Predicting Primary Sequence-Based Protein-Protein Interactions Using a Mercer Series Representation of Nonlinear Support Vector Machine

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 1 November 2022, accepted 16 November 2022, date of publication 21 November 2022,
date of current version 1 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3223994

RESEARCH ARTICLE

Predicting Primary Sequence-Based Protein-Protein Interactions Using a Mercer Series Representation of Nonlinear Support Vector Machine

OMID CHATRABGOUN[1,2], ALIREZA DANESHKHAH[1,3], MOHSEN ESMAEILBEIGI[4], NADER SOHRABI SAFA[5], ALI H. ALENEZI[6,7], AND ARAFATUR RAHMAN[5]

[1]School of Computing, Electronics and Mathematics, Coventry University, CV1 5FB Coventry, U.K.
[2]Department of Statistics, Faculty of Mathematical Sciences and Statistics, Malayer University, Malayer 65719-95863, Iran
[3]Centre for Computational Science and Mathematical Modelling, Coventry University, CV1 5FB Coventry, U.K.
[4]Department of Mathematics, Faculty of Mathematical Sciences and Statistics, Malayer University, Malayer 65719-95863, Iran
[5]School of Engineering, Computing and Mathematical Sciences, University of Wolverhampton, WV1 1LY Wolverhampton, U.K.
[6]Department of Electrical Engineering, Northern Border University, Arar 92341, Saudi Arabia
[7]Remote Sensing Unit, Northern Border University, Arar 92341, Saudi Arabia

Corresponding author: Omid Chatrabgsoun (ad8337@coventry.ac.uk)

ABSTRACT The prediction of protein-protein interactions (PPIs) is essential to understand the cellular processes from a medical perspective. Among the various machine learning techniques, kernel-based Support Vector Machine (SVM) has been commonly employed to discriminate between interacting and non-interacting protein pairs. The main drawback of employing the kernel-based SVM to datasets with many features, such as the primary sequence-based protein-protein dataset, is the significant increase in computational time of training stage. This increase in computational time is mainly due to the presence of the kernel in solving the quadratic optimisation problem (QOP) involved in nonlinear SVM. In order to fix this issue, we propose a novel and efficient computational algorithm by approximating the kernel-based SVM using a low-rank truncated Mercer series as well as desired. As a result, the QOP for the approximated kernel-based SVM will be very tractable in the sense that there is a significant reduction in computational time of training and validating stages. We illustrate the novelty of the proposed method by predicting the PPIs of "S. Cerevisiae" where the protein features extracted using the multiscale local descriptor (MLD), and then we compare the predictive performance of the proposed low-rank approximation with the existing methods. Finally, the new method results in significant reduction in computational time for predicting PPIs with almost as accuracy as kernel-based SVM.

INDEX TERMS Kernel-based SVM, protein-protein interactions, quadratic optimisation problem, Mercer series.

I. INTRODUCTION

The study of PPIs is very important for understanding the biological cellular functions, and it would be also very useful to better learning about the mechanisms of action of several diseases [1]. However, detecting PPIs in the laboratories (e.g. yeast two-hybrid systems (Y2H) [2], mass spectrometry (MS) [3], tandem affinity purification (TAP) [4], and protein chip [5]) would be time-consuming and very expensive [6]. Although, much progress has already been achieved in this direction, the problem is still far from being solved. For this reason, there has been much recent effort to develop techniques for computational prediction of PPIs including genomic context-based methods [7], [8], structure-based methods [9], [10], and sequence-based methods [11], [12].

The associate editor coordinating the review of this manuscript and approving it for publication was Ines Domingues.

Given the problems that arise in the first two methods, sequence-based methods have shown the advantage of generalisation because they require information only from amino-acid sequences [13]. In the majority of the sequence-based methods, the use of machine learning (ML) techniques (e.g., Random Forest and Naive Bayse) for building classifiers [14] has been strongly recommended. Furthermore, the existing approaches are typically focused on the binary classification frameworks for the protein pairs (interacting and non-interacting pairs), but they differ in a way that how the features extracted from the protein pairs. However, SVM [15] can be viewed as an efficient and interesting ML algorithm for determining PPIs [12], [16], [11]. Furthermore, the kernel-based SVM which resulted from combining the conventional SVM with different kernels provides even a more efficient tool to predict PPIs for the highly complex cases. The flexibility and accuracy of the kernel-based SVMs for the classification can be further enhanced by appropriately estimating the kernels' parameters [17], [18].

Despite the usefulness of the kernel-based SVMs as described above, but they would become computationally very demanding for the datasets with many features [19], [20]. In particular, this computational complexity would be frequently observed when kernel-based SVM was used for predicting the PPI. This can be expected due to the large number of features/attributes of this dataset that would result in making training and prediction very slow for the complex cases and consequently increasing the required time for classification [21]. This computational complexity is due to the presence of kernel in the quadratic optimisation problem (QOP) involved in kernel-based SVMs, which makes the required Hessian matrix in QOP to be very dense (for dataset with many features), and subsequently becomes ill-conditioned. It is thus very essential to develop an efficient approach to resolve the above optimisation challenge for the kernel-based SVMs, which could be resulted in overcoming the discussed computational complexity, and consequently reducing the computational time for predicting PPI while at least the same accuracy can be achieved. In this regard, there have been recently several efforts in reducing the computational complexity. For example, data reduction methods have been used to reduce the computational complexity and costs in the kernel-based SVMs by dimensional reduction of the original data [22]. The main drawback of these methods to predict PPIs with too many features, is that some valuable information in the sequence-based protein or their important features could be lost through the dimension reduction techniques. Also, using such methods for predicting PPIs is misleading because the attributes here aren't original data sets and only they are some indexes that extracted from sequenced-based proteins. An alternative method to resolve the above issue is to reduce the computational challenge by randomly selecting some columns of the kernel in the SVMs structure [23]. Selecting these columns are not always straightforward, and this could create another computational challenge. The methodology proposed in [24], which is based on transforming the kernel matrix into a sparse matrix, also has its own problems [25].

In this paper, we propose a novel method to predict PPIs by constructing a low-rank approximation of the kernel matrix, based on the truncated Mercer series expansion of the underlying kernel. There are some techniques for the kernel approximation based on the eigenvalues and eigenfunctions (e.g. Cholesky decomposition method [37], the RBF-QR method [38], and weighted SVD bases method [39]). However, solving updated QOP by deriving the alternative bases via decomposing the kernel matrix based on these strategies are still suffering from the same instabilities and ill-conditioning presented in the primary QOP. Alternatively, using the proposed method in this paper, the complex QOP described above will be replaced by a significantly simpler and computationally cheaper optimisation problem by resolving this issue by decomposition of the kernel matrix based on the Hilbert–Schmidt (HS) SVD as an alternative bases which enable us to numerically solve the QOP without any need to decompose the kernel matrix in the same unstable way as previous approaches. It is clear that SVD decomposition for the kernel matrix is not new topic but to the best of our knowledge this is the first time that we apply the decomposition of the kernel matrix based on the Hilbert–Schmidt (HS) SVD as an alternative basis for fixing computational issues in the kernel-based SVM. Therefore, using the proposed method, the complex QOP will be replaced by a significantly simpler and computationally cheaper optimization problem without any demand to decompose the kernel matrix. By considering this approximation, which is based on the truncated Mercer series theorem, we approximate matrix K in the second-order Dual optimization problem involved in QOP by using only the first M terms of the expansion. Fortunately, in many kernels for very small amounts of M, we can obtain very accurate approximations of the kernel K. The proposed low-rank SVM is constructed as a function of eigenvalues and eigenfunctions of Hilbert-Schmidt operator of the selected kernel. Since these eigenvalues are decreasing, the kernel approximation can be easily derived using the first several terms of the expansions. The resulting low-rank approximation of the kernel in SVM can be made as accurate as desired by appropriately truncating the Mercer series expansion at the first several terms. In order to maintain this accuracy and make the low-rank approximation computationally more tractable, a new Hessian matrix involved in the quadratic optimisation of the kernel-based SVM will be derived that is sparse, and very straightforward to compute. It should be noted that the computed Hessian matrix in the full-rank form of the SVM kernel is highly dense, which would cause the computation very complex. Finally, We apply the proposed method in this paper for predicting PPIs in "S. Cerevisiae". It is demonstrated that the proposed approach in this paper results in significant reduction in computational time, and would enhance flexibility and efficiency in predicting PPIs as accurate as desired.

This paper is organised as follows. In Section II, we describe the kernel-based SVM method very briefly,

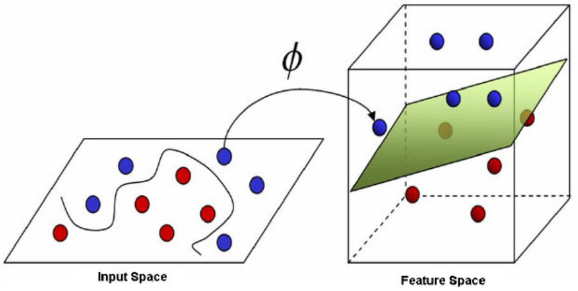


FIGURE 1. **Observed data points are not linearly separable in the input space, but they are in the feature space [26].**

and discuss its computational challenges for predicting PPIs with many features. Then, we define the low-rank kernel approximation, and how the computational challenges of the kernel-based SVMs can be fixed using the proposed method in Section III. Section IV is devoted to the multiscale Local Descriptor (MLD) technique for protein feature extractions and generating data from protein sequence-based structure. We present the numerical results for the proposed low-rank approximation for predicting PPI in Section V. We also discuss the advantages of the proposed method in comparison with the existing methods in this section. We evaluate the proposed methodology for predicting PPIs in "S. cerevisiae". Finally, some concluding remarks are given in Section VI.

II. PRELIMINARIES ON KERNEL-BASED SVM

We first provide a preliminary introduction to the kernel-based SVM technique. The main motivation behind using the kernel-based SVM for classifying PPIs is that it would not usually be possible to classify this data with the hyperplane decision boundary as resulted of employing the conventional SVM. An alternative way is to consider a feature space in place of the data itself or input space, and attempt to separate data in feature space by linear SVM or hyperplane. For example, the feature space can be considered as the distance of data in input space from each other or a function of this distance. It is evident from Figure 1, the feature space of the measurements that is denoted by $\phi_{\mathbf{x}}$, cannot be linearly separable within the input space, but they are separable in the feature space. Note that this feature space is potentially infinite-dimensional and therefore offers much more flexibility for separating data than finite dimensional input space. This remark has a theoretical foundation in the form of Cover's theorem [27], which ensures that data can not be separated by a hyperplane in input space while most likely will be linearly separable after being transformed into feature space by a suitable feature map. Thus, feature space based SVMs are viewed as proper techniques to resolve the classification challenges of the intricate data, in particular for prediction of PPIs.

Let us suppose the training dataset is given by $\mathcal{D} = \{(\mathbf{x}_i, y_i)|i = 1, \ldots, n\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ are the features, and the corresponding data values in the form of labels are given by $y_i \in \{-1, +1\}$. We denote $\mathbf{x}_i$ as the specific attributes for a given protein pair i. Furthermore, label y_i indicates whether the i-th protein pair interacts $(+1)$ or does not interact (-1). The kernel-based SVM will allow us to assign an appropriate label, either -1 or $+1$, to a future attributes for a given protein pair. It should be noted that a linear SVM classifies data by finding the best hyperplane, which separates all data points of one class from another one. However, the algorithms for non-linear classification are more or less similar to the linear one; but the measurements $\mathbf{x}_i$ in input space are simply replaced by their features $\phi_{\mathbf{x}_i}$ in the feature space.

The Dual problem for the kernel-based SVM for predicting PPIs using the transformed input data is given by

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \phi_{\mathbf{x}_i}^T\phi_{\mathbf{x}_j} - \sum_{i=1}^{n}\alpha_i$$

$$\text{s.t} \sum_{i=1}^{n}\alpha_i y_i = 0 \quad \text{and } 0 \leq \alpha_i \leq C, \tag{1}$$

where α_i are the Lagrange multipliers, and C is called box constraint (or penalty coefficient), which is a free parameter. When the given measurements are not perfectly separable the penalty coefficient C would enable us to resolve this situation. It is now possible to obtain feature space of the data using the framework of Reproducing Kernel Hilbert Space (RKHS), which considers the mapping of the introduced feature as $\phi : \Omega \longrightarrow H_K(\Omega)$ under the map

$$\mathbf{X} \longmapsto \phi_{\mathbf{X}} = K(., \mathbf{X}).$$

The map ϕ transfers $\mathbf{X}$ from input space Ω to feature space $H_K(\Omega)$, where $H_K(\Omega)$ is RKHS corresponded to the kernel $K(., .)$ [17]. The RKHS can be characterised within the inner product in the feature space, $H_K(\Omega)$, using the kernel K, as:

$$K(\mathbf{X}, \mathbf{Z}) = \phi_{\mathbf{X}}^T\phi_{\mathbf{Z}}. \tag{2}$$

Using Eq. (2), a more general form of the dual problem given in Eq. (1), can be defined, in the feature space, as follows:

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{n}\alpha_i,$$

$$\text{s.t} \sum_{i=1}^{n}\alpha_i y_i = 0 \quad \text{and,} \quad 0 \leq \alpha_i \leq C. \tag{3}$$

which is the modified version of dual problem for kernel-based SVM.

Given the crucial role of the kernels, the approach discussed above is named as a kernel-based SVM and provides the possibility of non-linear classification for the observed data. There are several features that make the kernel-based SVM more appealing. One of this features is their promising flexibility and efficiency of interacting with high-dimensional data, and the other one is their usefulness of encountering data that are non-linearly classified. These appealing properties of the kernel-based SVM can be achieved through some customisable parameters in the kernels (e.g., shape or scale parameter). Despite these advantages, this technique suffers from several shortcomings. One of the main drawbacks of this technique is that it will be computationally very expensive when it is used for classifying the data with many features (e.g., PPIs). This computational complexity is due to the presence of kernel in QOP in the structure of kernel-based SVMs. In fact, the corresponding Hessian matrix, of kernel-based SVM, in QOP will become very dense and ill-condition for data with many features. On the other hand, as far as the kernels in the SVMs structure has a high capacity for the classification of data with the nonlinear property, their computational challenges will mainly increase. In addition, the involved optimisation problem in the kernel-based SVMs will increase the computational cost and plays a major role in the training time and performance of this technique of classification. In this paper, we provide an efficient method for significantly reducing the computational complexity of the above optimisation problem of the kernel-based SVMs when they are used for classification of the data with many features. In the sequel, we will show how by using the low-rank approximation of the kernel, based on the Mercer series expansion of the kernel, it is possible to replace the QOP with a much simpler problem.

III. LOW-RANK APPROXIMATION FOR THE KERNEL-BASED SVMs

According to the Mercer's theorem [27], [29], each positive definite kernel can be expressed in terms of an infinite series expansion as follows:

$$K(\mathbf{x},\mathbf{z}) = \sum_{i=1}^{\infty} \lambda_i \varphi_i(\mathbf{x}) \varphi_i(\mathbf{z}). \tag{4}$$

In this expansion, λ_i and φ_i are positive eigenvalues and orthogonal eigenfunctions associated with the Hilbert-Schmidt operator of the kernel $K(.,.)$, respectively. It should be noted that the Hilbert Schmidt operator, $\kappa : L_2(\Omega) \longrightarrow L_2(\Omega)$, is defined as:

$$\kappa(f) = \int_{\Omega} K(\mathbf{x},\mathbf{z}) f(\mathbf{z}) d\mathbf{z},$$

such that the eigenvalues and eigenvectors used in the Mercer expansion can be obtained by solving the following eigenvalue problem:

$$(\kappa\varphi)(\mathbf{X}) = \lambda\varphi(\mathbf{X}).$$

Note that it would not be feasible to work with the infinite number of terms from the Mercer series to construct the low-rank approximation. However, given this fact that the eigenvalues of the Mercer expansion are naturally sorted in a decreasing order, the initial terms of this extension have a fundamental role in approximating the kernel $K(.,.)$. Therefore, we present the low-rank approximation of the kernel by truncating the series expansion given in Eq. (4) as illustrated in Eq. (5):

$$K(\mathbf{x},\mathbf{z}) = \sum_{k=1}^{m} \lambda_k \varphi_k(\mathbf{x}) \varphi_k(\mathbf{z}). \tag{5}$$

It is recommended to select m, the number of terms required to build the low-rank approximation given in Eq. (5), much smaller than n (the number of observed data). Furthermore, given $K(.,.)$ being a positive definite kernel with the series expansion given in Eq. (4), the truncated Mercer series based on m terms as suggested in Eq. (5), is the best m-term approximation from the perspective of least squares in $L_2(\Omega)$ for the kernel K [17]. Now, by taking into account these conditions, it is possible to present the matrix version of the approximation given in Eq. (5), satisfying in the dual problem Eq. (3), as

$$\mathbf{K} \simeq \boldsymbol{\phi}\boldsymbol{\Lambda}\boldsymbol{\phi}^T, \tag{6}$$

where a $n \times m$ matrix $\boldsymbol{\phi}$ is defined as

$$\boldsymbol{\phi} = \begin{bmatrix} \boldsymbol{\varphi}(\mathbf{X}_1)^T \\ \vdots \\ \boldsymbol{\varphi}(\mathbf{X}_n)^T \end{bmatrix},$$

$\boldsymbol{\varphi}(\mathbf{X})$ is defined as

$$\boldsymbol{\varphi}(\mathbf{X}) = \begin{bmatrix} \varphi_1(\mathbf{X}) \\ \vdots \\ \varphi_m(\mathbf{X}) \end{bmatrix},$$

and the diagonal matrix $\boldsymbol{\Lambda}_{m\times m}$ is also given by

$$\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_m \end{bmatrix}.$$

In other words, using the truncated Mercer series theorem, we approximate matrix $\mathbf{K}$ in the second-order dual optimisation problem defined in Eq. (3), using the expression given in Eq. (6). Since only the first m terms of the expansion are used, we use the symbol " $\simeq$" instead of " $=$". However, in many kernels for very small number of m, we can obtain very accurate approximations of the kernel $K(.,.)$, at which the symbol " $=$" can be used instead of " $\simeq$".

We now intend to reconstruct the dual optimisation problem described above in the matrix form, based on the derived approximation for matrix $\mathbf{K}$ as illustrated in Eq. (6). The original optimisation problem introduced in Eq. (3) can then be adapted in the matrix form as follows:

$$\min_{\boldsymbol{\alpha}} \left(\frac{1}{2}\boldsymbol{\alpha}^T \mathbf{D}_y \mathbf{K} \mathbf{D}_y \boldsymbol{\alpha} - \mathbf{e}^T\boldsymbol{\alpha} \right)$$
$$\text{Subject to } \mathbf{y}^T\boldsymbol{\alpha} = 0, \quad \text{and } \boldsymbol{\alpha} \in [0, C]^n,$$

where $\mathbf{D}_y$ is a diagonal matrix with elements $\{y_i\}_{i=1}^n$, on its main diagonal, and $\mathbf{e}$ contains a vector with elements 1. It is trivial to show that the approximated matrix $\mathbf{K}$ given in Eq. (6) can be also illustrated as:

$$\mathbf{K} = \left(\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{\phi}\right)^T \left(\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{\phi}\right).$$

Now, let us define $\mathbf{V} = \mathbf{D}_y\boldsymbol{\phi}\boldsymbol{\Lambda}^{\frac{1}{2}}$ and consequently $\mathbf{V}\mathbf{V}^T = \mathbf{D}_y\mathbf{K}\mathbf{D}_y$, the dual optimisation problem, given in Eq. (3), can then be rewritten as follows:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2} \left((\mathbf{V}^T\boldsymbol{\alpha})^T (\mathbf{V}^T\boldsymbol{\alpha}) - \mathbf{e}^T\boldsymbol{\alpha} \right)$$

$$\text{Subject to } \mathbf{y}^T\boldsymbol{\alpha} = 0, \quad \text{and } \boldsymbol{\alpha} \in [0, C]^n.$$

The matrix $\mathbf{V}^T\boldsymbol{\alpha}$, which contains the unknown vector $\boldsymbol{\alpha}$ can be also written as

$$\mathbf{V}^T\boldsymbol{\alpha} = \mathbf{I}_m\boldsymbol{\beta},$$

where $\mathbf{I}_m$ is an identity matrix of the order m and $\boldsymbol{\beta} \in \mathbb{R}^m$ as an arbitrary vector. With these considerations, the second-order dual optimisation problem presented in Eq. (3) can be then rewritten as follows:

$$\min_{\boldsymbol{\alpha},\,\boldsymbol{\beta}} \frac{1}{2}\begin{pmatrix}\boldsymbol{\beta}^T & \boldsymbol{\alpha}^T\end{pmatrix}\begin{pmatrix}\mathbf{I}_m & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}\begin{pmatrix}\boldsymbol{\beta}\\ \boldsymbol{\alpha}\end{pmatrix} - \begin{pmatrix}\mathbf{0} & \mathbf{e}^T\end{pmatrix}\begin{pmatrix}\boldsymbol{\beta}\\ \boldsymbol{\alpha}\end{pmatrix}$$

$$\text{Subject to } \begin{pmatrix}\mathbf{0} & \mathbf{y}^T\\ \mathbf{I}_M & \mathbf{V}^T\end{pmatrix}\begin{pmatrix}\boldsymbol{\beta}\\ \boldsymbol{\alpha}\end{pmatrix} = 0,\ \boldsymbol{\beta} \in \mathbb{R}^m, \quad \text{and}$$
$$\boldsymbol{\alpha} \in [0, C]^n.$$

It is noteworthy that, although the system of equations is of a higher order of $m + n$ (while the main system of equations is of the order n), but due to the much simpler structure, the computational cost of solving this system is far cheaper than the original system in Eq. (3). In addition, the Hessian matrix obtained in the new structure will also become very sparse, while the Hessian matrix for the kernel $K(.,.)$ is quite dense in the original form, which was so-called the full-rank form. Therefore, in the new structure, the vector computations and matrix analyses required for solving the second-order optimisation problem will be far less complex. Both of these changes would significantly reduce the required computational time for classification using the kernel-based SVM for complex data with many features. In the sequel, we pursue the obtained results to predict PPIs.

IV. PROTEIN FEATURE EXTRACTIONS

One of the most important steps in predicting proteins is the extraction of suitable properties from the amino acid sequence. Some feature extractions that succeeded in representing variable lengths of protein sequences are amino-acid composition (AAC) [29], dipeptide composition (DC) [29], tripeptide composition (TC) [30], pseudo-amino-acid composition (PseAAC) [31], and autocovariance (AC) [12]. Each of these techniques has its own disadvantages and advantages which a complete description of them is given in [14]. In this study, multiscale Local Descriptor (MLD) ([14]) feature representation scheme is used to extract features from a protein sequence. This scheme can capture multiscale local information by varying the length of protein-sequence segments. The MLD feature representation scheme facilitates the mining of interaction information from multiscale continuous amino acid segments, making it easier to capture multiple overlapping continuous binding patterns within a protein sequence. In fact, the MLD transforms the protein sequences into feature vectors by using a binary coding scheme. A protein sequence is transformed into groups based on the dipoles and side-chain volumes. The entire sequence is then divided into multiple sequence segments of varying lengths to describe local regions. In MLD, the protein sequence is divided into four equal-length segments ($S1$, $S2$, $S3$, and $S4$), following which 16 different combinations are derived using a 4-bit binary coding scheme. For example, 1100 refers to the continuous region constructed by $S1$ and $S2$. In MLD, only nine continuous sub-sequences are considered: 0001, 0010, 0011, 0100, 0110, 0111, 1000, 1100, and 1110. For each sub-sequence, the local descriptors Composition, Transition, and Distribution (CTD) ([32]) are calculated and concatenated. In CTD, the sequence is represented by seven groups of amino acids, which is the same as TC. Composition calculates the frequency of each group, Transition characterizes the frequency with which amino acids in one group are followed by amino acids in another group, and Distribution measures the location of the first, 25%, 50%, 75%, and 100% of the amino acids in the group. For example, the sub-sequence

$$AGCMTYCCACCCASYAGCCGYG,$$

would be transformed into

$$1123332212221331122131,$$

according to the amino-acid classification. The composition is 36.36% (= 8/22) for "1," 36.36% (= 8/22) for "2," and 27.27% (= 6/22) for "3." There are three types of transitions in this transformed sequence, giving a Transition of 28.57% (= 6/21) for "1" to "2" or "2" to "1"; 19% (= 4/21) for "1" to "3" or "3" to "1"; and 9.52% (= 2/21) for "2" to "3" or "3" to "2." In terms of Distribution, eight residues are represented by "1", the rankings of which at the first, 25%, 50%, 75%, and 100% of occurrences are 1st, 2nd (= 8 × 25%), 4th (= 8 × 50%), 6th (= 8 × 75%), and 8th (= 8 × 100%). The locations of "1" at the 1st, 2nd, 4th, 6th, and 8th positions in this sequence are 1, 2, 13, 17, and 22, respectively. Hence, the Distributions for "1" are 4.55% (= 1/22), 9.09% (= 2/22), 59.09% (= 13/22), 77.27% (= 17/22), and 100% (= 22/22). Similarly, the Distributions for "2" and "3" are 13.64%, 31.82%, 45.45%, 54.55%, and 86.36%; and 18.18%, 18.18%, 27.27%, 63.64%, and 95.45%, respectively. For each continuous region, CTD generates a 63-dimensional vector: 7 for composition, $21(= 7 \times [6/2])$ for Transition, and $35(= 7 \times 5)$ for Distribution. Nine sub-sequences are then calculated and concatenated for a $567(= 63 \times 9)$-dimensional feature vector. Finally, the PPI pair is characterized by concatenating the two vector spaces of two individual proteins. Thus, a very high-dimension vector of size "1134" has been constructed to represent each protein pair and used as a feature vector for input into SVM classifier.

Here, the PPI dataset which were derived by Guo et al. [12], are used to build the first prediction model. The dataset was downloaded from "S. Cerevisiae" core subset of database of interacting proteins (DIP) [33]. After the protein pairs that contain a protein with fewer than 50 residues or have more than 40 percent sequence identity were removed, the remaining 5594 protein pairs would form the golden standard positive dataset (GSP). The construction of a negative PPI dataset is very important for training and evaluating the prediction model. However, it is difficult to generate such a dataset because limited information about proteins

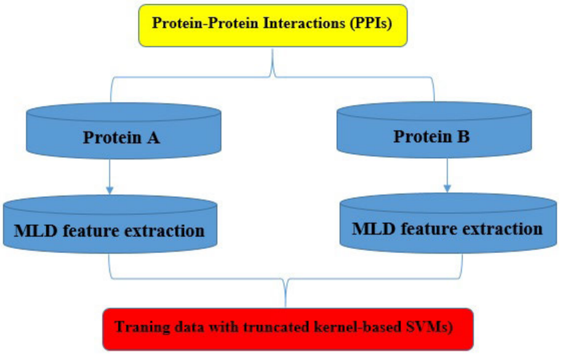

Protein-Protein Interactions (PPIs)
Protein A
Protein B
MLD feature extraction
MLD feature extraction
Traning data with truncated kernel-based SVMs)

FIGURE 2. Flowchart for constructing the protein-protein interactions.

that are really non-interactive exists. Here, the negative dataset is generated by first selecting non-interacting pairs, uniformly at random from the set of all proteins pairs, which are not known to be interacted. The protein pairs with same subcellular localisation information must be then excluded. Finally, the remaining 5594 protein pairs whose subcellular localisation is different will constitute the golden standard negative (GSN) dataset. By combining the above GSP and GSN datasets, the complete dataset consists of 11188 protein pairs, where half of them are from the positive dataset and the other half are from the negative dataset. A flowchart for construction of protein-protein interactions using presented technique in this paper is presented in Figure 2. It should be noted that, in this paper, we have used exactly the same PPI dataset as used in Guo et al [12]. The names of protein pairs and their sequences of the dataset are given in online supplementary material, which is available at https://sites.google.com/site/zhuhongyou/data-sharing.

V. RESULTS

In this section, in order to compare the predictive and computational performance of the full-rank kernel-based SVM against the corresponded low-rank for predicting PPIs, we first need to adjust the model parameters based on the algorithm presented below.

The prediction of PPI, using full and low-rank approximations for the kernel-based SVM, heavily depends on the parameters C (box constraint) and ϵ (Gaussian shape parameter). If the number of available features for each pair of proteins was less than or equal to three, the decision contours would be good way to show the impact of the various Gaussian kernel parametrisations on the low-rank approximation of the SVM. Unfortunately, because of the very large number of features per pair of available proteins (i.e., 1134) presenting by contour plot is not possible. Clearly, the choice of ϵ plays a significant role in classification performance, where larger ϵ encourages an SVM with more locality and smaller ϵ encourages less localized influence; this matches the standard localization behaviour for Gaussian kernel in an interpolation setting. A similar impact can occur for different C values such that smaller C values produce a less active decision contour, whereas large C encourages more local fluctuations.

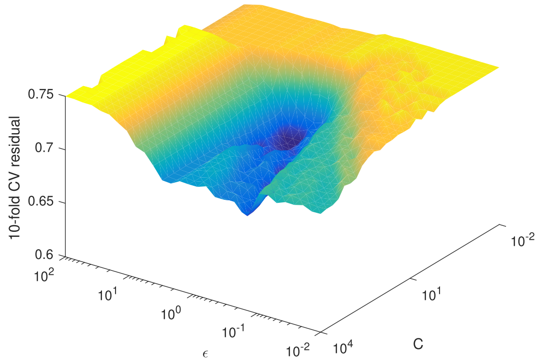
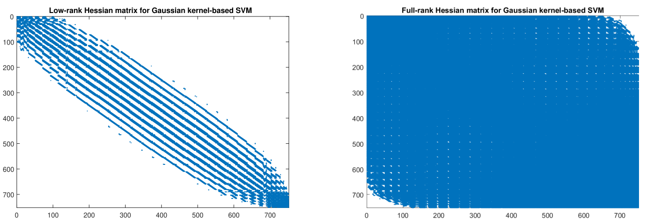


FIGURE 3. Surface plot allowing both ϵ and C to vary.

To determine optimal values for the parameters of the SVM, a more common technique in the machine learning community is to use k-fold cross-validation (CV) [36]. Here, a 10-fold CV scheme is used to measure the effectiveness of each of ϵ and C parameters. Figure 3 shows the computed CV residuals in terms of various values of ϵ and C. The CV residuals in this plot were computed by dividing the PPIs data into training and testing data, where 70% of the available pairs of the proteins (almost 7832) were selected as training data and remaining 30% (almost 3356) as testing data. It is clear that there is an optimal region for the 10-fold CV residuals, where decreases in ϵ are matched by increases in C. Finally, from Figure 3, "0.01" and "166.81" are chosen as the optimal values for ϵ and C, respectively. For these optimal values, the full-rank SVM and its low-rank approximation are compared against each other in terms of various metrics, as reported in Table 1. Although the computation time for the low-rank approximation has been dramatically reduced, but no significant changes in the classification performance measurements, including accuracy, precision, specificity, and sensitivity, can be observed. As a result, it can be concluded that there is no significant differences in the classification accuracy of these two approximation, but the computational time of classifying the PPIs data using the low-rank approximation is at least 78 times faster than the full-rank SVM. Note that the results illustrated in Table 1 are computed by MATLAB software on a High performance computing (HPC) with a dual 16-core 2.4GHz Intel CPU's with 64 GB RAM processor. Note that Matlab code is added as a Low-rank-SVM.zip file. To implement low rank SVM and compare with full rank and run these files first one should run rbfsetup.m and the main.m file. In Figure (4), the Hessian matrix for low-rank (a) and full-rank (b) Gaussian kernel-based SVM is added. Also, the condition number (CN) that measures sparsity for for Full rank SVM is 1.3153×10^6 while this amount for the corresponded low-rank is 3.1850×10^3. These results point out essentially that low-rank Hessian matrix is sparser and well-conditioned than Hessian matrix of the full-rank method. Also, the low-rank Hessian matrix can be swiftly implemented with considerably less computational cost (In minutes).



(a) (b)

FIGURE 4. Hessian matrix for low-rank (a) and full-rank (b) Gaussian kernel-based SVM.

TABLE 1. Comparison between the low-rank approximation and the full-rank SVM when $\epsilon = 0.01$, $C = 166.81$ and $m = 0.1n$.

	CPU time	Accuracy	Precision	Specificity	Sensitivity
Full-rank	2541.6125	0.8653	0.8870	0.8685	0.8627
Low-rank	32.2706	0.8483	0.8753	0.8580	0.8402

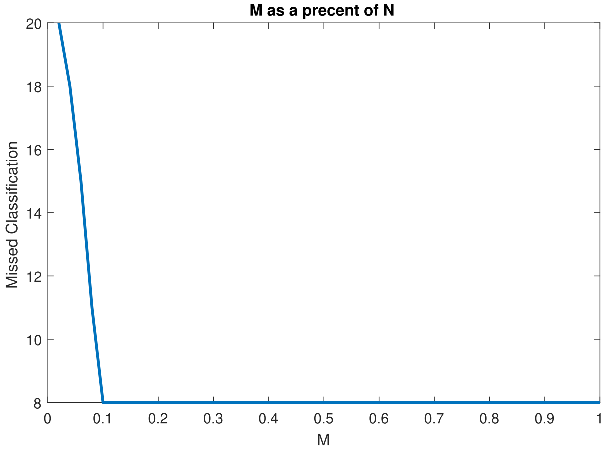


FIGURE 5. Number of missed classification items in terms of M as a percent of N.

TABLE 2. Comparison between the low-rank approximation and the full-rank SVM for different values of ϵ and when $C = 1$ and $m = 0.1n$.

ϵ	CPU time		Accuracy		Precision		Specificity		Sensitivity	
	Full	Low	Full	Low	Full	Low	Full	Low	Full	Low
0.01	136.0172	12.0606	0.5042	0.4994	0.5021	0.4997	0.0328	0.0155	0.9756	0.9833
0.1	128.1611	11.1133	0.6275	0.5238	0.6164	0.5167	0.5799	0.3105	0.6752	0.7372
1	124.0402	13.1116	0.5831	0.5861	0.5553	0.5579	0.3319	0.3427	0.8343	0.8296
5	118.0657	9.1239	0.5	0.5	0.5	0.5	0	0	1	1

TABLE 3. Comparison between the full-rank SVM and its the low-rank approximation for different values of C when $\epsilon = 0.1$ and $m = 0.1n$.

C	CPU time		Accuracy		Precision		Specificity		Sensitivity	
	Full	Low	Full	Low	Full	Low	Full	Low	Full	Low
1	120.2418	11.6134	0.6275	0.5238	0.6164	0.5167	0.3105	0.6275	0.7372	0.6654
10	169.5712	16.1121	0.6326	0.5992	0.6185	0.5990	0.5733	0.5983	0.6919	0.6001
100	275.6133	19.6543	0.6332	0.6111	0.6071	0.6202	0.5113	0.6490	0.7551	0.5733
1000	489.7655	23.2144	0.6377	0.5590	0.6088	0.5610	0.5048	0.5757	0.7706	0.5423
10000	789.6550	33.3001	0.6401	0.6311	0.6208	0.5911	0.5822	0.7812	0.7933	0.6877

Without considering the optimal values for the parameters, it is necessary to compare the classification performance metrics of the full-rank SVM with its correspond low-rank approximation, for the different parameter values of ϵ and C. Table 2 reports such a comparison for the different values of ϵ when $C = 1$ and $m = 0.01n$. We first discuss how the number of terms, m, required to construct the low-rank approximation, can be determined. As can be seen from the results of this table, as ϵ increases, although the classification performance metrics (including accuracy, precision, specificity and sensitivity) do not noticeably change, but the CPU time for the low-rank approximation will be considerably reduced (by average at least 12 times faster than the full-rank SVM).

Table 3 illustrates the results of comparing the classification performance metrics of the full-rank SVM and its low-rank approximation for the different values of C when $\epsilon = 0.1$ and $m = 0.1n$. As it can be deduced from the results reported in Table 3, by increasing the parameter C, the CPU time will also increase for the both models. However, increasing in the CPU time for the full-rank SVM is significantly (from 10 to 23 times) higher than the low-rank SVM. However, increasing in C will improve the classification performance metrics, but no noticeable differences between the computed metrics for these two models can be observed.

After finding the optimal values for ϵ and C using 10-fold CV, one of the remaining challenges would be determining m, the minimum terms required to construct the low-rank approximation (given in Eq. (5)) as accurate as desire. Since the accuracy of the kernel approximation and ultimately the low-rank kernel-based SVM are greatly dependent on truncating the Mercer's expansion at an appropriate cutting point, it is thus necessary to develop a criterion to effectively determine an appropriate truncating point, so-called m-value, required to construct the low-rank approximation. An appropriate choice of m-value would also lead to enhance accuracy of the classification performance metrics as mentioned above (i.e., accuracy, precision, specificity, and sensitivity). One could adapt a truncation criterion by only considering the magnitude of the eigenvalues, as suggested in [34] to approximate the radial basis functions. This truncation scheme is somewhat not suitable for the kernel-based SVM, because it requires the actual construction of the eigenfunctions to make a decision on the optimal value of m, while the entire construction process cannot be planned in advance. This truncating criterion would not be straightforward in practice.

An alternative method, which is based on analysing the truncation lengths of the kernels, especially Gaussian kernel, is proposed in [35]. It was shown that the truncation length should be chosen in accordance to n. It can be illustrated if the truncation length is determined according to this criterion, then training of the kernel-based SVM with the truncated kernel will have the same approximation order as with the full kernel. In other words, in order to determine an optimal value for m, we need to use an approach which is dependent on the total sample, n. As illustrated in Table 1, the classification performance metrics (0.8653, 0.8870, 0.8685 and 0.8627, associated to accuracy, precision, specificity and sensitivity, respectively) computed based on the full-rank SVM

TABLE 4. Computed classification performance metrics and CPU time for the low-rank approximation with different m values, but setting $\epsilon = 0.01$, and $C = 168.81$.

m	CPU time	Accuracy	Precision	Specificity	Sensitivity
$0.01n$	23.1109	0.6397	0.6465	0.6220	0.6566
$0.05n$	29.5468	0.7291	0.7609	0.7277	0.7304
$0.1n$	32.2706	0.8483	0.8753	0.8580	0.8402
$0.3n$	101.5689	0.8511	0.8793	0.8598	0.8456
$0.5n$	980.7875	0.8567	0.8806	0.8621	0.8498
$0.75n$	1679.2109	0.8566	0.8808	0.8645	0.8508
$0.9n$	2332.1245	0.8633	0.8810	0.8661	0.8511

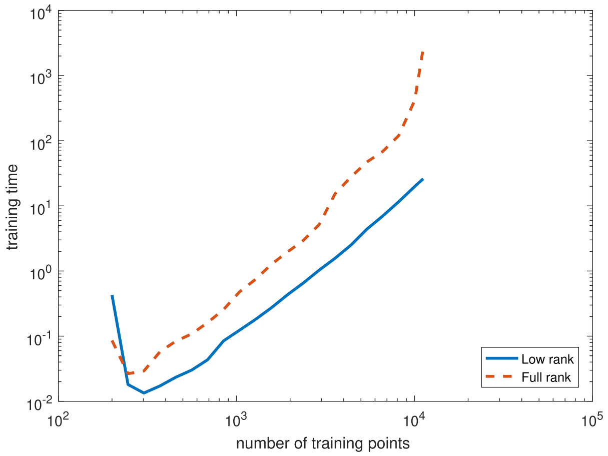


FIGURE 6. Training time in terms of the number of points in the low-rank and full-rank kernel-based SVM.

are almost the same to the ones computed by employing the low-rank approximation. However, the computational cost of the low-rank model is significantly lower than the full-rank. A more detailed study of comparing the classification performance metrics and CPU time for low-rank approximation with different m values, $\epsilon = 0.01$ and $C = 168.81$ is presented in Table 4. We can see that there is a jump in the values of classification performance metrics by setting $m = 0.1n$ (Fig. 5), and after that, no change was made. In addition, for the low-rank model with $m = 0.1n$, although there is negligible changes in the performance of this model compared to the full-rank, but the computation time, using the low-rank model, is significantly reduced. From now on, this strategy will by setting the m-value at "$m = 0.1n$".

After studying the classification performance of the low-rank kernel-based SVM based on the changes in the parameters as discussed above, we will now examine the capability of employing this new model structure in reducing training time in classification of the large-scale data in comparison with the full-rank method. Furthermore, the quadratic optimisation equation for the low-rank approximation of the SVM will be highly straightforward to solve. As a result, this will significantly reduce the training time in classifying the large-scale data, using the low-rank approximation for the kernel-based SVM. The relationship between the training time (in seconds) and the size of training data for the both SVM models (i.e., low-rank and full-rank) is illustrated in Figure 6. In order to draw this plot for the PPI dataset, different numbers of pair of proteins are considered, varying from 200 to 11188. For the small number of pairs of protein, there is no significant difference between the training time of the two models, although training time using the low-rank approximation is slightly lower than the one based on the full-rank model. However, a significant difference in training time between these two models can be observed for the large numbers of pair of proteins. It is very evident to observe that the time required to train the low-rank SVM model for the large number of proteins pairs is significantly lower, which suggests the performance of this model by taking into account achieving the same accuracy performance (in comparison to the full-rank model) is considerably better. For example, for the 11188 pair of proteins, the training time using the full-rank model is 2635.3386 (seconds), while this time using the low-rank approximation reduced to 26.0913 (seconds). This training time for the low-rank SVM is over 100 times faster than the Full-rank SVM. This comparison was implemented by setting $\epsilon = 0.01$, $C = 168.81$, and $m = 0.1n$.

VI. CONCLUSION

In this paper, we introduced a novel low-rank approximation for the kernel-based SVM in order to predicting protein-protein interactions in "S. Cerevisiae" based on the truncated Mercer series expansion of the underlying kernel. In this methodology, the computational challenges due to the complex QOP associated with the standard kernel-based SVM are fixed by replacing the QOP with a much simpler and computationally efficient optimisation problem. This would significantly reduce the training time required for the classification using the approximated kernel-based SVM. The numerical results reveal significant reduction in computational time for predicting PPIs without losing noticeable accuracy and sensible changes to the other classification performance metrics. However, for a small number of training pair of proteins, this difference in computational time was negligible, but when the training number of proteins pairs increased, a significant difference between the computational performance of the two models was observed which suggests the overall performance of the low-rank SVM is considerably better. In other words, it can be concluded that the approximated kernel-based SVM outperforms the full-rank kernel-based by achieving the same classification accuracy rates, but by significantly reducing the computational cost (over 100 times faster in training the classifier) for classification/prediction of a dataset with many features like protein-protein interactions. Finally, the introduced technique in this paper can applied for reducing computational time in classification using kernel-based SVMs for another dataset like promoter recognition in DNA sequences.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number IF-2020-NBU-412.

REFERENCES

[1] P. Uetz, L. Giot, G. Cagney, T. A. Mansfield, R. S. Judson, J. R. Knight, D. Lockshon, V. Narayan, M. Srinivasan, and P. Pochart, ''A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae*,'' *Nature*, vol. 403, no. 6770, pp. 623–627, 2000.

[2] N. J. Krogan, G. Cagney, H. Y. Yu, G. Q. Zhong, X. H. Guo, and A. Ignatchenko, ''Global landscape of protein complexes in the yeast *Saccharomyces cerevisiae*,'' *Nature*, vol. 440, pp. 637–643, Mar. 2006.

[3] A. C. Gavin, M. Bosche, R. Krause, P. Grandi, M. Marzioch, and A. Bauer, ''Functional organization of the yeast proteome by systematic analysis of protein complexes,'' *Nature*, vol. 415, pp. 141–147, Jan. 2002.

[4] Y. Ho, A. Gruhler, A. Heilbut, G. D. Bader, L. Moore, and S. L. Adams, ''Systematic identification of protein complexes in *Saccharomyces cerevisiae* by mass spectrometry,'' *Nature*, vol. 415, pp. 180–183, Jan. 2002.

[5] H. Zhu and M. Snyder, ''Protein chip technology,'' *Current Opinion Chem. Biol.*, vol. 7, no. 1, pp. 55–63, Feb. 2003.

[6] J. R. Bock and D. A. Gough, ''Predicting protein–protein interactions from primary structure,'' vol. 17, pp. 60–455, May 2001.

[7] M. Pellegrini, E. M. Marcotte, M. J. Thompson, D. Eisenberg, and T. O. Yeates, ''Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles,'' *Proc. Nat. Acad. Sci. USA*, vol. 96, no. 8, pp. 4285–4288, Apr. 1999.

[8] A. J. Enright, I. Iliopoulos, N. C. Kyrpides, and C. A. Ouzounis, ''Protein interaction maps for complete genomes based on gene fusion events,'' *Nature*, vol. 402, no. 6757, pp. 86–90, Nov. 1999.

[9] R. Singh, D. Park, J. Xu, R. Hosur, and B. Berger, ''Struct2Net: A web service to predict protein-protein interactions using a structure-based approach,'' *Nucleic Acids Res.*, vol. 38, pp. W508–W515, Jul. 2010.

[10] M. Ohue, Y. Matsuzaki, N. Uchikoga, T. Ishida, and Y. Akiyama, ''MEGADOCK: An all-to-all protein-protein interaction prediction system using tertiary structure data,'' *Protein Peptide Lett.*, vol. 21, no. 8, pp. 766–778, 2014.

[11] J. Shen, J. Zhang, X. Luo, and H. Jiang, ''Predicting protein–protein interactions based only on sequences information,'' *Proc. Nat. Acad. Sci. USA*, vol. 104, pp. 4337–4341, Mar. 2007.

[12] Y. Guo, L. Yu, Z. Wen, and M. Li, ''Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences,'' *Nucleic Acids Res.*, vol. 36, no. 9, pp. 3025–3030, May 2008.

[13] Y. C. T. Wei, K. Sumikoshi, S. T. N. Terada, K. Kadota, and K. Shimizu, ''Prediction protein–protein interactions using sequence homology and machine learning methods, research,'' *Res. J. Life Sci., Bioinf., Pharmaceutical Chem. Sci.*, vol. 3, pp. 1–26, Sep. 2017.

[14] Z.-H. You, K. C. C. Chan, and P. Hu, ''Predicting protein–protein interactions from primary protein sequences using a novel multi-scale local feature representation scheme and the random forest,'' *PLoS ONE*, vol. 10, no. 5, May 2015, Art. no. e0125811.

[15] V. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag, 2013.

[16] S. Martin, D. Roe, and J.-L. Faulon, ''Predicting protein–protein interactions using signature products,'' *Bioinformatics*, vol. 21, no. 2, pp. 218–226, Jan. 2005.

[17] G. Fasshauer and M. McCourt, *Kernel-Based Approximation Method Using Matlab*. Singapore: World Scientific, 2016.

[18] L. Lan, Z. Wang, S. Zhe, W. Cheng, J. Wang, and K. Zhang, ''Scaling up kernel SVM on limited resources: A low-rank linearization approach,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 2, pp. 369–378, Feb. 2019.

[19] I. Steinwart and A. Christmann, ''Support vector machines,'' in *Information Sciences and Statistics*. Springer-Verlag, New York, 2008.

[20] B. Scholkopf and A. J. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, U.K.: MIT Press, 2002.

[21] L. Chih-Jen, ''Support vector machines and kernel methods: Status and challenges,'' Talk K. U. Leuven Optim. Eng. Center, Dept. Comput. Sci., Nat. Taiwan Univ., New Taipei, Taiwan, 2013.

[22] X. Nguyen, L. Huang, and A. D. Joseph, *Support Vector Machines, Data Reduction, and Approximate Kernel Matrices*. Berlin, Germany: Springer-Verlag, 2008.

[23] F. Bach, ''Sharp analysis of low-rank kernel matrix approximations,'' in *Proc. 26th Annu. Conf. Learn. Theory*, vol. 30, pp. 1–25, 2013.

[24] M. S. Andersen and L. Vandenberghe, ''Support vector machine training using matrix completion techniques,'' *J. Mach. Learn.*, vol. 12, pp. 22–41, Mar. 2014.

[25] M. Esmaeilbeigi, O. Chatrabgoun, A. Hosseinian-Far, R. Montasari, and A. Daneshkhah, ''A low cost and highly accurate technique for big data spatial–temporal interpolation,'' *Appl. Numer. Math.*, vol. 153, pp. 492–502, Jul. 2020.

[26] L. A. Sosa, ''An overview of non-linear kernel functions for solving the human face recognition problem,'' Doctoral dissertation, Univ. California UCLA, Los Angeles, CA, USA, 2016.

[27] T. M. Cover, ''Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition,'' *IEEE Trans. Electron. Comput.*, vol. EC-14, no. 3, pp. 326–334, Jun. 1965.

[28] T. Poggio, R. Rifkin, S. Mukherjee, and P. Niyogi, ''General conditions for predictivity in learning theory,'' *Nature*, vol. 428, no. 6981, p. 378, 2004.

[29] C. Yang, R. Duraiswami, and L. S. Davis, ''Efficient kernel machines using the improved fast Gauss transform,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 17, pp. 1561–1568, 2004.

[30] M. Bhasin and G. P. S. Raghava, ''Classification of nuclear receptors based on amino acid composition and dipeptide composition,'' *J. Biol. Chem.*, vol. 279, no. 22, pp. 23262–23266, May 2004.

[31] L. Liu, Y. Cai, W. Lu, K. Feng, C. Peng, and B. Niu, ''Prediction of protein–protein interactions based on PseAA composition and hybrid feature selection,'' *Biochem. Biophysical Res. Commun.*, vol. 380, no. 2, pp. 318–322, Mar. 2009.

[32] Y. Zhou, Y. Gao, and Y. Zheng, ''Prediction of protein-protein interactions using local description of amino acid sequence,'' in *Advances in Computer Science and Education Applications*, vol. 202. Berlin, Germany: Springer, 2011, pp. 254–262.

[33] UniProt Consortium, ''UniProt: A hub for protein information,'' in *Nucleic Acids Research*. Oxford, U.K.: Oxford Univ. Press, 2014, doi: 10.1093/nar/gku989.

[34] B. Fornberg and C. Piret, ''A stable algorithm for flat radial basis functions on a sphere,'' *SIAM J. Sci. Comput.*, vol. 30, no. 1, pp. 60–80, Jan. 2008.

[35] M. Griebel, C. Rieger, and B. Zwicknagl, ''Multiscale approximation and reproducing kernel Hilbert space methods,'' *SIAM J. Numer. Anal.*, vol. 53, no. 2, pp. 852–873, Jan. 2015.

[36] M. Kuhn and K. Johnson, *Feature Engineering and Selection: A Practical Approach for Predictive Models*. Boca Raton, FL, USA: CRC Press, 2019.

[37] M. Raissi, P. Perdikaris, and G. E. Karniadakis, ''Machine learning of linear differential equations using Gaussian processes,'' *J. Comput. Phys.*, vol. 348, pp. 683–693, Nov. 2017.

[38] B. Fornberg, E. Larsson, and N. Flyer, ''Stable computations with Gaussian radial basis functions,'' *SIAM J. Sci. Comput.*, vol. 33, no. 2, pp. 869–892, Jan. 2011.

[39] M. Pazouki and R. Schaback, ''Bases for kernel-based spaces,'' *J. Comput. Appl. Math.*, vol. 236, no. 4, pp. 575–588, Sep. 2011.

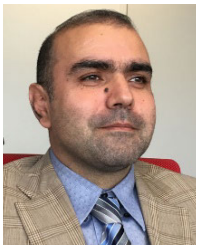
OMID CHATRABGOUN received the Ph.D. degree in mathematical statistics from Shahid Chamran University, Iran, in 2015, for a thesis entitled ''Estimating Copula Density Function Through Multiwavelets.'' Following that, he continued learning about data science and mostly focused on copula-based supervised and unsupervised learning. During his academic career, he has taught some courses relating to his expertise at graduate and undergraduate levels, including statistical learning and machine learning. He is currently an Assistant Professor of data science and AI with Coventry University. He is the coauthor of many research outputs, published as peer-reviewed papers, book chapters, and conference presentations. He has experience of gaining research grants and has been involved in consultancy and knowledge exchange projects with a range of research and industrial centers, including the Iran National Science Foundation (INSF) and the Research Institute for Grapes and Raisin (RIGR). His primary research interests include data science using (deep) Gaussian process and kernel-based support vector machines (SVMs), stochastic optimization, and machine learning (especially (deep) Gaussian process, neural networks, and stochastic differential equation).

ALIREZA DANESHKHAH received the Ph.D. degree from the University of Warwick for a thesis titled "Estimation in causal graphical models."

He was a member of the Warwick Centre for Predictive Modelling where he has developed deep learning methods to probabilistically simulate highly complex systems. He also worked as the Course Director of utility asset management at the Water Institute, Cranfield University. He is currently an Associate Professor and the Curriculum Lead of Data Science and AI. He is also an Associate of the Coventry Research Centre for Computational Science and Mathematical Modelling (CSMM). He has served as a Principal Investigator, a Collaborator, and a Researcher for several EPSRC, NHS, NERC, DEFRA, and industrial-based research projects in developing various Bayesian machine learning methods in tackling highly complex engineering and environmental case studies in the presence of both limited and big data. He is the coauthor of three books in expert judgment, advanced reliability methods, and digital twins and has an established list of published journal articles, book chapters, and conference communications to his name. His primary research interests include Bayesian elicitation of expert's probabilistic statements and model structure, modeling high-dimensional data using Bayesian networks, dynamic Bayesian networks, pair-copula Bayesian network models, and simulating highly complex engineering and environmental systems using Gaussian process emulators and deep learning approaches. He has applied these methods to a wide range of applications, including urban and coastal flood modeling, health, economics, decision-making under uncertainty, and risk assessment of networked systems.

Dr. Daneshkhah is a fellow of the Royal Statistical Society, a member of the International Society of Bayesian Analysis, and an Associate Member of the Institute of Mathematics and its Applications.

MOHSEN ESMAEILBEIGI is currently an Associate Professor with the Department of Mathematics, Malayer University, Malayer, Iran. His primary research interests include numerical analysis, radial basis function, and partial and ordinary differential equations.



NADER SOHRABI SAFA has five years postdoctoral experience in cyber security with Nelson Mandela Metropolitan University, South Africa, and Cyber Security Centre, WMG, University of Warwick. He was a Lecturer with Coventry University before joining the University of Wolverhampton. He is currently a Senior Lecturer in cyber security with the School of Mathematics and Computer Science, Faculty of Engineering, University of Wolverhampton. He is a member of the Cyber Security Research Institute. His academic experience and 15 years industry experience in the field of computer science and cyber security have provided a strong background for his work. He has participated in several large scale projects, such as PETRAS, Data Protection in Industry, Human Aspects of Information Security, and so on. He is a member of the editorial board and a reviewer in several well-known journals in the domain of cyber security. He is also a PC member and an organizer of several international conferences.

ALI H. ALENEZI received the B.S. degree in electrical engineering from King Saud University, Saudi Arabia, the M.S. degree in electrical engineering from the Royal Institute of Technology KTH, Sweden, and the Ph.D. degree in electrical engineering from the New Jersey Institute of Technology, USA, in 2018. He is currently an Assistant Professor with the Department of Electrical Engineering, Northern Border University, Saudi Arabia. His research interests include acoustic communication, wireless communications, and 4G and 5G networks using UAVs.

ARAFATUR RAHMAN received the Ph.D. degree in electronic and telecommunications engineering from the University of Naples Federico II, Naples, Italy, in 2013. He has more than ten years research and teaching experience in the domain of computer and communications engineering. He was an Associate Professor with the Faculty of Computing, Universiti Malaysia Pahang, where he had conducted undergraduate and master's courses and supervised more than 21 B.Sc., five M.Sc., and five Ph.D. students. He worked as a Postdoctoral Research Fellow with the University of Naples Federico II, in 2014, and a Visiting Researcher with the Sapienza University of Rome, in 2016. He is currently a Senior Lecturer with the School of Mathematics and Computer Science, University of Wolverhampton, U.K. He has developed an excellent track record of academic leadership as well as management and execution of international ICT projects that are supported by agencies in the U.K., Italy, EU, and Malaysia. He has coauthored of around 100 prestigious IEEE and Elsevier journals, such as IEEE Transactions on Industrial Informatics, IEEE Transactions on Services Computing, *IEEE Communications Magazine*, *Journal of Network and Computer Applications* (JNCA) (Elsevier), and *Future Generation Computer Systems* (FGCS) (Elsevier), and conference publications, such as IEEE GLOBECOM and IEEE DASC. His research interests include the Internet of Things (IoT), wireless communication networks, cognitive radio networks, 5G, vehicular communication, cyber physical systems, big data, cloud-fog-edge computing, and machine learning and security. He has received number of prestigious international research awards, notably the Best Paper Award at ICNS 2015, Italy; IC0902 Grant, France; Italian Government Ph.D. Research Scholarship and IIUM Best Masters Student Award; Best Supervisor Award at UMP; Awards in International Exhibitions, including Best Innovation Award, MTE 2020, Malaysia, Diamond and Gold in BiS 2017, U.K., the Best of the Best Innovation Award and Most Commercial IT Innovation Award, Malaysia, and Gold and Silver medals in iENA 2017, Germany. He was a Petron, the General Chair, an Organizing Committee, the Publicity Chair, the Session Chair, the Programme Committee, and a member of Technical Programme Committee (TPC) in numerous leading conferences worldwide, such as IEEE GLOBECOM, IEEE DASC, and IEEE iSCI. He has served as an advisory board member, an Editor for*Computers* (MDPI), a Lead Guest Editor for IEEE Access and *Computers*, and an Associate Editor for IEEE Access.