. We propose a new technique to reformulate (6) in the next section which will preserve the privacy of patient data, decision and server side parameters without compromising the classification performance.

III. Privacy preserving decision support system

In this section, we develop an algorithm which utilizes the healthcare knowledge available in the remote location via the Internet while preserving privacy. Hence, we consider a client-server scenario where the remote server uses (6) as a decision making tool. As shown in Fig. 1, a clinician sends the patient data, $\mathbf{t}$, over the Internet and obtains support from server to make a decision. However, the clinician is reluctant to reveal the patient data or the decision to the server due to privacy concerns. At the same time the server desires not to leak any parameter values of the classification function as thus would be a breach of privacy of the training clinical data samples which relate to other patients. In this section we show how to preserve the privacy of the patient data $\mathbf{t}$ and the decision from the server and the server side parameters from the clinician. First, let us explain the required building blocks in the next section.

A. Homomorphic Encryption

One of the building blocks for our technique is homomorphic encryption. For concreteness and without loss of generality, our descriptions are based on the Paillier cryptosystem [14] although any other homomorphic encryption schemes could be used. The Paillier cryptosystem is an additively homomorphic public-key encryption scheme, whose provable semantic security is based on the decisional composite residuosity problem: it is mathematically intractable to decide whether an integer z is an n-residue modulo n^2 for some composite n, i.e. whether there exists some $y \in \mathbb{Z}_{n^2}^*$ such that $z = y^n \mod n^2$. Let $n = pq$ where p and q are two large prime numbers. A message $m \in \mathbb{Z}_n$ can be encrypted using the Paillier cryptosystem as $[\![m]\!] = g^m r^n \mod n^2$ where $g \in \mathbb{Z}_{n^2}^*$ and $r \in \mathbb{Z}_n^*$. The Paillier cryptosystem is said to be an additively homomorphic cryptosystem because for some given encryptions $[\![m_1]\!]$ and $[\![m_2]\!]$, the encryption $[\![m_1 + m_2]\!]$ of the sum $m_1 + m_2$ in the plain-domain and the encryption $[\![m_1.\alpha]\!]$ of the product of m_1 with a constant α in the plain-domain can respectively be computed efficiently in the encrypted-domain as

$$[\![m_1 + m_2]\!] = [\![m_1]\!][\![m_2]\!], \; [\![m_1.\alpha]\!] = [\![m_1]\!]^\alpha. \quad (7)$$

In the setting considered in this paper, the clinician distributes a public-key to the server while keeping his private-key secret. The server is able to perform encryptions under

TABLE I

OVERVIEW OF VARIABLES WHICH ARE KNOWN TO CLINICIAN AND/OR TO SERVER (KNOWN-✓, UNKNOWN-X).

Variables (in plain-domain)	Known to Clinician	Known to Server
$public-key$	✓	✓
$private-key$	✓	X
$\mathbf{t}$	✓	X
$\alpha_s,\ y_s,\ \gamma,\ \mathbf{x}_s,\ b$	X	✓
c_1	✓	✓
$c_2,\ c_3,\ c_{4,s}\ c_{5,s},\ c_{6,s}$	X	✓
$d_{1,s},\ d_3$	X	✓
$d_{2,s}$	X	X

this public-key and exploits the homomorphic properties of the Paillier cryptosystem to perform the required linear operations in the encrypted-domain. However, only the clinician is able to decrypt any encrypted messages using his corresponding private-key.

B. Decision Support Function in the Encrypted-Domain

In (6), the server knows $\alpha_s,\ y_s,\ \gamma,\ \mathbf{x}_s,\ s \in\ S$ and b in the plain-domain (refer to Table I for the other variables). The clinician encrypts each element of the patient data using the public-key and sends the encrypted data and the corresponding public-key to the server. Note that because the encryption is performed with the clinician's public-key, no one including the server could decrypt this to obtain the values of the elements thus the patient data are protected against being revealed even to the server taking part in this process. Since the server only has the encrypted patient data, it has to compute (6) in the encrypted-domain using homomorphic and two-party secure computation properties.

Generally, the variables associated with (6) are continuous data. Since the Paillier cryptosystem only supports integers, all the variables in (6) will be quantized to the nearest integer value during the computation in the encrypted-domain, which will potentially lead to deterioration of performance [24], [25]. Hence, it is crucial to reformulate (6) into a form which is suitable for encrypted-domain operations. To address this issue, we propose a novel technique for scaling each variable in (6) by a positive large number. More specifically, let us multiply the decision function in (6) by $c_2e^{c_3} > 0$ as

$$f(\mathbf{t}) = sign\left\{c_2e^{c_3}\left[\sum_{s\in S}\alpha_s y_s e^{-\gamma||\mathbf{x}_s-\mathbf{t}||_2^2} + b\right]\right\}, \quad (8)$$

where $c_2,\ c_3 \in \mathbb{R}^+$, hence, the solutions of (6) and (8) are equal. Let us define the scaled decision function in (8) as

$$d(\mathbf{t}) = c_2e^{c_3}\left[\sum_{s\in S}\alpha_s y_s e^{-\gamma||\mathbf{x}_s-\mathbf{t}||_2^2} + b\right] \quad (9)$$

and since $-\gamma||\mathbf{x}_s-\mathbf{t}||_2^2 = -\gamma\mathbf{x}_s'\mathbf{x}_s - \gamma\mathbf{t}'\mathbf{t} + 2\gamma\mathbf{x}_s'\mathbf{t}$, it can be modified as

$$\begin{aligned} d(\mathbf{t}) &= c_2e^{c_3}\left[\sum_{s\in S}\alpha_s y_s e^{-\gamma\mathbf{x}_s'\mathbf{x}_s}e^{-\gamma\mathbf{t}'\mathbf{t}+2\gamma\mathbf{x}_s'\mathbf{t}} + b\right], \\ &= \sum_{s\in S}(c_2\alpha_s y_s e^{-\gamma\mathbf{x}_s'\mathbf{x}_s})(e^{c_3}e^{-\gamma\mathbf{t}'\mathbf{t}+2\gamma\mathbf{x}_s'\mathbf{t}}) + (c_2e^{c_3}b). \end{aligned} \quad (10)$$

Let us define $c_3 = c_{4,s} + c_{5,s} + c_{6,s}$, where $c_{4,s},\ c_{5,s},\ c_{6,s} \in \mathbb{R}^+$. Hence, (10) can be modified as

$$\begin{aligned} d(\mathbf{t}) &= \sum_{s\in S}(c_2\alpha_s y_s e^{-\gamma\mathbf{x}_s'\mathbf{x}_s}e^{c_{4,s}}) \times (e^{c_{5,s}}e^{c_{6,s}-\gamma\mathbf{t}'\mathbf{t}+2\gamma\mathbf{x}_s'\mathbf{t}}) \\ &\qquad + (c_2e^{c_3}b), \\ &= \sum_{s\in S}(c_2\alpha_s y_s e^{-\gamma\mathbf{x}_s'\mathbf{x}_s}e^{c_3-c_{5,s}-c_{6,s}}) \\ &\qquad \times (e^{c_{5,s}}e^{c_{6,s}-\gamma\mathbf{t}'\mathbf{t}+2\gamma\mathbf{x}_s'\mathbf{t}}) + (c_2e^{c_3}b), \end{aligned} \quad (11)$$

and we define

$$d_{1,s} = (c_2e^{c_3-c_{5,s}-c_{6,s}})\alpha_s y_s e^{-\gamma\mathbf{x}_s'\mathbf{x}_s},\ s \in S, \quad (12)$$

$$d_{2,s} = (e^{c_{5,s}})e^{c_{6,s}-\gamma\mathbf{t}'\mathbf{t}+2\gamma\mathbf{x}_s'\mathbf{t}},\ s \in S, \quad (13)$$

$$d_3 = (c_2e^{c_3})b, \quad (14)$$

so that (11) and (8) can be replaced as,

$$d(\mathbf{t}) = \sum_{s\in S} d_{1,s}d_{2,s} + d3, \quad (15)$$

and

$$f(\mathbf{t}) = sign\left\{d(\mathbf{t})\right\}. \quad (16)$$

Note that, $c_2e^{c_3-c_{5,s}-c_{6,s}}$, $c_{5,s}$ and $c_2e^{c_3}$ have respectively been used to scale the variables associated in (12), (13) and (14). Variable $c_{6,s}$ in (13) has been used to mask the value $-\gamma\mathbf{t}'\mathbf{t} + 2\gamma\mathbf{x}_s'\mathbf{t}$. We generate fresh random values of $c_{6,s}$ in the range of $\mathbf{x}_s'\mathbf{x}_s$ for different s values. This masking can be used to preserve the privacy of variables computed by the server. We explain this in detail later in this section. All the variables associated in this section are given in Table I for convenience.

Now the server needs to compute (15) followed by (16) to complete the decision making process. The server knows all the variables associated with (12) and (14) in the plain-domain, hence it can easily compute (12) and (14) without interacting with the clinician. In order to obtain the whole decision function in (15), the server also needs to compute (13). Since the patient data, $\mathbf{t}$ in (13), are available to the server only in the encrypted-domain, the server cannot directly compute (13) in the plain-domain. To proceed, the server needs to normalize the patient data, then compute $c_{5,s}+c_{6,s}-\gamma\mathbf{t}'\mathbf{t}+2\gamma\mathbf{x}_s'\mathbf{t}$ and finally the exponentiation. Let us explain each step in the following subsections.

1) Normalizing the test sample: Before computing (13), the server needs to normalize the patient data as in (2). Denote the patient data at the clinician as $\tilde{\mathbf{t}} = [\tilde{t}_1, \ldots, \tilde{t}_n]'$. The clinician scales each element of $\tilde{\mathbf{t}}$ by $c_1 > 0$ to avoid quantization errors during the encryption. Then the clinician encrypts the scaled patient data and sends $[\![c_1\tilde{\mathbf{t}}]\!] = \left[[\![c_1\tilde{t}_1]\!], \ldots, [\![c_1\tilde{t}_n]\!]\right]'$ and the

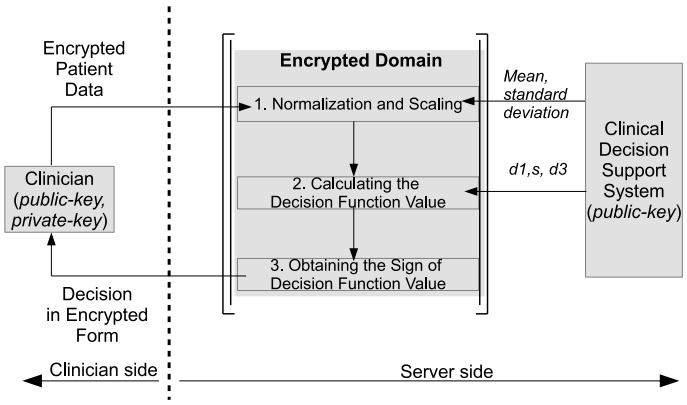

Fig. 4. Privacy-preserving decision support system based on a SVM. The clinician supplies patient data in the encrypted format to the server.

corresponding public-key to the server (see Fig. 4). Now the server will obtain the scaled and normalized patient data in the encrypted-domain using (2) and homomorphic properties as

$$[\![c_1\mathbf{t}]\!] = [\![\frac{\mathbf{c_1}\tilde{\mathbf{t}} - c_1\bar{\mathbf{x}}}{\sigma^2}]\!] = [\![\frac{c_1\tilde{\mathbf{t}}}{\sigma^2} - \frac{c_1\bar{\mathbf{x}}}{\sigma^2}]\!]. \quad (17)$$

Let us define a mean vector $\bar{\mathbf{x}} = [\bar{x}_1, \ldots, \bar{x}_n]'$ and normalized patient data as $\mathbf{t} = [t_1, \ldots, t_n]'$. Hence, each element of (17) is given by

$$[\![c_1 t_i]\!] \quad = \quad [\![\frac{c_1\tilde{t}_i}{\sigma^2} - \frac{c_1\bar{x}_i}{\sigma^2}]\!], \quad \forall i. \quad (18)$$

Since the server knows the vector $\bar{\mathbf{x}}$, and scalars c_1 (assuming both the server and clinician know c_1) and σ in the plain-domain, the server can easily compute the values $-\frac{c_1\bar{x}_i}{\sigma^2} = (-1).\frac{c_1\bar{x}_i}{\sigma^2}$, $\forall i$ and encrypt each of its components by exploiting homomorphic properties $[\![(-1).\frac{c_1\bar{x}_i}{\sigma^2}]\!] = [\![\frac{c_1\bar{x}_i}{\sigma^2}]\!]^{(-1)}$, $\forall i$. Similarly, encryption of $\frac{c_1\tilde{t}_i}{\sigma^2}$ can be obtained as $[\![\frac{c_1\tilde{t}_i}{\sigma^2}]\!] = [\![c_1\tilde{t}_i]\!]^{\frac{1}{\sigma^2}}$, $\forall i$. Hence, the scaled and normalized value of the patient data in (18) can be obtained in the encrypted-domain as follows:

$$\begin{aligned}[\![c_1 t_i]\!] = [\![\frac{c_1\tilde{t}_i}{\sigma^2} - \frac{c_1\bar{x}_i}{\sigma^2}]\!] &= [\![\frac{c_1\tilde{t}_i}{\sigma^2}]\!].[\![-\frac{c_1\bar{x}_i}{\sigma^2}]\!] \quad \forall i, \\ &= [\![c_1\tilde{t}_i]\!]^{\frac{1}{\sigma^2}}.[\![\frac{c_1\bar{x}_i}{\sigma^2}]\!]^{(-1)}, \forall i. \quad (19)\end{aligned}$$

Note that every computation in (19) can be performed by the server without interacting with the clinician. Now the server can use the encrypted, normalized and scaled patient data

$$[\![c_1\mathbf{t}]\!] = [[\![c_1 t_1]\!], \ldots, [\![c_1 t_n]\!]]', \quad (20)$$

to compute (13).

2) Computing $(c_{5,s} + c_{6,s} - \gamma\mathbf{t}'\mathbf{t} + 2\gamma\mathbf{x}_s'\mathbf{t})$ in (13): To do this, let us raise the power of (13) by c_1^3 to yield

$$\begin{aligned}(d_{2,s})^{c_1^3} &= e^{c_1^3(c_{5,s}+c_{6,s}-\gamma\mathbf{t}'\mathbf{t}+2\gamma\mathbf{x}_s'\mathbf{t})}, \ s \in S, \\ &= e^{c_1^3c_{5,s}+c_1^3c_{6,s}-c_1^3\gamma\mathbf{t}'\mathbf{t}+2c_1^3\gamma\mathbf{x}_s'\mathbf{t}}, \ s \in S, \\ &= e^{c_1^3c_{5,s}+c_1^3c_{6,s}+(2c_1^2\gamma\mathbf{x}_s)'(c_1\mathbf{t})-(c_1\gamma)(c_1\mathbf{t})'(c_1\mathbf{t})}, \ s \in S. \quad (21)\end{aligned}$$

In (21), the server knows c_1, $c_{5,s}$, $c_{6,s}$, γ and $\mathbf{x}_s$, $s \in S$ in the plain-domain. Hence, the server can compute the term $c_1^3c_{5,s} + c_1^3c_{6,s}$ in (21) in the plain-domain. Since $c_1\mathbf{t}$ is available at the server only in the encrypted-domain (i.e. (20)), the server needs to exploit the homomorphic properties to compute the term $(2c_1^2\gamma\mathbf{x}_s)'(c_1\mathbf{t})$ in (21) in the encrypted-domain. Let us define $\mathbf{x}_s = [x_{s,1}, \ldots, x_{s,n}]'$, $s \in S$. Now the server computes the term $(2c_1^2\gamma\mathbf{x}_s)'(c_1\mathbf{t})$ in (21) in the encrypted-domain as

$$\begin{aligned}[\![(2c_1^2\gamma\mathbf{x}_s)'(c_1\mathbf{t})]\!] &= [\![\sum_{i=1}^{n}(2c_1^2\gamma x_{s,i})(c_1 t_i)]\!], \\ &= \prod_{i=1}^{n}[\![(2c_1^2\gamma x_{s,i})(c_1 t_i)]\!], \\ &= \prod_{i=1}^{n}[\![c_1 t_i]\!]^{2c_1^2\gamma x_{s,i}}. \quad (22)\end{aligned}$$

Unfortunately, the server cannot compute the term $-(c_1\gamma)(c_1\mathbf{t})'(c_1\mathbf{t})$ in (21) without interacting with the clinician. Hence, the server additively blinds the scaled and normalized patient data with uniformly distributed random vector $\mathbf{r} = [r_1, \ldots, r_n]'$ to obtain $[\![\hat{\mathbf{t}}]\!] = [\![c_1\mathbf{t}+\mathbf{r}]\!] = [\![c_1\mathbf{t}]\!].[\![\mathbf{r}]\!]$. Then the server sends $[\![\hat{\mathbf{t}}]\!]$ to the clinician. The clinician decrypts the received $[\![\hat{\mathbf{t}}]\!]$ and obtains $\hat{\mathbf{t}}$ in the plain-domain. Then the clinician calculates $\hat{\mathbf{t}}'\hat{\mathbf{t}}$ and encrypts and sends $[\![\hat{\mathbf{t}}'\hat{\mathbf{t}}]\!]$ back to the server. Now the server extracts $[\![(c_1\mathbf{t})'(c_1\mathbf{t})]\!]$ from $[\![\hat{\mathbf{t}}'\hat{\mathbf{t}}]\!]$ using homomorphic properties as follows:

$$\begin{aligned}[\![(c_1\mathbf{t})'(c_1\mathbf{t})]\!] &= [\![\hat{\mathbf{t}}'\hat{\mathbf{t}} - 2c_1\mathbf{t}'\mathbf{r} - \mathbf{r}'\mathbf{r}]\!], \\ &= [\![\hat{\mathbf{t}}'\hat{\mathbf{t}}]\!][\![-2c_1\mathbf{t}'\mathbf{r}]\!][\![-\mathbf{r}'\mathbf{r}]\!], \\ &= [\![\hat{\mathbf{t}}'\hat{\mathbf{t}}]\!].[\![-\mathbf{r}'\mathbf{r}]\!].\prod_{i=1}^{n}[\![c_1 t_i]\!]^{-2r_i} \quad (23)\end{aligned}$$

The server then computes $[\![-(c_1\gamma)(c_1\mathbf{t})'(c_1\mathbf{t})]\!]$ using (23) and the scalar $-c_1\gamma$ as

$$[\![-(c_1\gamma)(c_1\mathbf{t})'(c_1\mathbf{t})]\!] = [\![(c_1\mathbf{t})'(c_1\mathbf{t})]\!]^{-c_1\gamma}. \quad (24)$$

After obtaining all the terms, the server can compute the whole term $c_1^3c_{5,s}+c_1^3c_{6,s}+(2c_1^2\gamma\mathbf{x}_s)'(c_1\mathbf{t})-(c_1\gamma)(c_1\mathbf{t})'(c_1\mathbf{t})$ in (21) in the encrypted-domain using the homomorphic properties as

$$\begin{aligned}&[\![c_1^3c_{5,s} + c_1^3c_{6,s} + (2c_1^2\gamma\mathbf{x}_s)'(c_1\mathbf{t}) - (c_1\gamma)(c_1\mathbf{t})'(c_1\mathbf{t})]\!] \\ = \ &[\![c_1^3c_{5,s} + c_1^3c_{6,s}]\!].[\![(2c_1^2\gamma\mathbf{x}_s)'(c_1\mathbf{t})]\!].[\![-(c_1\gamma)(c_1\mathbf{t})'(c_1\mathbf{t})]\!]. \quad (25)\end{aligned}$$

3) Exponentiation using secure two-party computations: The only part left is exponentiation of $c_1^3c_{5,s} + c_1^3c_{6,s} + (2c_1^2\gamma\mathbf{x}_s)'(c_1\mathbf{t}) - (c_1\gamma)(c_1\mathbf{t})'(c_1\mathbf{t})$ to obtain (21). Since (25) is in the encrypted-domain, the server cannot do the exponentiation as in (21) and thus will interact with the clinician to complete this exponentiation. The server sends (25) to the clinician, who decrypts and obtains $c_1^3c_{5,s} + c_1^3c_{6,s} + (2c_1^2\gamma\mathbf{x}_s)'(c_1\mathbf{t}) - (c_1\gamma)(c_1\mathbf{t})'(c_1\mathbf{t})$. Then the clinician divides the decrypted component by c_1^3 and obtains $c_{5,s} + c_{6,s} + (2\gamma\mathbf{x}_s)'(\mathbf{t}) - (\gamma)(\mathbf{t})'(\mathbf{t})$. It is worth noting that the values $c_{5,s}$ and $c_{6,s}$ have been used as a scaling factor and masking factor, respectively to protect $(2\gamma\mathbf{x}_s)'(\mathbf{t})-(\gamma)(\mathbf{t})'(\mathbf{t})$. Note that $c_{5,s}+c_{6,s} < c_3$ and the range of $c_{6,s}$ must be the same as the range of $\mathbf{x}_i'\mathbf{x}_j, \forall\, i, j$. Since the range $c_{6,s}$ is the same as the range of $\mathbf{x}_s'\mathbf{t}$

(i.e. $\mathbf{x}'_s\mathbf{x}_s$) the clinician cannot extract any useful information from the decrypted $c_{5,s} + c_{6,s} + (2\gamma\mathbf{x}_s)'(\mathbf{t}) - (\gamma)(\mathbf{t})'(\mathbf{t})$. Note that for every component of (13) a fresh random value $c_{6,s}$ must be generated.

Now the clinician computes and encrypts $e^{c_{5,s}+c_{6,s}+(2\gamma\mathbf{x}_s)'(\mathbf{t})-(\gamma)(\mathbf{t})'(\mathbf{t})}$ and returns it to the server. The server has received (13) in the encrypted-domain (i.e. $[\![e^{c_{5,s}+c_{6,s}+(2\gamma\mathbf{x}_s)'(\mathbf{t})-(\gamma)(\mathbf{t})'(\mathbf{t})}]\!]$, $s \in S$), so it can compute (15) in the encrypted-domain as

$$\begin{aligned}[\![d(\mathbf{t})]\!] &= [\![\sum_{s\in S} d_{1,s}d_{2,s} + d_3]\!] = [\![d_3]\!][\![\sum_{s\in S} d_{1,s}d_{2,s}]\!], \\ &= [\![d_3]\!]\prod_{s\in S}[\![d_{1,s}d_{2,s}]\!] = [\![d_3]\!]\prod_{s\in S}[\![d_{2,s}]\!]^{d_{1,s}}. \quad (26)\end{aligned}$$

In order to complete the classification, the server needs to compute (16). Since $d(\mathbf{t})$ is in the encrypted-domain (i.e. (26)) the server needs to obtain the sign of an encrypted-number to complete this.

4) Obtaining the sign of an encrypted value: Let us denote two strings yes and no to represent the decision. Assume that if the sign of $d(\mathbf{t})$ is positive then the decision for the given patient data is yes and if the sign of $d(\mathbf{t})$ is negative then the decision for the given patient data is no. Since the server has the value of $d(\mathbf{t})$ in the encrypted-domain as in (26), we show in this section how to obtain the decision for the patient data in the encrypted-domain. Let us assume that $\mid d(\mathbf{t}) \mid < 10^l$, $l \in \mathbb{Z}$ in the plain-domain. Note that since the training and test data samples are normalized, the value of l can be determined using the scale factor $c_2e^{c_3}$ used in (8).

Now the server computes a new variable in the encrypted-domain as

$$[\![z]\!] = [\![10^l + d(\mathbf{t})]\!] = [\![10^l]\!].[\![d(\mathbf{t})]\!]. \quad (27)$$

Since $|d(\mathbf{t})| < 10^l$, the most significant digit of z is either 1 (i.e. if $d(\mathbf{t}) > 0$) or 0 (i.e. if $d(\mathbf{t}) < 0$). Let us denote the most significant digit of z as $\tilde{z} \in \{1, 0\}$. Hence, the decision Dec, can be obtained as

$$Dec = \tilde{z}.(yes - no) + no. \quad (28)$$

The most significant digit $\tilde{z}$ could be computed using the following linear operation:

$$\tilde{z} = 10^{-l}.\left[z - (z \mod 10^l)\right], \quad (29)$$

where subtraction sets the least significant digits of z to 0 while the multiplication shifts the most significant digit down. Since the z in (27) is in the encrypted-domain the server needs to obtain the $\tilde{z}$ in (29) in the encrypted-domain. This can be performed as follows:

$$\begin{aligned}[\![\tilde{z}]\!] &= [\![10^{-l}.\left[z - (z \mod 10^l)\right]]\!], \\ &= \left([\![z]\!].[\![z \mod 10^l]\!]^{-1}\right)^{10^{-l}}. \quad (30)\end{aligned}$$

However the z available at the server is encrypted, thus similar to the process leading to the server being able to compute (26), the server engages the clinician in a secure two-party computation protocol to compute $[\![z \mod 10^l]\!]$.

The server blinds $[\![z]\!]$ using a uniformly distributed random value r as

$$[\![z_r]\!] = [\![z + r]\!] = [\![z]\!].[\![r]\!],$$

and this is sent to the clinician who decrypts $[\![z_r]\!]$ and reduces $z_r \bmod 10^l$. The result i.e., $[\![z_r \bmod 10^l]\!]$ is then encrypted and returned to the server who retrieves $[\![z \bmod 10^l]\!]$ as

$$[\![z \bmod 10^l]\!] = [\![z + r \bmod 10^l]\!].[\![r \bmod 10^l]\!]^{-1}[\![\lambda]\!]^{10^l},$$

where $\lambda \in \{0, 1\}$ is used to avoid underflow (i.e. $\lambda = 0$ if $z + r \bmod 10^l > r \bmod 10^l$ or $\lambda = 1$ if $z + r \bmod 10^l < r \bmod 10^l$). The server knows $z + r \bmod 10^l$ in the plain-domain while the clinician knows $r \bmod 10^l$ in the plain-domain. Comparing two integers in encrypted-domain has been widely studied in the literature [26]. Now the server can compute $[\![\tilde{z}]\!]$ using (30). The obtained $[\![\tilde{z}]\!]$ can be used in (28) to obtain the decision for the patient data in the encrypted-domain, as follows:

$$[\![Dec]\!] = [\![\tilde{z}.(yes - no) + no]\!] = [\![\tilde{z}]\!]^{(yes-no)}.[\![no]\!]. \quad (31)$$

Now $[\![Dec]\!]$ can be returned to the clinician who decrypts it to find the decision of the patient data (see Fig. 4).

C. Information Leakage

In the proposed algorithm, the private key resides at clinician side, hence it is not possible for the remote server who participates in this classification operation to decrypt the test sample or the classification result. However, the remote server interacts with the clinician when the homomorphic properties of Paillier cryptography are not sufficient to complete the task. During the interaction any encrypted values sent by the server could be decrypted by the clinician. It is possible to formally analyse whether this interaction can reveal any server side parameters to the clinician. The server first interacts with the clinician to compute $[\![(c_1\mathbf{t})'(c_1\mathbf{t})]\!]$ from $[\![(c_1\mathbf{t})]\!]$ (see between (22) and (23)). If the server sends $[\![(c_1\mathbf{t})]\!]$ without any preprocessing then it may possible for the clinician to infer the normalization parameters using (17). However, the sever sends only $[\![\hat{\mathbf{t}}]\!] = [\![c_1\mathbf{t} + \mathbf{r}]\!]$, where the addition of random variables $\mathbf{r} = [r_1, \ldots, r_n]'$ makes it infeasible for the clinician to extract any information about the normalization parameters from $\hat{\mathbf{t}}$. Secondly, the server interacts with the clinician to exponentiate the encrypted value as described in Section III-B-3 where the server adds a random value $c_{6,s}$ in $c_{5,s} + c_{6,s} + (2\gamma\mathbf{x}_s)'(\mathbf{t}) - (\gamma)(\mathbf{t})'(\mathbf{t})$ where the range of $c_{5,s}$ is the same as the range of $(\gamma\mathbf{x}_s)'(\mathbf{x}_s)$. Note that for every support vector, $c_{6,s}$ is generated freshly. Hence, this randomization makes it infeasible for the clinician to extract any server side parameters. However, this interaction reveals the number of support vectors used for classification. Since there is no relation between size of the data set and the number of support vectors used for classification, this leakage is not a breach to privacy of the data set used in the training phase. Finally, the server interacts with the clinician for modulo reduction in order to obtain the sign of the decision function $d(\mathbf{t})$ (see between (27) and (31)). Since, $d(\mathbf{t})$ is included in z in (27), revealing z may leak the decision function value to clinician. Hence, the server adds a random value r to z

TABLE II

SOME EXAMPLES OF NORMALIZED TRAINING SAMPLES OF THE WBC AND PID DATA SETS. THE FIRST FOUR SAMPLES ARE BENIGN WHILE THE LAST FOUR SAMPLES ARE MALIGNANT.

	Fea. 1	Fea. 2	Fea. 3	Fea. 4	Fea. 5	Fea. 6	Fea. 7	Fea. 8	Fea. 9
Sample 1 [WBC]	-0.1243	0.1970	-0.6986	-0.7383	-0.6366	-0.5541	-0.6966	-0.1754	-0.6101
Sample 2 [WBC]	-0.1196	0.1970	0.2823	0.2666	0.7585	1.6919	1.7700	-0.1754	-0.2827
Sample 3 [PID]	-0.8443	-1.1227	-0.1551	0.5306	-0.6944	-0.6745	-0.3681	-0.1902	-
Sample 4 [PID]	-0.8443	-0.9976	-0.1551	0.1544	0.1195	-0.4858	-0.9209	-1.0412	-
Sample 5 [WBC]	-0.0967	1.2590	2.2442	2.2764	1.8048	1.6919	1.7700	2.2964	1.3543
Sample 6 [WBC]	-0.0570	0.1970	-0.0447	-0.0684	0.0609	-0.5541	-0.1485	0.2365	0.3721
Sample 7 [PID]	0.6395	0.8478	0.1524	0.9067	-0.6944	0.2057	0.4612	1.4266	-
Sample 8 [PID]	1.2331	1.9425	-0.2576	-1.2874	-0.6944	-1.0894	0.5964	-0.1051	-

TABLE III

THE CLASSIFICATION RESULTS FOR THE WBC DATA SET FOR DIFFERENT VALUES OF γ IN THE PLAIN-DOMAIN.

WBC	$\gamma = 0.1$	$\gamma = 0.2$	$\gamma = 0.3$	$\gamma = 0.5$	$\gamma = 1$	$\gamma = 10$	$\gamma = 20$	$\gamma = 50$
Benign (444)	436 (98.20%)	435 (97.97%)	435 (97.97%)	435 (97.97%)	435 (97.97%)	433 (97.52%)	433 (97.52%)	433 (97.52%)
Malignant (237)	212 (89.45%)	212 (89.45%)	215 (90.72%)	215 (90.72%)	215 (90.72%)	229 (96.62%)	228 (90.20%)	228 (90.20%)
Overall Accuracy (681)	648 (95.15%)	647 (95.01%)	650 (95.45%)	650 (95.45%)	650 (95.45%)	662 (97.21%)	661 (97.06%)	661 (97.06%)

TABLE IV

THE CLASSIFICATION RESULTS FOR THE PID DATA SET FOR DIFFERENT VALUES OF γ IN THE PLAIN-DOMAIN.

PID	$\gamma = 0.1$	$\gamma = 1$	$\gamma = 5$	$\gamma = 10$	$\gamma = 15$	$\gamma = 20$	$\gamma = 25$	$\gamma = 30$
Benign (500)	466 (93.20%)	421 (84.20%)	442 (88.40%)	462 (92.40%)	482 (96.40%)	473 (94.60%)	444 (88.80%)	429 (85.80%)
Malignant (268)	106 (39.55%)	166 (61.94%)	204 (76.11%)	230 (85.82%)	239 (89.17%)	227 (84.70%)	230 (85.82%)	218 (81.34%)
Overall Accuracy (768)	572 (74.48%)	587 (76.43%)	646 (84.11%)	692 (90.10%)	721 (93.88%)	700 (91.15%)	674 (87.76%)	647 (84.24%)

before sending it to the clinician. Again this randomization makes it infeasible for the clinician to extract any server side information. Overall our proposed method not only preserves the privacy of the patient information but also the server side classification parameters.

IV. PERFORMANCE ANALYSIS

In this section we analyse the performance of the proposed encrypted-domain algorithm. We compare the accuracy of the proposed encrypted-domain method with the conventional plain-domain method. For the experiment, we consider two data sets from the UCI machine learning repository called the Wisconsin Breast Cancer (WBC) and Puma Indian Diabetic (PID) data sets [27]. The WBC data set contains 681 samples where 444 samples are benign (non-cancerous) and 237 samples are malignant (cancerous) while PID data set contains 768 samples where 500 samples are malignant and 268 samples are benign. The number of features for each sample in WBC and PID data sets are nine and eight, respectively (excluding class label attribute). Table II shows some examples of training samples after normalization from the WBC and PID data sets.

For evaluation, we used a leave-one-out approach [28], that is, one sample is removed from the data set and all the remaining samples are used for training the SVM. The removed sample will be used as patient data. This procedure will be repeated for a different left out sample each time until all the samples are used. In order to analyse the proposed methods we first conduct an experiment in the plain-domain. Later we do the same experiment in the encrypted-domain for various scaling factors.

A. Experiments in the Plain-Domain

In all experiments, we assume that the training data are not linearly separable and therefore we use a Gaussian kernel method as in (6). Initially, we need to determine empirically an appropriate value for γ in (6). Hence, we have obtained Table III and Table IV for the WBC and PID data sets, respectively in the plain-domain using the method described in Section II. These tables show the classification accuracy for various γ values. Let us explain the sixth result column (i.e. $\gamma = 10$) in Table III. When $\gamma = 10$, the total number of correctly classified benign samples is 433 out of 444 (97.52%) and that of malignant samples is 229 out of 237 (90.72%). In total 662 samples were correctly classified out of 681 (97.21%). Similarly, when $\gamma = 15$ for PID data set as in Table IV, the total number of correctly classified benign samples is 482 out

of 500 (96.40%) and that of malignant samples is 239 out of 268 (89.17%). In total 721 samples were correctly classified out of 768 (93.88%). Since $\gamma = 10$ for WBC data set and $\gamma = 15$ for PID data set provide higher accuracy than other values in this experiment, without loss of generality, we use $\gamma = 10$ for WBC data set and $\gamma = 15$ for PID data set for the experiments in the encrypted-domain. We also noticed that the average numbers of support vectors used for WBC and PID data sets are 205 and 535, respectively.

B. Experiments in the Encrypted-Domain

We have now evaluated our algorithm with 2048-bit key size. We tested our proposed privacy-preserving algorithm in a computer with 3.40 GHz processor and 8 GB of RAM running on Windows 64-bit operating system. The algorithm is written in C++ using GNU GMP library version 4.2.4. Both the server and clinician were modeled as different threads of a single program, which passes variables to each other.

As we mentioned in Section III-B, the scaling factors c_1, c_2, c_3 and $c_{5,s}$ have influence on the classification accuracy in the encrypted-domain due to the fact that the Paillier cryptosystem only encrypts integers. When we set $c_1 = 1$, $c_2 = 1$, $c_3 = 0$ and $c_{5,s} = 0$, the classification accuracy has reduced to 0%, which shows the importance of the scaling factor in the encrypted-domain.

Scalar c_1 is a linear scalar and has been used to scale the patient data in (20). We noticed that each element of the normalized training samples of the WBC and PID data sets are in the range of $\pm 10^{-4}$ (see Table II), hence, we have chosen $c_1 = 10^4$. Scalar c_2 is also a linear scalar and it has been used in (12) and (14). In order to get six decimal point accuracy, we have chosen $c_2 = 10^6$ in all the experiments. Scalar c_3 is an exponential scalar and used in (12) and (14). The Paillier cryptosystem only encrypts integers in $\mathbb{Z}_n$, hence, $0 < c_3 < \log_e(n)$. The scalar $c_{5,s}$ must be chosen such that $c_{5,s} + c_{6,s} < c_3$, where $c_{6,s}$ is a masking factor in the range of $\mathbf{x}'_i\mathbf{x}_j \ \forall i, j$. Next we obtain a classification accuracy for different values of scaling factors.

Table V shows the accuracies in the encrypted-domain for different values of $c_{5,s}$ when $c_3 = 10$. The scalar $c_{5,s}$ can take any value between 0 and c_3 and is not necessarily an integer. For the WBC data set, the classification accuracy in the encrypted-domain is equal to the classification accuracy in the plain-domain when $5 \leq c_{5,s} \leq 9$ (i.e. $\gamma = 10$ column in Table III). Similarly, for the PID data set, the classification accuracy in the encrypted-domain is equal to the classification accuracy in the plain-domain when $7 \leq c_{5,s} \leq 9$ (i.e. $\gamma = 15$ column in Table IV). However, the performance of the encrypted-domain algorithm deteriorates when $c_{5,s} < 3$ ($c_{5,s} < 5$) for WBC data set (PID data set) due to the quantization effect of the Paillier encryption. Hence, the decision function becomes independent of the patient data and provides infeasible results. Overall, the proposed method does not degrade the classification accuracy even when the classification is conducted in the encrypted-domain for appropriate scaling factors.

TABLE V
CLASSIFICATION RESULTS FOR THE WBC AND PID DATA SETS IN THE ENCRYPTED-DOMAIN WHEN $c_3 = 10$.

WBC ($\gamma = 10$)	$c_{5,s} = 0$	$c_{5,s} = 2$	$c_{5,s} = 5$	$c_{5,s} = 9$
Benign (444)	N/A	N/A	433 (97.52%)	433 (97.52%)
Malignant (237)	4 (1.69%)	170 (71.73%)	229 (96.62%)	229 (96.62%)
Overall Accuracy (681)	4 (0.58%)	170 (24.92%)	662 (97.21%)	662 (97.21%)
PID ($\gamma = 15$)	$c_{5,s} = 0$	$c_{5,s} = 2$	$c_{5,s} = 5$	$c_{5,s} = 9$
Benign (500)	N/A	71 14.20%	452 90.40%	482 (96.4%)
Malignant (268)	3 (1.11%)	23 (8.58%)	239 (89.17%)	239 (89.17%)
Overall Accuracy (768)	3 (0.39%)	94 (12.23%)	691 (89.97%)	721 (93.88%)

C. Analysing the Factors Related to Accuracy

Equation (6) clearly shows that the classification function only depends on the number of support vectors (i.e. $|S|$) and the corresponding $\alpha_s \ \forall s$ and b. Hence, after the training phase, the classification task becomes independent of the size of the data set.

In the encrypted-domain classification equation, $d_{1,s} \ \forall s$ and d_3 can be calculated by the server in the plain-domain. Since the test sample, $\mathbf{t}$, given to the server is in the encrypted format, $d_{2,s} \ \forall s$ need to be computed by the server in the encrypted-domain by interacting with the clinician. Hence, let us closely look at $d_{2,s} = e^{c_{5,s}} e^{c_{6,s} - \gamma \mathbf{t}'\mathbf{t} + 2\gamma \mathbf{x}'_s \mathbf{t}} = e^{c_{5,s} + c_{6,s} - \gamma \mathbf{t}'\mathbf{t} + 2\gamma \mathbf{x}'_s \mathbf{t}}$, where the server computes $c_{5,s} + c_{6,s} - \gamma \mathbf{t}'\mathbf{t} + 2\gamma \mathbf{x}'_s \mathbf{t}$ in the encrypted-domain. Since the Paillier cryptography approximates the values to integers (floor values) before encryption, it is crucial to scale the values in the test sample $\mathbf{t}$ and support vectors $\mathbf{x_s}$. Scaler c_1 has been used to scale the test sample and $c_{5,s}$ has been used to scale the $c_{6,s} - \gamma \mathbf{t}'\mathbf{t} + 2\gamma \mathbf{x}'_s \mathbf{t}$. Scaler c_2 has been used to avoid the approximation errors in $d_{1,s}$ and d_3. Hence, classification errors due to integer replacements during Paillier encryption are solely dependent on choices of these scalars in the encrypted-domain.

The choice of c_1 depends on values in the test sample and support vectors. Table II in the revised manuscript shows that these values are in the range of $\pm 10^{-4}$ and there was not significant improvement in performance when we used more than four decimal points. Hence, the error is dependent on how many decimal points would be enough to get a good accuracy. Which means if there is no significant increment in performance then using more decimal points in values will not be useful. Since there is no significant improvement in performance beyond four decimal points c_1 has chosen to be 10^4.

As shown in Section III-B-3, $e^{c_{5,s}}$ has been used to scale $e^{c_{6,s} - \gamma \mathbf{t}'\mathbf{t} + 2\gamma \mathbf{x}'_s \mathbf{t}}$. Hence, the classification error is dependent on the value $e^{c_{5,s}}$ too. Table VI depicts the required $c_{5,s}$ for various decimal value accuracy (i.e. $e^{2.3026} = 10^1$).

TABLE VI

CONVERSION TABLE FOR $c_{5,s}$.

$c_{5,s}$	2.3026	4.6052	6.9078	9.2103	11.5129
Accuracy	10^1	10^2	10^3	10^4	10^5

In our experiment, there is no improvement in performance beyond three decimal point accuracy for $c_{5,s}$ (i.e. $c_{5,s} > 6.9$). Similarly, c_2 has been fixed to 10^6. Overall there will not be significant improvement in accuracy even if we increase these scalers beyond the values mentioned earlier. However, from equations (6) and (8), it is obvious that the accumulated error of the decision function (i.e. $d(\mathbf{t})$) due to the wrong choice of c_1 (i.e. if $c_1 = 10^2$) is proportional to the $number\ of\ support\ vectors \times e^{dimension\ of\ data}$ and the accumulated error in the decision function due to the wrong choice of c_2 and $c_{5,s}$ being proportional to the $number\ of\ support\ vectors$.

D. Communication Complexity

The communication cost of the proposed algorithm highly depends on the size of Paillier cryptography; in our implementation the size of a encrypted sample is 2048 bits long. Sending an encrypted test sample with N number of features consumes $2.048N\ Kbits$ of bandwidth in the communication channel. In the proposed algorithm, the server interacts with the clinician for three times. During the second interaction (i.e. for exponentiation) the server sends $|S|$ number of encrypted values (i.e. equal to the total number of support vectors) while in the first and last interaction the server sends only one value. Hence, the communication cost for our algorithm is upper-bounded by the second interaction which requires $2.048|S|\ Kbits$ of bandwidth. Since the number of support vectors should be less than the size of the data set, the worse case bandwidth requirement for both WBC and PID data sets are $0.174MB$ and $0.197MB$, respectively.

E. Computation Complexity

We measure the computation complexity in terms of average runtime required for the proposed algorithm when the size of the security parameter $N = 2048$. The average times required for WBC and PID data sets are 41 and 92 seconds respectively. It is noted that the average time is increasing linearly, with the number of support vectors used for classification.

V. RELATED WORK

In general, data classification is a combination of two phases: training phase and testing phase. The first phase, training a classifier, requires a large collection of data. There are various organizations publish their customers data for research and monetary purposes. Publishing a person specific data set (e.g. data related to patients of a cancer hospital) may reveal individuals identity and breach the privacy of patients. However, there are various privacy preserving techniques (i.e. anonymization techniques and data perturbation techniques) have been well studied in literature to preserve the privacy of individuals in the data sets ([29]–[31] and references there in). However, the proposed work in this paper considers the privacy in the second phase of the data classification task, where clinicians only require to send the test data of their patient to the remote server where classifier is already established. Since the proposed method preserve the privacy of training data set, it is possible for any organization with large data to provide a classification as a service to anybody through the Internet rather than anonymize and publish the data set in plain-domain. Hence, our method is different from the data anonymization and data perturbation based methods.

A SVM has been used in bio medical engineering to diagnose various diseases in the plain-domain ([9]–[11] and references therein). Note that, any algorithm in the plain-domain cannot be used to provide decision support via the Internet due to the privacy issue. Recently, Mathew and Obradovic proposed a privacy preserving framework for clinical decision support using a decision tree based machine learning technique [32]. The work in [32] supports prevention of personally identifiable information leakage. However, the authors in [32] only considered the privacy of the training data set by assuming that the training data is available from more than one location. In our work, we are not only preserving the privacy of the training data set but also the patient data and the result. Moreover, the algorithm developed for the decision tree cannot be directly extended to a SVM.

Let us review some of the privacy-preserving SVM algorithms developed in the data mining literature. The majority of works in datamining were developed for the distributed setting [33]–[36]. More specifically, in [33]–[36], the researchers assumed that different parties hold parts of the training data sets. Hence, they developed protocols to securely train a common classifier without each party needing to disclose its own training data to other parties. After the training each party holds part of the classification parameters and support vectors. In order to classify a new data, each party has to be involved equally to compute part of the kernel matrix and then all parties together or the trusted third party will classify the new data. The works in [34]–[36] exploited the secure multi-party integer summation to cooperatively compute the kernel matrix. Basically, each party generates the Gramm matrix using scalar products of the training and new data samples. This Gramm matrix is later revealed to the trusted third party who will compute the kernel matrix and then classify the new data. Revealing the Gramm matrix may leak the private data and therefore privacy cannot be entirely preserved.

The work in [33] proposed for the first time a strongly privacy-enhanced protocol for a polynomial kernel based SVM using cryptographic primitives where the authors assumed that the training data are distributed. Hence, to preserve privacy, they developed a protocol to perform secure kernel sharing, prediction and training using secret sharing and homomorphic encryption techniques. At the end of the training each party will hold a share of the secret. In the testing phase all parties collaboratively perform the classification using their shared secrets. At the end of the protocol each party will hold a share of the predicted class label. Since the work is based on secret sharing, all the parties must be involved in every operation of calculating the kernel values and predicting the

class. Hence, it is suitable only for the distributed scenario and not for the client-server model considered in this paper. In the client-server model, the client just sends the new data in the encrypted-domain and is minimally involved in interactions with the server during the classification process. Moreover, the method developed in [33] considered only the polynomial kernel and so it cannot be modified directly to work with the Gaussian kernel based SVM considered in this paper as these kernel functions are of different forms.

The recent work in [37] discusses the issue of releasing the trained SVM classifier without violating the privacy of support vectors. While the Gaussian kernel was considered, a Taylor series was exploited to approximate the infinite dimension of the Gaussian kernel into finite dimension and adhere negligible performance loss. Since this works purely in the plain-domain, it cannot be modified to the clinician-server scenario considered in this paper.

VI. Conclusions

In this paper we have proposed a privacy-preserving decision support system using a Gaussian kernel based support vector machine. Since the proposed algorithm is a potential application of emerging outsourcing techniques such as cloud computing technology, rich clinical data sets (or healthcare knowledge) available in remote locations could be used by any clinicians via the Internet without compromising privacy, thereby enhancing the decision making ability of healthcare professionals. We have exploited the homomorphic properties of the Paillier cryptosystem within our algorithm where the cryptosystem only encrypts integer values. Hence, we proposed a novel technique to scale the continuous variables involved in the process without compromising the performance and privacy. To validate the performance, we have evaluated our method on two medical data sets and the results showed that the accuracy is up to 97.21%. Importantly, the benefit of our encrypted-domain method is that patient data need not be revealed to the remote server as they can remain in encrypted form at all times, even during the diagnosis process.

References

[1] Garg, A. X., Adhikari, N. J., McDonald, H., et al. Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: A systematic review. In: The Journal of the American Medical Association (JAMA), vol. 293, no. 10, pp. 1223–1238. (2005)

[2] Carson, E. R., Cramp, D. G., Morgan, A., and Roudsari, A. V.: Clinical decision support, systems methodology, and telemedicine: their role in the management of chronic disease. IEEE Trans. Information Technology in Biomedicine, vol.2, no.2, pp.80–88. (Jun. 1998)

[3] Lisboa, P. J., and Taktak, A. F. G.: The use of artificial neural networks in decision support in cancer: a systematic review. In: Neural networks, vol. 19, pp. 408–415. (2006)

[4] Baskaran, V., Guergachi, A., Bali, R. K., and Naguib, R. N. G.: Predicting breast screening attendance using machine learning techniques. In: IEEE Trans. Information Technology in Biomedicine, vol. 15, no. 2, pp. 251–259. (2011)

[5] Shin, H., and Markey, M. K.: A machine learning perspective on the development of clinical decision support systems utilizing mass spectra of blood samples. In Journal of Biomedical Informatics, vol. 39, no. 2, pp. 227–248. (2006)

[6] Sundareswaran, S., Squicciarini, A. C., and Lin, D.: Ensuring distributed accountability for data sharing in the cloud. In: IEEE Trans. Dependable and Secure Computing, vol. 9, no. 4, pp. 555–567. (Jul.–Aug. 2012)

[7] Pearson, S., and Charlesworth, A.: Accountability as a way forward for privacy protection in the cloud. In: Proc. 1st Int'l Conf. Cloud Computing (CloudCom), pp. 131–144, Beijing, China. (2009)

[8] Pearson, S., Shen, Y., and Mowbray, M.: A privacy manager for cloud computing. In: Proc. Int'l Conf. Cloud Computing (CloudCom), pp. 90–106, Beijing, China. (2009)

[9] Barakat, N., Bradley, A. P., and Barakat, M. N. H.: Intelligible support vector machines for diagnosis of diabetes mellitus. In: IEEE Trans. Information Technology in Biomedicine, vol. 14, no. 4, pp. 1114–1120. (2010)

[10] Ajemba, P. O., Ramirez, L., Durdle, N. G., Hill, D. L., and Raso, V. J.: A support vectors classifier approach to predicting the risk of progression of adolescent idiopathic scoliosis. In: IEEE Trans. Information Technology in Biomedicine, vol. 9, no. 2, pp. 276–282. (2005)

[11] Guler, I., and Ubeyli, E. D.: Multiclass support vector machines for EEG-signals classification. In: IEEE Trans. Information Technology in Biomedicine, vol. 11, no. 2, pp. 117–126. (2007)

[12] Hsu, C-W., Chang, C-C., and Lin, C-J.: A Practical Guide to Support Vector Classification. Department of Computer Science National Taiwan University, Taiwan. (2010)

[13] Frank, A., and Asuncion, A.: UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science. (2010).

[14] Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Proc. 17th Int'l Conf. Theory and Application of Cryptographic Techniques, pp. 223–238, Prague, Czech Republic. (1999)

[15] Goldreich, O.: Secure Multi-Party Computation. Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, Isareal. (1998)

[16] Cortes, C., and Vapnik, V.: Support-vector networks. In: Machine Learning, vol. 20, no. 3, pp. 273–297. (1995)

[17] Vapnik, V.: An overview of statistical learning theory. In: IEEE Trans. Neural Networks, vol. 10, pp. 988–999. (1999)

[18] Tong, S., and Koller, D.: Support vector machine active learning with applications to text classification. In: Journal of Machine Learning Research, Vol. 2, pp. 45–66. (2002)

[19] Kotsia, I., and Pitas, I.: Facial expression recognition in image sequences using geometric deformation features and support vector machines. In: IEEE Trans. Image Process., vol. 16, no. 1, pp. 172–187. (2007)

[20] Bergsma, S., Lin, D., and Schuurmans, D.: Improved natural language learning via variance-regularization support vector machines. In: Proc. 14th Conf. Computational Natural Language Learning, pp. 172–181, Stroudsburg, PA, USA. (2010)

[21] Ben-Hur, A., Ong, C., S., Sonnenburg, S., Scholkopf, B., and Ratsch, G.: Support vector machines and kernels for computational biology. In: PLoS Computational Biology — www.ploscompbiol.org, vol. 4, no. 10, pp. 1–10. (2008)

[22] Boyd, S. and Vandenberghe, L.: Convex Optimization. Cambridge, U.K.: Cambridge Univ. Press. (2004)

[23] Vapnik, V.: The Nature of Statistical Learning Theory. In: Springer-Verlag New York, Inc., New York, NY, USA. (1995)

[24] Erkin, Z., Franz, M., Guajardo, J., Katzenbeisser, S., Lagendijk, I., and Toft. T.: Privacy-preserving face recognition. In: Proc. 9th Int'l Symposium on Privacy Enhancing Technologies, pp. 235–253, Seattle, WA. (2009)

[25] Rahulamathavan, Y., Phan, R.C.-W., Chambers, J.A., and Parish, D.J.: Facial Expression Recognition in the Encrypted Domain Based on Local Fisher Discriminant Analysis. IEEE Trans. Affective Computing, vol. 4, no. 1, pp. 83–92. (Jan.–Mar. 2013)

[26] Damgård, I., Geisler, M., and Krigaard, M.: Efficient and secure comparison for online auctions. In: Proc. 12th Australasian Conf. Information Security and Privacy, pp. 416-430, Townsville, Australia. (2007)

[27] Mangasarian, O. L., Street, W. N., and Wolberg, W. H.: Breast cancer diagnosis and prognosis via linear programming. In Operations Research, vol. 43, pp. 570–577. (Jul.-Aug. 1995)

[28] Cawley, G. C., and Talbot, N. L. C.: Efficient leave-one-out cross-validation of kernel fisher discriminant classifiers. In: Pattern Recognition, vol. 36, no. 11, pp. 2585–2592. (2003)

[29] Xiao-Bai Li, and Sarkar, S.: A Tree-Based Data Perturbation Approach for Privacy-Preserving Data Mining. In IEEE Trans. Knowledge and Data Engineering, vol. 18, no. 9, pp. 1278–1283. (Sep. 2006)

[30] Benjamin C. M. Fung, Ke Wang, and Philip S. Yu: Anonymizing Classification Data for Privacy Preservation. In IEEE Trans. Knowledge and Data Engineering, vol. 19, no. 5, pp. 711–725. (May 2007)

[31] Mingxuan Yuan, Lei Chen, Yu, P.S., and Ting Yu: Protecting Sensitive Labels in Social Network Data Anonymization. In IEEE Trans. Knowledge and Data Engineering, vol. 25, no. 3, pp. 633–647. (Mar. 2013)

[32] Mathew, G., and Obradovic, Z.: A privacy-preserving framework for distributed clinical decision support. In proc. IEEE 1st Int'l Conf. Computational Advances in Bio and Medical Sciences, pp.129–134. (2011)
[33] Lipmaa, H., Laur, S., and Mielikainen, T.: Cryptographically private support vector machines. In: Proc. 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 618–624. ACM Press, Philadelphia, USA. (Aug. 2006)
[34] Yu, H., Jiang, X., and Vaidya. J.: Privacy-preserving support vector machine using nonlinear kernels on horizontally partitioned data. In: Proc. ACM Symposium on Applied Computing (SAC), pp. 603–610, Dijon, France. (2006)
[35] Yu, H., Vaidya, J., and Jiang, X.: Privacy-preserving SVM classification on vertically partitioned data. In: Proc. 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 647–656, Singapore. (2006)
[36] Chen, K., and Liu, L.: Privacy preserving data classification with rotation perturbation. In: Proc. 5th IEEE Int'l Conf. Data Mining, pp. 589–592, Washington, DC, USA. (2005)
[37] Lin, K-P., and Chen, M-S.: On the design and analysis of the privacy-preserving SVM classifier. In: IEEE Trans. Knowledge and Data Engineering, vol. 23, no. 11, pp. 1704–1717. (Nov. 2011)

Yogachandran Rahulamathavan received the B.Sc. degree (first-class honors) in electronic and telecommunication engineering from the University of Moratuwa, Sri Lanka, in 2008 and a Ph.D. degree in Signal Processing from Loughborough University, UK in 2011.

From April 2008 to September 2008, he was an Engineer at Sri Lanka Telecom, Sri Lanka and from November 2011 to March 2012, he was a Research Assistant with the Advanced Signal Processing Group, School of Electronic, Electrical and Systems Engineering, Loughborough University, UK. He is currently working as a Research Fellow with the Information Security Group, School of Engineering and Mathematical Sciences, City University London, UK.

Dr Rahulamathavn received a scholarship from Loughborough University to pursue his Ph.D. degree. His research interests include signal processing, machine learning and information security and privacy.

Dr. Suresh Veluru is a Research Fellow in the School of Engineering and Mathematical Sciences at City University London, United Kingdom. Prior to this, he worked as a research fellow in Computer Science at University of York, United Kingdom and University of New Brunswick, Canada. He received his PhD in Computer Science from Indian Institute of Technology Guwahati, India in 2009. His research interests include pattern recognition, data mining, natural language processing, artificial intelligence and privacy preserving data mining.

Raphael Phan obtained his Ph.D (Eng) in security from Multimedia University. Prior to joining Loughborough University, he was Director of the Information Security Research (iSECURES) Laboratory at Swinburne University of Technology from 2004 to 2007; and a senior researcher in the Security & Cryptography Lab (LASEC) at the Ecole Polytechnique Fdrale de Lausanne (EPFL), Switzerland between 2007 and 2008.

He is in the Editorial Board of Cryptologia, and the Cryptology & Information Security series of IOS Press. He is General Chair of Mycrypt '05 and Asiacrypt '07, Program Chair of ISH '05, and serves in technical Program Committees of international conferences since 2005. Raphael is co-designer of BLAKE, one of the five hash functions in the final of the NIST SHA-3 competition. His research interests include diverse areas of security and privacy, recently in particular in privacy preservation and processing of data in the encrypted domain.

Jonathon A. Chambers (S'83-M'90-SM'98-F'11) received the Ph.D. degree in signal processing from the Imperial College of Science, Technology and Medicine, Imperial College London, London, U.K., in 1990.

From 1991 to 1994, he was a Research Scientist with Schlumberger Cambridge Research Center, Cambridge, U.K. In 1994, he returned to Imperial College London, as a Lecturer in signal processing and was promoted as a Reader (Associate Professor) in 1998. From 2001 to 2004, he was the Director of the Centre for Digital Signal Processing and a Professor of signal processing with the Division of Engineering, King's College London, London. From 2004 to 2007, he was a Cardiff Professorial Research Fellow with the School of Engineering, Cardiff University, Wales, U.K. In 2007, he joined the Department of Electronic and Electrical Engineering, Loughborough University, Loughborough, U.K., where he currently heads the Advanced Signal Processing Group and serves as the Associate Dean Research with the School of Electronic, Electrical and Systems Engineering. He is a co-author of the books Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability (Wiley, 2001) and EEG Signal Processing (Wiley, 2007). He has advised more than 50 researchers through to Ph.D. graduation and published more than 400 conference proceedings and journal articles, many of which are in IEEE journals. His research interests include adaptive and blind signal processing and their applications.

Dr. Chambers is a Fellow of the Royal Academy of Engineering, U.K., and the Institution of Electrical Engineers. He was the Technical Program Chair of the 15th International Conference on Digital Signal Processing (2007) and the 2009 IEEE Workshop on Statistical Signal Processing, both held in Cardiff, U.K., and a Technical Program Cochair for the 36th IEEE International Conference on Acoustics, Speech, and Signal Processing (2011), Prague, Czech Republic. He is the recipient of the first QinetiQ Visiting Fellowship in 2007 "for his outstanding contributions to adaptive signal processing and his contributions to QinetiQ" as a result of his successful industrial collaboration with the international defense systems company QinetiQ. He has served on the IEEE Signal Processing Theory and Methods Technical Committee for six years and is currently a member of the IEEE Signal Processing Society Awards Board and the European Signal Processing Society Best Paper Awards Selection Panel. He has also served as an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING for three terms over the periods 1997-1999, 2004-2007, and 2011- (and is currently an Area Editor).