A Privacy-Preserving Gaussian Process Regression Framework Against Membership Inference Attacks Using Random Unitary Transformation

Md. Rashedul Islam[1], Jannatul Ferdous Akhi[2], Takayuki Nakachi[3]
Graduate School of Engineering and Science, University of the Ryukyus, Okinawa, Japan[1,2]
Information Technology Center, University of the Ryukyus, Okinawa, Japan[3]

***Abstract*—As artificial intelligence (AI) systems become increasingly embedded in sensitive domains such as healthcare and finance, they face heightened vulnerabilities to privacy threats. A prominent type of attack against AI is the membership inference attack (MIA), which aims to determine whether specific data instances were used in a model's training set, thereby posing a serious risk of sensitive information disclosure. This study focuses on Gaussian Process (GP) models, which are widely adopted for their probabilistic interpretability and ability to quantify predictive uncertainty, and examines their susceptibility to MIAs. To mitigate this threat, a novel defense mechanism based on Random Unitary Transformation (RUT) is introduced, which encrypts training and testing inputs using orthonormal matrices. Unlike Differential Privacy-based Gaussian Processes (DP-GPR), which rely on noise injection and often degrade model performance, the proposed method preserves both the structural integrity and predictive fidelity of the GP model without injecting noise into the learning process. Two configurations are evaluated: i) encryption applied to both training and test data, and ii) encryption applied only to training data. Experimental results on a medical dataset demonstrate that the framework significantly reduces the effectiveness of MIAs while maintaining high predictive accuracy. Comparative analysis with DP-GPR models further confirms that the proposed method achieves competitive or stronger privacy protection with less impact on model utility. These findings underscore the potential of structure-preserving transformations as a practical and effective alternative to noise-based privacy mechanisms in GP models, particularly in privacy-critical machine learning applications.**

***Keywords*—Gaussian process; differential privacy; random unitary transformation; membership inference attack; machine learning**

I. Introduction

Machine learning (ML) is now widely used in many industries due to its popularity and efficacy. However, ethical and legal restrictions restrict the dissemination of private data in sectors such as healthcare and banking. Protecting data privacy is therefore essential at every stage of the ML lifecycle, from developing models to deployment [1]. The widespread use of sensitive data in training neural network models has raised concerns regarding privacy preservation. To assess whether a model inadvertently reveals information about its training data, membership inference attacks (MIAs) [2] have become a widely adopted evaluation technique [3], [4]. In such attacks, adversaries attempt to determine whether a data point was included in the training set by leveraging the model's behavior when presented with that sample. These attacks pose significant risks, especially when inclusion in the training data reveals sensitive information. For instance, if a model is trained using medical imaging data such as MRI scans, disclosing that a particular image was part of the training set could reveal confidential health details. Similarly, training on criminal offender databases may inadvertently expose an individual's criminal background. If an adversary knows a data instance, learning that it was used in training constitutes a breach of confidentiality. MIAs are widely recognized as indicators of privacy risks when ML models are externally accessible. Beyond their standalone impact, MIAs often serve as foundations for advanced attacks such as property inference, where adversaries aim to uncover global attributes of the training data, and profiling attacks, which attempt to infer sensitive characteristics about individuals represented in the data [5].

A widely recognized method in machine learning is the Gaussian Process (GP), which is based on Bayesian nonparametrics [6]. General practitioners can convert linear data into nonlinear formats by integrating domain expertise and knowledge into kernel functions. By fine-tuning the hyperparameters of these kernel functions using Bayes' theorem, it is possible to achieve exceptionally precise estimates. A GP can represent an unlimited number of units in one hidden layer of a neural network [7]. The GP provides both uncertainty (variance) and forecast mean values. By utilizing the uncertainty associated with the GP, it becomes straightforward to assess whether the test data is included in the training data.

A. Related Work

In [1], the authors proposed a privacy technique for GP membership inference to tackle this issue. Differential Privacy-based Gaussian Process (DP-GPR) has become the prevailing framework for ensuring privacy in machine learning [8]. By limiting the influence that any single data point can have on a model's output, DP-GPR provides strong privacy guarantees. This principle has been effectively incorporated into a wide range of machine learning models that assume data point independence, including applications in deep learning, Bayesian regression , and general Bayesian inference techniques such as Markov chain Monte Carlo and variational inference [9], [10], [11], [12]. While (DP-GPR) provides a worst-case guarantee for privacy protection, f-Membership Inference Privacy (f-MIP) leverages noisy stochastic gradient descent (SGD) as a model-agnostic approach to defend against inference attacks. The results show that f-MIP can protect privacy effectively

while using much less noise than DP-GPR, which helps the model keep better performance and accuracy. In contrast, the addition of excessive noise, as typically required by DP-GPR, can reduce model accuracy and disrupt the balance between predictions for training and non-training data. This imbalance may unintentionally create new patterns that adversaries can exploit to infer whether a given test sample was part of the training data. Both DP-GPR and f-MIP defend against inference attacks by injecting noise into the training process. While f-MIP typically uses less noise than DP-GPR and thereby achieves relatively better accuracy, it still introduces some degree of performance degradation due to noise perturbation.

Despite these advances, existing defenses share a fundamental limitation: they depend on noise-based mechanisms that inevitably weaken model utility. Although effective in theory, these approaches reduce accuracy, distort statistical structure, and require careful parameter tuning, making them less practical for real-world deployment. This leaves a clear research gap for defenses that can maintain predictive fidelity while still ensuring strong resistance to MIAs.

B. Contributions of this Study

To address this issue, we propose a secure computation-based defense against MIAs on GP [1]. Our approach uses a RUT to encrypt the data, enabling privacy-preserving processing. By preserving essential geometric properties such as norms, distances, and inner products, the proposed method avoids the trade-off between privacy and accuracy that characterizes noise-based techniques. As a result, attackers cannot determine whether specific test samples were part of the training dataset. Our method addresses the following two scenarios:

1) Case 1: Both the training and test data are encrypted. When the same private key is used for both, the model achieves prediction accuracy comparable to conventional non-encrypted GP regression. In contrast, if different private keys are used (for example, to simulate an attacker scenario), accurate predictions cannot be made, and it becomes infeasible to determine whether test samples were included in the training data.

2) Case 2: Training data is encrypted, while the test data remains non-encrypted. The model fails to make accurate predictions, making it impossible for attackers to determine whether test samples were included in the training set.

The proposed methodology provides a strong defense against membership inference for attackers, while maintaining high prediction accuracy for legitimate users without additional safeguards.

The remainder of this study is organized as follows: Section II introduces the concept of membership inference attacks and their implications for machine learning models. Section III reviews the fundamentals of Gaussian Process Regression. Section IV presents the proposed defense framework against membership inference attacks using Random Unitary Transformation. Section V describes the experimental setup and provides a comparative evaluation of the proposed method. Section VI concludes the study with key findings, while Section VII discusses limitations and outlines directions for future research.

[1]Part of this work has been presented at CSP 2025 [13].

II. Membership Inference Attack

MIAs target ML models with the intent of determining whether a data record was included in the training dataset. These attacks pose serious privacy risks, especially when the mere inclusion of a record reveals sensitive information. For instance, if an attacker learns that a clinical record was used to train a disease-specific model, it may strongly suggest that the individual associated with that record has the disease in question. A recent report by the U.S. National Institute of Standards and Technology (NIST) [14] classifies such inferences as violations of confidentiality. These risks are particularly concerning for organizations offering Machine Learning as a Service (MLaaS), where exposing models to external queries can unintentionally breach legal privacy protections. Veale et al. [15], for example, highlight that MIAs can cause ML outputs to be considered personal data under the General Data Protection Regulation (GDPR) [16].

The idea of MIAs was first introduced by Homer et al. [17] in the genomics domain. They showed that an attacker could use summary statistics from a genomics dataset to find out if a specific person's genome was included. Later, researchers [18], [19] found that similar attacks could also be used on location data. In machine learning, Shokri et al. [2] were the first to show how MIAs could be used against classification models. They found that by only looking at a model's prediction results—without knowing anything about how the model was built—an attacker could tell whether a certain data point was used during training. Since then, extensive research has explored MIAs across various model types, including regression models [20], generative models [21], and embedding models [22]. Alongside this, a growing body of literature has proposed a range of defense mechanisms aimed at mitigating the risk of MIAs while maintaining model performance.

To further illustrate the architecture of a MIA, we present a conceptual overview in Fig. 1. The diagram depicts how an adversary employs a shadow model to approximate the behavior of a target model and then trains an attack model to distinguish between member and non-member data points based on observed prediction outputs. This framework represents the standard MIA strategy widely adopted in the literature and forms the basis of our experimental implementation. The target model is trained using features and corresponding predictions from training data. An adversary queries the target model using features of new data and collects output responses. These outputs, along with corresponding input features, are used to train an attack model (shadow model) that learns to classify whether a data sample was part of the target model's training set (member) or not (non-member).

III. Gaussian Process Regression

A. Gaussian Process

We focus on a regression task where the inputs are vectors $\mathbf{x}_i \in \mathbb{R}^D$ and the outputs are scalar values $y_i \in \mathbb{R}$. Suppose we have a training dataset $D_{\text{train}} = \{\mathbf{X}, \mathbf{Y}\}$, where the input and output matrices are represented, as in Eq. (1):

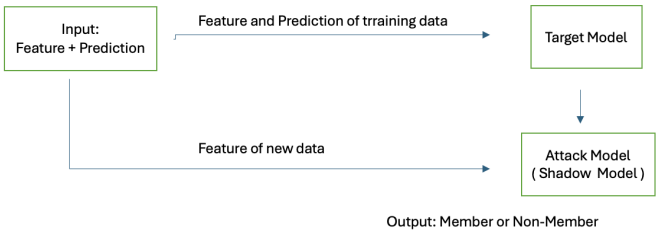

Fig. 1. Architecture of a membership inference attack (MIA) framework.

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}. \tag{1}$$

The outputs are modeled, as in Eq. (2), where $f(X)$ represents the unknown function to be inferred, and ϵ is Gaussian noise with zero mean and variance σ^2, capturing uncertainty and measurement errors.

$$\mathbf{Y} = f(\mathbf{X}) + \epsilon. \tag{2}$$

This formulation captures measurement noise and reflects uncertainty in the observations. The function $f(\mathbf{X})$ is assumed to follow a GP prior, as expressed in Eq. (3):

$$f(\mathbf{X}) \sim \mathcal{GP}(\mathbf{0},\ \mathbf{K(X,X)}). \tag{3}$$

The mean is set to zero for simplicity, and the covariance between inputs is defined by the kernel $\mathbf{K}$, which is usually a symmetric, positive semi-definite matrix. A commonly used covariance function is the RBF kernel, defined in Eq. (4):

$$\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \theta_1 \exp\left(-\frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{\theta_2}\right) + \sigma^2 \delta(i,j), \tag{4}$$

where, $\delta(i,j)$ 1, if i=j, otherwise 0. To fit the model, the hyperparameters θ_1, θ_2 and σ^2 are optimized by minimizing the negative log marginal likelihood, as shown in Eq. (5):

$$\min_{\mathbf{K},\sigma} \mathbf{Y}^T \left(\mathbf{K(X,\ X)} + \sigma^2 \mathbf{I}\right)^{-1} \mathbf{Y} + \log_2 \left|\mathbf{K(X,\ X)} + \sigma^2 \mathbf{I}\right|. \tag{5}$$

This optimization is typically done via gradient descent [23].

B. Gaussian Process Regression

To make predictions for new input data $\mathbf{x}^* \in \mathbb{R}^D$, we define a test set $D_{\text{test}} = \{\mathbf{X}^*, \mathbf{Y}^*\}$ in Eq. (6):

$$\mathbf{X}^* = \begin{bmatrix} \mathbf{x}_1^* \\ \mathbf{x}_2^* \\ \vdots \\ \mathbf{x}_M^* \end{bmatrix}, \quad \mathbf{Y}^* = \begin{bmatrix} y_1^* \\ y_2^* \\ \vdots \\ y_M^* \end{bmatrix}. \tag{6}$$

The joint distribution of Y and $f(X^*)$ follows a multivariate Gaussian, as expressed in Eq. (7):

$$\begin{bmatrix} \mathbf{Y} \\ f(\mathbf{X}^*) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{K(X,\ X)} + \sigma^2 \mathbf{I} & \mathbf{K(X,X^*)} \\ \mathbf{K(X^*, X)} & \mathbf{K(X^*, X^*)} \end{bmatrix}\right). \tag{7}$$

In this formulation, $\mathbf{K(X,X^*)}$ denotes the covariance matrix capturing the dependencies between the N training inputs and the Mtest inputs. Conditioning on the training data, the posterior distribution of $f(X^*)$ is obtained, as in Eq. (8):

$$p(f(\mathbf{X}^*)|(\mathbf{X,Y,X}^*)) \sim \mathcal{N}(f(\mathbf{X}^*), \sigma^2(\mathbf{X}^*)), \tag{8}$$

where, the predictive mean and variance for test inputs are derived using Eq. (9) and Eq. (10), respectively.

$$f(\mathbf{X}^*) = \mathbf{K}(\mathbf{X}^*, \mathbf{X})^T \left[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}\right]^{-1} \mathbf{Y} \tag{9}$$

$$\sigma^2(\mathbf{X}^*) = \mathbf{K}(\mathbf{X}^*, \mathbf{X}^*) - \mathbf{K}(\mathbf{X}^*, \mathbf{X})^T \left[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}\right]^{-1} \mathbf{K}(\mathbf{X}^*, \mathbf{X}). \tag{10}$$

These equations give us both the predicted mean and the uncertainty (variance) for the test data.

IV. Defending Against Gaussian Process Membership Inference Attack

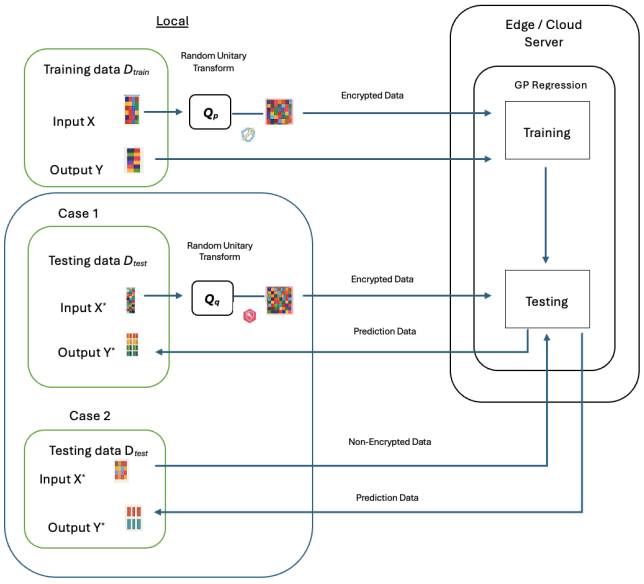

Fig. 2. System configuration of defending against Gaussian process membership inference attack.

A. System Configuration

Fig. 2 illustrates the system architecture designed to defend against MIAs in GP models. The process begins with the collection of training data $D_{\text{train}} = \{\mathbf{X}, \mathbf{Y}\}$, at a local site. To protect the input features $\mathbf{X}$, a random unitary matrix $\mathbf{Q}_p$,

generated using a private encryption key p is applied. This transforms the raw inputs into their encrypted form $\hat{\mathbf{X}}$, which is then transmitted along with the corresponding outputs $\mathbf{Y}$ to an edge or cloud server. At the inference stage, test inputs $\mathbf{X}^*$ are similarly encrypted using another random unitary matrix $\mathbf{Q}_q$, derived from a separate secret key q, resulting in encrypted test inputs $\hat{\mathbf{X}^*}$. These are also sent to the edge or cloud server for processing. This encryption scheme ensures data confidentiality while enabling efficient GP-based inference in distributed environments.

Using the encrypted datasets, the server computes the kernel function and estimates the mean and variance of predictions. When both training and test data are encrypted using the same transformation key (i.e., $p = q$, the predicted results (mean and variance) remain consistent with those produced using non-encrypted data. This property guarantees that encryption does not distort the inference outcomes. In scenarios where the attacker attempts to manipulate the system by inserting non-encrypted test data, the server still processes the data using the encrypted training inputs. The resulting inconsistency in kernel computations between the encrypted and non-encrypted datasets limits the attacker's ability to infer training membership, thereby enhancing the system's privacy protection.

B. Secure Computation

To ensure secure GP regression, the proposed method employs random unitary transformations to encrypt both training and testing data. For the training inputs $\mathbf{X}$, a unitary matrix $\mathbf{Q}_p$ is created using a private key p, resulting in encrypted inputs $\hat{\mathbf{X}} = \mathbf{X}\mathbf{Q}_p$. Similarly, the test inputs $\mathbf{X}^*$ are encrypted as $\hat{\mathbf{X}^*} = \mathbf{X}^*\mathbf{Q}_q$, using a different unitary matrix $\mathbf{Q}_q$ generated with a separate key q, as in Eq. (11):

$$\hat{\mathbf{X}} = \mathbf{X}\mathbf{Q}_p = \begin{bmatrix} \hat{\mathbf{x}}_1 \\ \hat{\mathbf{x}}_2 \\ \vdots \\ \hat{\mathbf{x}}_N \end{bmatrix}, \quad \hat{\mathbf{X}}^* = \mathbf{X}^*\mathbf{Q}_q = \begin{bmatrix} \hat{\mathbf{x}}_1^* \\ \hat{\mathbf{x}}_2^* \\ \vdots \\ \hat{\mathbf{x}}_M^* \end{bmatrix}. \tag{11}$$

This transformation is applied to every element of the input vectors, meaning all input features are protected during transmission and computation. The GP model then estimates the function $f(\hat{\mathbf{X}})$ from the encrypted training data, modeling the outputs, as in Eq. (12):

$$\mathbf{Y} = f(\hat{\mathbf{X}}) + \epsilon. \tag{12}$$

This equation says that the observed output $\mathbf{Y}$ (e.g., a medical measurement) is made up of the model's prediction $f(\hat{\mathbf{X}})$ plus some random noise ϵ to account for uncertainty or measurement error.

Next, we establish a joint Gaussian distribution that encompasses both the training outputs and the model's function values at the encrypted test points, as detailed in Eq. (13):

$$\begin{bmatrix} \mathbf{Y} \\ f(\hat{\mathbf{X}}^*) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{K}(\hat{\mathbf{X}}, \hat{\mathbf{X}}) + \sigma^2\mathbf{I} & \mathbf{K}(\hat{\mathbf{X}}, \hat{\mathbf{X}}^*) \\ \mathbf{K}(\hat{\mathbf{X}}^*, \hat{\mathbf{X}}) & \mathbf{K}(\hat{\mathbf{X}}^*, \hat{\mathbf{X}}^*) \end{bmatrix} \right). \tag{13}$$

This joint covariance matrix characterizes a multivariate normal distribution over the training outputs and the function values at the test points. The top-left block $K(\hat{\mathbf{X}}, \hat{\mathbf{X}}) + \sigma^2 I$ captures the dependencies among the encrypted training inputs. The bottom-right $\mathbf{K}(\hat{\mathbf{X}^*}, \hat{\mathbf{X}^*})$, reflects the covariance structure among the encrypted test inputs. The off-diagonal blocks, $\mathbf{K}(\hat{\mathbf{X}}, \hat{\mathbf{X}}^*)$ and $\mathbf{K}(\hat{\mathbf{X}}^*, \hat{\mathbf{X}})$, represent the interaction between the training and test data. The term $\sigma^2 I$ accounts for observation noise, introducing a measure of uncertainty into the model.

From this joint distribution, we derive the posterior prediction—that is, the model's estimate for the test outputs based on the training data:

$$f(\hat{\mathbf{X}}^*) \quad = \quad \mathbf{K}(\hat{\mathbf{X}}^*, \hat{\mathbf{X}})^T \left[\mathbf{K}(\hat{\mathbf{X}}, \hat{\mathbf{X}}) + \sigma^2\mathbf{I} \right]^{-1} \mathbf{Y}. \tag{14}$$

This Eq. (14) is used to generate predictions for the encrypted test inputs $\hat{\mathbf{X}}^*$. It leverages the relationships between test and training inputs, as captured by the kernel function, the internal structure among the training inputs, and the observed training outputs $\mathbf{Y}$.

Eq. (15) provides the variance (uncertainty) of those predictions:

$$\begin{aligned} \sigma^2(\hat{\mathbf{X}}^*) \quad &= \quad \mathbf{K}(\hat{\mathbf{X}}^*, \hat{\mathbf{X}}^*) \\ &- \quad \mathbf{K}(\hat{\mathbf{X}}^*, \hat{\mathbf{X}})^T \left[\mathbf{K}(\hat{\mathbf{X}}, \mathbf{X}) + \sigma^2\mathbf{I} \right]^{-1} \mathbf{K}(\hat{\mathbf{X}}^*, \hat{\mathbf{X}}). \end{aligned} \tag{15}$$

This expression quantifies the model's predictive uncertainty for each test input. A lower variance indicates higher confidence in the prediction, whereas a higher variance suggests greater uncertainty. The magnitude of this variance reflects how closely the test input resembles the training data distribution.

C. Encryption Based on Random Unitary Transform

Prior research has investigated secure sparse coding techniques based on random unitary transformation [24], [25], [26], [27], demonstrating that it is possible to protect sensitive data while preserving key geometric structures. To ensure data privacy in Gaussian Process Regression (GPR), this study adopts a transformation-based encryption mechanism utilizing random unitary matrices. This technique obfuscates input data while preserving essential geometric properties required for kernel-based learning. Unlike additive noise methods such as DP-GPR, this approach enables secure learning without degrading model accuracy.

Let $\mathbf{x}_i^* \in \mathbb{R}^D$ denote a raw input vector. A unitary matrix $\mathbf{Q}_p \in \mathbb{C}^{DXD}$, generated using a private key p, is applied to encrypt the data, as shown in Eq. (16):

$$\hat{\mathbf{x}_i} = \mathbf{x}_i \mathbf{Q}_p. \tag{16}$$

This transformation yields an encrypted representation $\hat{\mathbf{x}}_i$, where $\mathbf{Q}_p$ satisfies the unitary condition given in Eq. (17):

$$\mathbf{Q}_p^H \mathbf{Q}_p = \mathbf{I}. \tag{17}$$

Here, $\mathbf{Q}_P^H$ denotes the Hermitian transpose, and $\mathbf{I}$ is the identity matrix. This property ensures that the transformation is norm-preserving and invertible, enabling the encrypted inputs to retain structural similarity to the original data. The encryption preserves the following fundamental properties:

- Property 1: Norm Isometry

$$||\mathbf{x}_\mathrm{i}||_2^2 = ||\widehat{\mathbf{x}}_\mathrm{i}||_2^2$$

- Property 2: Distance Preservation

$$||\mathbf{x}_\mathrm{i} - \mathbf{x}_\mathrm{j}||_2^2 = ||\widehat{\mathbf{x}}_\mathrm{i} - \widehat{\mathbf{x}}_\mathrm{j}||_2^2$$

- Property 3: Inner Product Preservation

$$\mathbf{x}_i^H \mathbf{x}_\mathrm{j} = \widehat{\mathbf{x}}_i^H \widehat{\mathbf{x}}_j$$

These preserved characteristics are essential for accurate kernel computation in GPR, particularly when using the Radial Basis Function (RBF) kernel. The kernel function on the encrypted data is expressed, as in Eq. (18):

$$K(\hat{\mathbf{x}_i}, \hat{\mathbf{x}_j}) = \theta_1 \exp\left(-\frac{||\hat{\mathbf{x}_i} - \hat{\mathbf{x}_j}||^2}{\theta_2}\right) + \sigma^2 \delta(i,j). \tag{18}$$

Substituting $\hat{\mathbf{x}}_i = \mathbf{x}_i \mathbf{Q}_p$ and $\hat{\mathbf{x}}_j = \mathbf{x}_j \mathbf{Q}_p$, and applying the unitary property, we obtain the distance preservation relation shown in Eq. (19):

$$||\hat{\mathbf{x}_i} - \hat{\mathbf{x}_j}||^2 = ||(\mathbf{x}_i - \mathbf{x}_j)\mathbf{Q}_p||^2 = ||(\mathbf{x}_i - \mathbf{x}_j)||^2. \tag{19}$$

Hence, the kernel function becomes: $K(\hat{\mathbf{x}_i}, \hat{\mathbf{x}_j})$ = $K(\mathbf{x}_i, \mathbf{x}_j)$ This result confirms that encrypting the input vectors with the same key does not alter the pairwise kernel values, thereby preserving model fidelity during training.

While the analysis primarily focuses on the RBF kernel, the properties of random unitary transformation (RUT)—such as norm preservation, inner product invariance, and distance preservation—also ensure compatibility with other kernel functions. For example, the linear kernel, which directly relies on inner products, remains unaffected by RUT. Similarly, polynomial kernels, which are functions of inner products, also retain their behavior under unitary transformations, enabling broad applicability of the proposed method across kernel-based models.

Similarly, if the test inputs $\mathbf{x}_i^* \in \mathbb{R}^D$ are encrypted with the same unitary matrix $\mathbf{Q}_p$, the kernel between test points remains invariant, as expressed as in Eq. (20):

$$K(\hat{\mathbf{x}_i^*}, \hat{\mathbf{x}_j^*}) = K(\mathbf{x}_i^*, \mathbf{x}_j^*). \tag{20}$$

However, when training and test inputs are encrypted using different keys (i.e., $\mathbf{Q}_p \neq \mathbf{Q}_q$), the kernel function between them is modified, as in Eq. (21):

$$K(\hat{\mathbf{x}_i}, \hat{\mathbf{x}_j}) = \theta_1 \exp\left(-\frac{||\hat{\mathbf{x}_i}\mathbf{Q}_q - \hat{\mathbf{x}_j}\mathbf{Q}_p||^2}{\theta_2}\right) + \sigma^2 \delta(i,j). \tag{21}$$

Since $\mathbf{Q}_p \neq \mathbf{Q}_q$, the transformation no longer preserves the distances between training and test samples. Consequently, $K(\hat{\mathbf{x}_i^*}, \hat{\mathbf{x}_j}) \neq K(\mathbf{x}_i, \mathbf{x}_j)$. This mismatch distorts the kernel values, making it infeasible for an adversary to infer training membership based on similarity, thereby enhancing the model's privacy guarantees.

When the same transformation key is applied to both training and test inputs (that is, $\mathbf{Q}_p = \mathbf{Q}_q$), the encrypted GP model produces predictions and uncertainties identical to the plaintext case, as shown in Eq. (22) and Eq. (23):

$$f(\widehat{\mathbf{X}}^*) = f(\mathbf{X}^*), \tag{22}$$

$$\sigma^2(\widehat{\mathbf{X}}^*) = \sigma^2(\mathbf{X}^*). \tag{23}$$

This property ensures that the encryption does not compromise the accuracy of the model.

Finally, when only the training inputs are encrypted (i.e., test data remains non-encrypted), the kernel between test and training data becomes distorted, as expressed in Eq. (24):

$$\begin{aligned} K(\mathbf{x}_\mathrm{i}^*, \widehat{\mathbf{x}}_\mathrm{j}) &= \theta_1 \exp\left(-\frac{|\mathbf{x}_\mathrm{i}^* - \widehat{\mathbf{x}}_\mathrm{j}|^2}{\theta_2}\right) + \sigma^2 \delta(i,j) \\ &= \theta_1 \exp\left(-\frac{|\mathbf{x}_i \mathbf{I} - \mathbf{x}_j \mathbf{Q}_p|^2}{\theta_2}\right) + \sigma^2 \delta(i,j) \\ &\neq K(\mathbf{x}_i, \mathbf{x}_j). \end{aligned} \tag{24}$$

Since the test inputs are in their original form and the training inputs are encrypted, the kernel function again deviates from its non-encrypted counterpart, further reinforcing privacy against inference attacks.

V. Experimental Evaluations

To evaluate the effectiveness of the proposed approach, a series of experiments were conducted using a diabetes dataset commonly employed in medical data analysis.

A. Simulation Conditions

To assess the effectiveness of the proposed secure Gaussian Process Regression (secGPR) framework, experiments were conducted using a publicly available medical dataset related to diabetes. This dataset, sourced from the scikit-learn library, comprises 442 individual records, each containing 10 baseline clinical features such as age, sex, body mass index (BMI), and a quantitative measure of disease progression recorded one year after baseline assessment [28], [29]. For the purpose of model training, a subset of 353 samples was used, while the remaining 89 records were reserved for testing. In this

experimental setup, the input matrix $\mathbf{X} \in \mathbb{R}^{NXD}$ represents the 10-dimensional clinical features, and the corresponding target values $\mathbf{Y} \in \mathbb{R}^{N}$ indicate the diabetes progression scores. This configuration allows for a rigorous evaluation of the secGPR model's ability to accurately predict continuous outcomes under various encryption scenarios.

The random unitary transformation matrix $\mathbf{Q}_p$, used to encrypt the input features, was generated through the Gram–Schmidt orthogonalization process to ensure it satisfies the unitary constraint. This matrix was applied to the input data prior to training and testing in the secure learning environment.

B. Simulation Results

This section presents a comparative analysis of the predictive performance between the conventional GP model and the proposed secure approach.

1) Non-encrypted GP (Where both the training data $\boldsymbol{X}$ and the testing data $\boldsymbol{X}^*$ remain non-encrypted): When the test data was included in the training set, the conventional (non-encrypted) GP model exhibited near-perfect predictive performance, achieving an average error of 1.515×10^{-11} and 1.000×10^{-10}, as reported in Table I. While these results indicate extremely high accuracy, they also reveal a significant security concern: the model's highly deterministic behavior enables adversaries to easily infer whether specific data points were used during training, thus exposing it to MIAs.

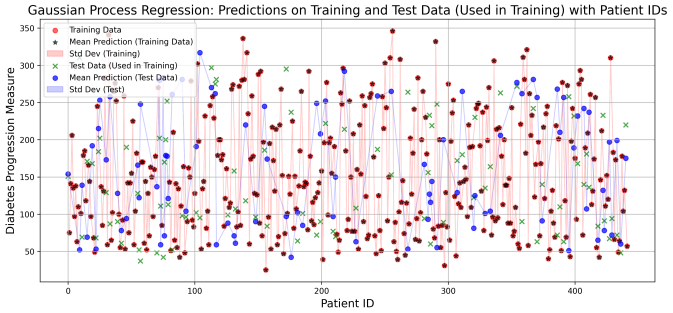

Fig. 3. Mean prediction and uncertainty of conventional GP for non-encrypted training and testing data.

Conversely, when the test data was excluded from the training process, the model's predictive accuracy declined markedly, with the average error rising to 64.32 and the variance increasing to 585.26. This outcome reflects a more realistic deployment scenario in which unseen inputs yield greater uncertainty and lower confidence. The sharp contrast between these two settings highlights a fundamental vulnerability of traditional GPR models in privacy-sensitive applications: they lack an effective balance between predictive performance and data confidentiality.

Fig. 3 illustrates the prediction results of the GPR model in the non-encrypted baseline configuration, where both training and test data are used in their original, untransformed form. The x-axis represents patient IDs, and the y-axis corresponds to the diabetes progression measure.

In this figure:

- Red circles denote the ground truth values of the training data.
- Brown stars represent the model's mean predictions for training samples, with the red shaded region indicating the corresponding predictive uncertainty (standard deviation).
- Green crosses mark the test data samples that were included in the training set.
- Blue circles show the predicted means for these test samples, while the blue shaded region visualizes their associated uncertainty.

The predictions for both training and test data exhibit excellent alignment with the true values, accompanied by consistently low variance. This high confidence and low prediction error—particularly for test data reused during training—indicates strong memorization behavior. While desirable from a model accuracy standpoint, this deterministic prediction pattern introduces a substantial privacy vulnerability. Specifically, the marked discrepancy in model behavior between training and unseen test inputs (not shown in this figure) can be exploited in MIAs. In this setting, an adversary can reliably infer whether a data point was part of the training set by observing prediction confidence and error.

TABLE I. COMPARATIVE PERFORMANCE ACROSS METHODS

Scenario	Condition	Average Error	Average Variance
Non-encrypted GP	Test Data Included	1.515×10^{-11}	1.000×10^{-10}
	Test Data Excluded	64.32	585.26
DP-GPR	Test Data Included	45.7158	342.4383
	Test Data Excluded	50.2886	345.6273
Proposed GP: Case 1 ($P = Q$)	Test Data Included	1.515×10^{-11}	1.000×10^{-10}
	Test Data Excluded	64.32	585.26
Proposed GP: Case 1 ($P \neq Q$)	Test Data Included	102.604	838.253
	Test Data Excluded	87.07	850.8863
Proposed GP: Case 2	Test Data Included	113.96	843.99
	Test Data Excluded	103.74	827.38

2) DP-GPR: The DP-GPR based model achieves moderate privacy protection by injecting noise into the training process. It achieves relatively lower error and variance compared to Case 2 (included: error = 45.7158, variance = 342.4383; excluded: error = 50.2886, variance = 345.6273), while still maintaining acceptable accuracy. However, the residual gap between seen and unseen data behaviors indicates that some MIA risk remains. Moreover, DP-GPR distorts the internal structure of the data and relies heavily on tuning privacy budgets, potentially impacting generalization.

Fig. 4 illustrates the predictive performance of the DP-GPR model when the test data is included in the training set. The horizontal axis represents the individual patient IDs, while the vertical axis corresponds to the diabetes progression measure. The red dots denote the true test data values, serving as the ground truth. The blue line with star markers represents the mean predictions made by the DP-GPR model. The pink shaded region around the prediction curve indicates the standard deviation of the predictions, capturing the model's predictive uncertainty. As shown, the prediction curve closely follows the general trend of the true values, but several true points deviate significantly—especially in regions of higher variance.

These deviations are partially accounted for by the uncertainty bands, which widen in areas where the model expresses less confidence. The presence of these variability bands is a result of the noise added by the DP-GPR mechanism, which affects both prediction sharpness and confidence calibration.

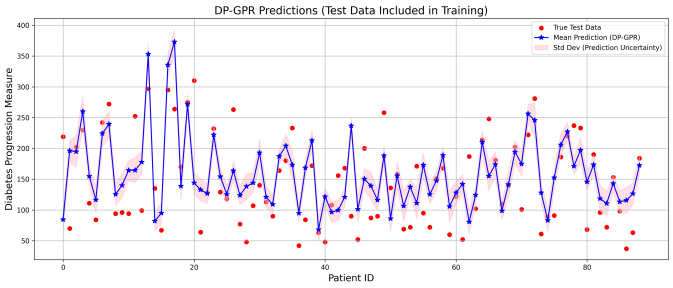

Fig. 4. Mean prediction and uncertainty of DP-GPR.

Fig. 4 highlights a key characteristic of DP-GPR: while the model maintains reasonable alignment with the true data, it introduces a moderate level of uncertainty due to privacy-preserving noise. Importantly, the overall prediction behavior remains consistent, though less precise compared to non-private or fully encrypted models, such as those in the proposed method's Case 1.

3) Case 1 (Both the training data $\boldsymbol{X}$ and the test data $\boldsymbol{X}^*$ were encrypted using random unitary transformations): Two scenarios were evaluated to assess the effectiveness of the proposed method: one in which the same key was used for both transformations ($\mathbf{Q}_p = \mathbf{Q}_q$), and another in which different keys were applied ($\mathbf{Q}_p \neq \mathbf{Q}_q$).

In the key mismatch scenario ($\mathbf{Q}_p \neq \mathbf{Q}_q$), the model's predictive performance declined due to the inconsistency in the transformed input spaces. When test data was included in the training set, the average prediction error and variance were 102.60 and 838.25, respectively. When test data was excluded, the error reduced to 87.07, accompanied by a slight increase in variance to 850.89. These results indicate that although encryption mismatches degrade accuracy, they also enhance privacy by disrupting kernel-based similarity, thereby complicating membership inference attempts.

In contrast, the key match scenario ($\mathbf{Q}_p = \mathbf{Q}_q$) yielded prediction results indistinguishable from those of the non-encrypted model. Specifically, when test data was part of the training set, the average error and variance were extremely low— 1.515×10^{-11} and 1.000×10^{-10}, respectively. Even when the test data was excluded, the model maintained strong performance with an error of 64.32 and variance of 585.26. These findings demonstrate that when consistent encryption is applied, the proposed method retains full predictive accuracy while ensuring data confidentiality.

Fig. 5 presents the predictive results of the proposed GPR model under the Case 1 configuration, where both training and test inputs are encrypted using the same random unitary matrix (i.e., $\mathbf{Q}_p = \mathbf{Q}_q$). The x-axis represents patient IDs, while the y-axis denotes diabetes progression measurements.

In the plot, red dots correspond to the true training data, with black stars showing the model's predicted means and red

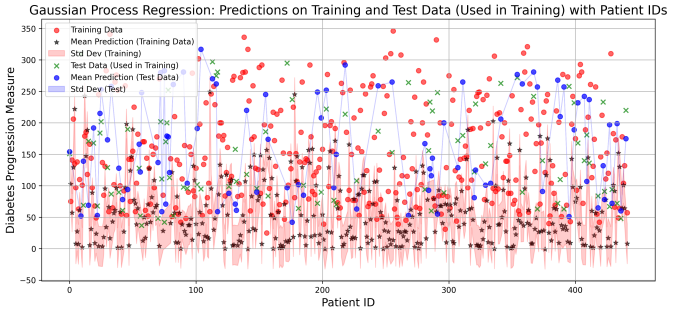

Fig. 5. Mean prediction and uncertainty of proposed method (Case 1) for encrypted training and testing data.

shaded areas indicating prediction uncertainty. Green crosses represent the encrypted test data (used in training), while blue circles and blue shaded regions indicate their predicted means and standard deviations, respectively.

The figure shows that the model achieves highly accurate predictions and low uncertainty for both datasets, indicating that structural data relationships are preserved under consistent encryption.

4) Case 2 (Where training data $\boldsymbol{X}$ is encrypted and testing data $\boldsymbol{X}^*$ is non-encrypted): This asymmetric configuration, referred to as a mixed encryption state, resulted in the poorest predictive performance among all evaluated scenarios. When the test data was included in the training set, the model exhibited the highest average error and variance, recorded at 113.96 and 844.00, respectively (see Table I). Excluding the test data from training slightly improved the outcomes, with the average error decreasing to 103.74 and the variance dropping to 827.38. Despite this modest reduction, the performance remained significantly worse than in fully encrypted or non-encrypted settings.

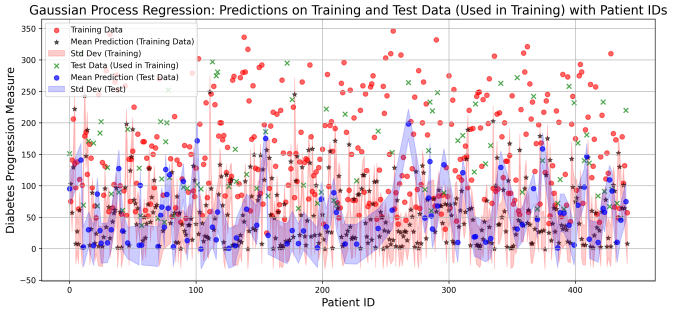

Fig. 6. Mean prediction and uncertainty of proposed method (Case 2) for encrypted training data and non-encrypted testing data.

Fig. 6 illustrates the prediction performance of the proposed GPR model under the Case 2 configuration, in which the training data is encrypted using a random unitary matrix $\mathbf{Q}_p$, while the test data remains non-encrypted. The x-axis represents the patient IDs, and the y-axis indicates the diabetes progression measure.

In this figure:

- Red circles correspond to the ground truth values for the training data.

- Black stars represent the mean predictions for the training data, and the red shaded area denotes the associated prediction uncertainty (standard deviation).
- Green crosses mark the true test data values (used in training).
- Blue circles indicate the predicted mean values for the test data, with the blue shaded area visualizing their predictive uncertainty.

Despite the fact that test data are included in training, the model demonstrates substantially degraded prediction accuracy and elevated uncertainty across both training and test datasets. This performance degradation arises from the incompatibility between the encrypted training data and the non-encrypted (or differently encrypted) test data, which undermines kernel consistency and limits the model's ability to capture meaningful relationships.

These results indicate that the mixed encryption scenario in Case 2 provides a high level of privacy by intentionally disrupting the correspondence between encrypted and non-encrypted data. The resulting reduction in predictive accuracy is not a drawback, but rather a desirable characteristic from a privacy perspective—specifically, it ensures that attackers cannot make accurate estimations about whether a given sample was part of the training set. The sharp degradation in model performance serves to obscure membership signals, thereby strengthening the model's resistance to membership inference attacks (MIAs).

VI. Conclusion

This study introduced a noise-free privacy-preserving framework for Gaussian Process Regression based on Random Unitary Transformation. The method supports configurations that either preserve accuracy (Case 1) or prioritize privacy robustness (Case 2). Compared with DP-GPR, which relies on noise injection, the proposed framework maintains model fidelity while enhancing resistance to MIAs.

The results demonstrate that Case 1 achieves excellent predictive accuracy, comparable to the non-encrypted baseline, while ensuring structural privacy without introducing noise. However, the observed discrepancy in prediction behavior between training and unseen data introduces a moderate vulnerability to MIAs. In contrast, Case 2 offers the strongest privacy protection, as it produces uniformly high error and uncertainty regardless of whether a data point was part of the training set. This consistency effectively conceals membership status, though at the cost of reduced model utility.

Compared to DP-GPR, which achieves moderate privacy through noise injection, the proposed framework provides a noise-free alternative that either maintains model fidelity or maximizes privacy robustness. While DP-GPR strikes a balance between accuracy and privacy, it fails to eliminate membership leakage entirely and distorts the data's statistical structure.

Overall, the proposed approach demonstrates flexibility in supporting different privacy scenarios without compromising predictive accuracy. In Case 1, where the same key is used to encrypt both training and test data, the model maintains predictive performance equivalent to the non-encrypted baseline. However, this configuration introduces a potential attack surface: if the shared key is ever leaked, inferred, or reused, it could allow adversaries to align encrypted inputs and compromise membership privacy. In contrast, Case 2, where only the training data is encrypted, offers stronger privacy protection by fully disrupting alignment between training and test distributions. The resulting decrease in predictive accuracy is not a limitation, but a privacy-enhancing effect—by weakening the correspondence between training and test samples, it becomes significantly more difficult for adversaries to infer membership or reconstruct sensitive data. These findings highlight the promise of unitary transformation-based encryption as a practical, interpretable, and privacy-preserving solution for sensitive machine learning applications.

VII. Limitations and Future Work

While the proposed encryption-based GPR framework demonstrates strong potential in balancing privacy and predictive performance, several limitations warrant further investigation. First, the current evaluation is conducted on a single dataset with a moderate number of features and samples. Additional experiments on larger, more diverse datasets—particularly in high-dimensional or real-time environments—are necessary to assess the scalability and generalizability of the approach. Second, although Case 2 offers strong resistance to MIAs, it does so at the cost of significantly degraded predictive accuracy, which may limit its practicality in applications where precision is critical. Furthermore, this work assumes a passive attacker model; evaluating robustness under more aggressive or adaptive adversarial strategies (e.g., shadow models, reconstruction attacks) remains an open area.

Future research could explore hybrid defense mechanisms that combine unitary transformation with adaptive noise injection or output randomization to further enhance privacy without significantly compromising predictive accuracy. Investigating how this framework performs under federated learning or distributed settings is also a promising direction—particularly in scenarios where local privacy constraints and communication efficiency are critical. These directions would broaden the applicability of the proposed method and address emerging challenges in real-world privacy-preserving machine learning systems.

References

[1] T. Leemann, M. Pawelczyk, and G. Kasneci, "Gaussian membership inference privacy," Advances in Neural Information Processing Systems, 36, pp.73866-73878, 2024.

[2] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," In 2017 IEEE symposium on security and privacy (SP) (pp. 3-18), 2017.

[3] S. K. Murakonda, and R. Shokri, "Ml privacy meter: Aiding regulatory compliance by quantifying the privacy risks of machine learning," arXiv preprint arXiv:2007.09339, 2020.

[4] S. Song, and D. Marn, "Introducing a new privacy testing library in tensorflow," URL https://blog. tensorflow. org/2020/06/introducing-new-privacy-testing-library. html, 2020.

[5] E. De Cristofaro, "An overview of privacy in machine learning," arXiv preprint arXiv:2005.08679, 2020.

[6] H. Liu, Y. S. Ong, X. Shen, and J. Cai, "When Gaussian process meets big data: A review of scalable GPs," IEEE Transactions on neural networks and learning systems, 31(11), 4405-4423, 2020.

[7] R. M. Neal, "Bayesian learning for neural networks," (Vol. 118). Springer Science & Business Media, 2012.

[8] C. Dwork, F. McSherry, K. Nissim and A. Smith, "Calibrating noise to sensitivity in private data analysis," In Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3 (pp. 265-284). Springer Berlin Heidelberg, 2006.

[9] M. Abadi, A. Chu, I. Goodfellow, H.B. McMahan, I. Mironov, K. Talwar and L. Zhang, October. "Deep learning with differential privacy," In Proceedings of the 2016 ACM SIGSAC conference on computer and communications security (pp. 308-318), 2016.

[10] G. Bernstein, and D.R. Sheldon, "Differentially private bayesian linear regression," Advances in Neural Information Processing Systems, 32, 2019.

[11] M. Heikkilä, J. Jälkö, O. Dikmen and A. Honkela, "Differentially private markov chain monte carlo," Advances in Neural Information Processing Systems, 32, 2019.

[12] J. Jälkö, O. Dikmen and A. Honkela, "Differentially private variational inference for non-conjugate models," arXiv preprint arXiv:1610.08749, 2016.

[13] M. R. Islam, J. F. Akhi, and T. Nakachi, "Defending Against Gaussian Process Membership Inference Attack," presented at the *IEEE Conf. on Cyber Security and Privacy (CSP)*, Okinawa, Japan, 2025.

[14] E. Tabassi, K.J. Burns, M. Hadjimichael, A.D. Molina-Markham and J.T. Sexton, "A taxonomy and terminology of adversarial machine learning," NIST IR, pp.1-29, 2019.

[15] M. Veale, R. Binns, and L. Edwards, "Algorithms that remember: model inversion attacks and data protection law," Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 376(2133), p.20180083, 2018.

[16] European Parliament and Council of the European Union, "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)," Official Journal of the European Union, vol. L 119, pp. 1–88, May 2016.

[17] N. Homer, S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, J.V. Pearson, D.A. Stephan, S.F. Nelson and D.W. Craig, "Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays," PLoS genetics, 4(8), p.e1000167, 2008.

[18] A. Pyrgelis, C. Troncoso and E. De Cristofaro, "Knock knock, who's there? Membership inference on aggregate location data," arXiv preprint arXiv:1708.06145, 2018.

[19] A. Pyrgelis, C. Troncoso, and E. De Cristofaro, "Measuring membership privacy on aggregate location time-series," Proceedings of the ACM on Measurement and Analysis of Computing Systems, 4(2), pp.1-28, 2020.

[20] U. Gupta, D. Stripelis, P.K. Lam, P. Thompson, J.L. Ambite and G. Ver Steeg, "Membership inference attacks on deep regression models for neuroimaging," In Medical Imaging with Deep Learning (pp. 228-251). PMLR, 2021.

[21] J. Hayes, L. Melis, G. Danezis and E.L. De Cristofaro, "Membership Inference Attacks Against Generative Models"; URL https://api. semanticscholar. org/CorpusID, 202588705, 2018.

[22] C. Song and A. Raghunathan, "Information leakage in embedding models," In Proceedings of the 2020 ACM SIGSAC conference on computer and communications security (pp. 377-390), 2020.

[23] C. K. Williams, and C. E. Rasmussen, "Gaussian processes for machine learning," (Vol. 2, No. 3, p. 4). Cambridge, MA: MIT press, 2006.

[24] T. Nakachi, Y. Bandoh, and H. Kiya, "Secure overcomplete dictionary learning for sparse representation," IEICE Transactions on Information and Systems, 103(1), 50-58, 2020.

[25] T. Nakachi, and H. Kiya, "Secure OMP computation maintaining sparse representations and its application to EtC systems," IEICE Transactions on Information and Systems, 103(9), 1988-1997, 2020.

[26] Y. Wang, and T. Nakachi, "A privacy-preserving learning framework for face recognition in edge and cloud networks," IEEE Access, 8, 136056-136070, 2020.

[27] Y. Bandoh, T. Nakachi, and H. Kiya, "Distributed secure sparse modeling based on random unitary transform," IEEE Access, 8, 211762-211772, 2020.

[28] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," The Annals of Statistics, vol. 32, no. 2, pp. 407–499, 2004

[29] scikit-learn, https://scikit-learn.org/stable/index.html.