Send Orders for Reprints to reprints@benthamscience.ae

The Open Bioinformatics Journal, **2015,** *9,* **1-12**

Open Access

Protein-Protein Interaction Prediction using PCA and SVR-PHCS

Saeideh Mahmoudian, Abdulaziz Yousef and Nasrollah Moghadam Charkari*

Electrical and Computer Engineering Faculty, Tarbiat Modares University, Tehran, Iran

Abstract: Protein-Protein Interactions (PPIs) play a key role in many biological systems. Thus, identifying PPIs is critical for understanding cellular processes. Many experimental techniques were applied to predict PPIs. The data extracted using these techniques are incomplete and noisy. In this regard, a number of computational methods include machine learning classification techniques have been developed to reduce the noise data and predict new PPIs.

Since, using regression methods to solve classification problems has good results in other applications. Therefore, in this paper, a regression view is applied to the PPI prediction classification problem, so a new approach is proposed using Principal Component Analysis (PCA) and Support Vector Regression (SVR) which has been improved by a new Parallel Hierarchical Cube Search (PHCS) method. Firstly, PCA algorithm is implemented to select an optimal subset of features which leads to reduce processing time and to lessen the effect of noise. Then, the PPIs would be predicted, by using SVR. To get a better performance of SVR, a new PHCS method has been applied to select the appropriate values of SVR parameters. The obtained classification accuracy of the proposed method is 74.505% on KUPS (The University of Kansas Proteomics Service) dataset which outperforms the other methods.

Keywords: Protein-Protein Interaction prediction, Machine Learning approach, Support Vector Regression, Parallel Hierarchical Cube Search.

1. INTRODUCTION

Proteins have the major responsibility in cellular process, such as, signal transduction, gene regulation, cell-cell contact and many additional processes [1]. These responsibilities are performed by the interaction between proteins. Therefore, prediction of PPI improves the knowledge of the cell functionality, protein functions [2-4], gene functions [5], signaling pathway [6] and disease proteins finding [7].

Several high-throughput experimental approaches have been introduced to predicting PPIs, including yeast two-hybrid systems [8], mass Spectrometry [9], protein chip [10] and so on. Unfortunately, the data produced by these methods consist of a large amount of false positive and false negative. Moreover, these methods suffer from high computational time and they only seem able to identify small fraction of all interactions that exist in the cell [11].

In recent years, great efforts have been done to develop some reliable computational methods for predicting PPIs. These methods are mostly classified based on the type of data sources used in the prediction procedure. For instance, some of them use gene data [12], including gene neighborhood [13], gene fusion [1, 14], phylogenetic profile [15] and mirror-tree [16]. Some other methods employ structural information [17, 18], protein sequence [19-23] and domain information [24-26]. Since each of these datasets provides partial information about the interacting pairs, many researchers have attempted to integrate several data source for predicting PPIs with more reliability [27-29]. Results have indicated that the integration of protein pairs information can improve the quality of protein interaction data [30, 31].

Among the proposed machine learning methods for predicting protein-protein interactions, Support Vector Machine (SVM) has shown a better performance than other single classifiers such as: Decision Tree and K-Nearest Neighbor [30]. SVM constructs a hyperplane or set of hyperplanes in feature space which can be used for classification and regression and are capable of dealing with high dimensional input features [32]. A version of SVM for regression is called Support Vector Regression (SVR). SVR outperforms the SVM due to better generalization performance and more robustness against outliers [33]. Like SVM, SVR method requires tuning and setting its parameters properly to achieve better performance and minimize an estimate of the Generalization error [34, 35].

In this paper, a new method, consists of feature extraction and SVR improved by a new Parallel Hierarchical Cube Search (PHCS) method, is presented for solving PPI prediction problem. Since the integration of various data sources produces a high dimensional feature vector, applying feature extraction algorithm would be necessary to reduce processing time and to lessen noise effects. In this paper, at first PCA (Principal component analysis) algorithm is used for feature extraction, then SVR algorithm is carried out for classification. To improve the performance of the model, a

*Address correspondence to this author at the Electrical and Computer Engineering Faculty, Tarbiat Modares University, Tehran, Iran;
E-mail: charkari@modares.ac.ir

1875-0362/15

new Parallel Hierarchical Cube Search (PHCS) method is implemented for tuning SVR kernel parameters optimally. This method improves the performance of prediction system without increasing the overall learning time significantly.

KUPS dataset [28] has been used to evaluate and compare the performance of the proposed method. KUPS is freely available at http://www.ittc.ku.edu/chenlab. The result of the experiments indicates how the classification accuracy has been increased to 74.505% in comparison with other works. This paper is organized as follows: background of protein-protein interactions prediction and support vector regression are discussed in section 2. SVR based on Parallel Hierarchical Cube Search (SVR-PHCS) is presented in section 3. Performance evaluation and experimental results are shown in section 4. Section 5 is conclusion.

2. BACKGROUND

2.1. Protein-Protein Interaction Prediction

Many interesting machine learning methods such as Naïve Bayes [36, 37], Support Vector Machine [20, 38, 39], Decision Tree [40, 41], Random Forest [42] and K-Nearest Neighbor [3,43] have been applied on protein-protein interaction prediction problem.

In 2005, Chen and Liu considered protein domain information to present a domain-based random forest for inferring protein interactions [24].

Ixia et al. in 2010 suggested a Moran autocorrelation descriptor to translate the sequences of protein to numerical feature vector and then to predict PPI's, applying rotation forest method [44]. Xing and Dunson In 2011, proposed a new Bayesian integration method called deemed Nonparametric Bayes Ensemble Learning (NBEL) to predict PPI using the sequence of protein pairs [37].

In 2006 Nanni and Lumini attempted to combine multiple K-local Hyperplane distance Nearest Neighbor (HKNN) classifiers with different physicochemical properties of protein sequence to obtain better classification result [45].

In 2007, Shen *et al.* proposed a new method based on SVM with a kernel function [38]. They applied conjoint triad composition method for constructing feature vectors from sequences of protein pairs. In 2010, Ixia et al. presented a meta approach for PPI prediction which predicts PPIs by combining six independent predictors based on SVM [19].

It is necessary to mention that all the above methods employed one type of data source to predict PPIs.

Qi *et al.* In 2005, used multiple high throughput biological data sources to construct their features vector, including: Y2H, Gene Expression, Protein Expression, Gene Neighborhood, Domain-Domain. Then, they presented a hybrid of random forests and weighted k-nearest neighbour for predicting PPI [29].

They also employed a Mixture-of-Feature-Experts (MFE) method to improve the classification accuracy in this other study in 2007 [31]. The results of these methods show that integration of multiple data sources could improve the prediction of PPI.

Using an appropriate classification technique is crucial in all the mentioned methods for prediction of PPIs. Since, there are some attempts to use regression methods to solve the classification problem in the literature of machine learning [34]. In this work, Support Vector Regression (SVR) is applied as one of the powerful methods in the field of machine intelligence to proper classification of PPIs.

Since the selection of optimal values for the parameters in the SVR model is important to improve the performance of model and minimize an estimate of the Generalization error [46], in this paper a new Parallel Hierarchical Cube Search (PHCS) method is introduced. PHCS selects the optimal value of SVR parameters by searching three dimensional spaces in parallel and hierarchically. To evaluate the efficiency and validity of method, KUPS dataset [28] has been employed, which is the aggregation of different data sources related by PPIs.

2.2. Support Vector Regression (SVR)

The Support Vector Machine (SVM) is known as a popular and useful technique for data classification and regression in machine learning. Let $X = \{(x_i, y_i)\}_{i=1}^{n}$ be a set of *n* training samples, where x_i is input sample and y_i is the corresponding class. Generally, $x_i \in \mathcal{R}^m$ while $y_i \in \{+1, -1\}$ in classification problem and $y_i \in R$ in regression problem. The main idea is to find a linear separating hyperplane $f(x)$ to maximize the distance between two classes;

$$f(x) = w.x + b = \sum_{i=1}^{n} w_i.x_i + b \quad (1)$$

Where, w and b are the weight vector and bias, respectively. In some cases, data in the original input space cannot be linearly separated, and therefore some nonlinear kernel functions should be used. Polynomial, sigmoid and Radial Basis Function (RBF) are the most well-known kernel functions. These kernel functions implicitly map their inputs into high-dimensional feature spaces.

The optimal hyperplane can be determined as follows;

$$Minimize\ \frac{\|\omega\|^2}{2} \text{ With subject to } y_i(\omega x + b) \geq 1\ \forall i \quad (2)$$

Equation (2) is a nonlinear optimization problem with inequality constrains. This problem is solved by using Lagrange multipliers method that represents the following optimization problem (Some kernel tricks are used for nonlinear separating problem):

$$Maximize\ \omega(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{k=1}^{m} \alpha_i y_i\, \alpha_k y_k K(x_i x_k)$$

$$subject\ to\ \sum_{i=1}^{m} \alpha_i y_i = 0\ and\ 0 \leq \alpha_i \leq C\ \forall i\ and\ K(x_i x_j) = \exp\left(\frac{-\|x_i - x_j\|^2}{2\gamma^2}\right) \quad (3)$$

In Equation (3), c and γ are two parameters which are determined experimentally. A linear decision function can be written as $f(x) = sign\ (\sum_{i=1}^{m} \propto_i y_i x x_i + b)$ where b is given by $(y_i - w^T x_i)$. In cases where the decision function is non-linear, the input space is mapped to another Euclidean

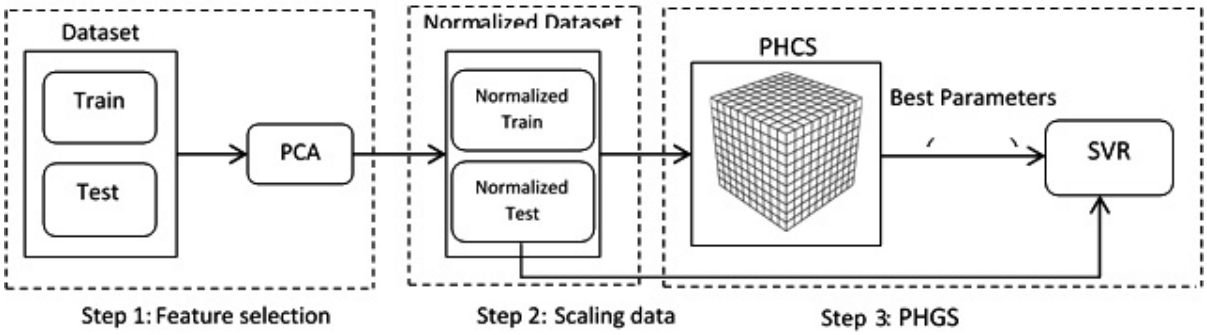

Fig. (1). Steps of SVR-PHCS method.

space by the kernel function in advance. This decision function is formulated as:

$$f(x) = sign\ (\sum_{i=1}^{m} \propto_i y_i K(x, x_i) + b) \quad (4)$$

In SVR the mathematical formulation has to consider the approximation errors. SVM solve the regression problem by introducing a ε -insensitive loss function $L_\varepsilon(y)$.

$$L_\varepsilon(y) = |y - f(x)|_\varepsilon = \max(0, |y - f(x)| - \varepsilon) \quad (5)$$

Considering the above function, SVR is performed regression by minimizing the following function:

$$Min\ \omega, b, \xi_i \xi_i^* \frac{1}{2} \|\omega^2\| + C\ \sum_{i=1}^{m} (\xi_i + \xi_i^*),$$

$$subject\ to\ \forall i \in \{1, \ldots, m\}, \xi_i, \xi_i^* \geq 0,$$

$$(\omega . x_i + b) - y_i \leq \varepsilon + \xi_i\ , y_i(\omega . x_i + b) \leq \varepsilon + \xi_i^* \quad (6)$$

Where slack variable ξ_i, represents the upper training error and ξ_i^* is the lower training error. In non-linear SVR, the following equation indicates kernel expansion of the decision function f which is defined as follows;

$$f(x, \alpha_i^*, \alpha_i) = \sum_{i=1}^{m} (\alpha_i^* - \alpha_i) K(x_i, x) + b \quad (7)$$

The SVR parameters (c, γ, ε) directly effect on the classification performance and the complexity of regression. Tuning and setting of these parameters to get a better decision function, is an open research problem. The main contribution of this study is on this problem. Therefore, a new PHCS method to optimally select these parameters is proposed.

3. SVR BASED ON PARALLEL HIERARCHICAL CUBE SEARCH (SVR-PHCS)

In this section, details of the proposed PCA and SVR-PHCS method are introduced. Accordingly, the progress has been started with proper number of features which is extracted by using PCA, and then attempt to obtain optimum parameters value of SVR which uses the RBF kernel function. The method consists of the following steps (Fig. **1**):

1. Feature selection
2. Scaling data
3. Parallel Hierarchical Cube Search (obtain the best (c, γ, ε) by cross validation scoring)

In this section each of these steps is explained in details.

3.1. Feature Selection

Biological datasets are generally very large, dimensional and noisy. One of these datasets is KUPS. KUPS is a highly dimensional dataset created by aggregating of multiple data sources, but not all features are effective in the prediction. So, a set of feature extraction methods are usually employed for dimensionality reduction [47, 48]. In this way, the irrelevant and redundant features are put away from a dataset to reduce the data dimensionality. It cause low complexity of data, increases the search speed and consequently increases the performance of the classification.

Among these, PCA is one of the most widely used algorithms for dealing with this problem. PCA is a linear combination that changes the coordinate system of data (feature vector) to a new one, such that the new set of features are linear functions of the original features and uncorrelated. Here, the greatest variance by any projection of the data lies on the first coordinate, and the second greatest variance on the second coordinate, and so on. After applying PCA, the features which lead to a better accuracy were selected.

Table **1** show the average and variance of accuracy when the numbers of feature change from 50 to 400 on KUPS dataset after ten runs. As it is found, when the numbers of features are 250, the better accuracy would be obtained.

3.2. Scaling Data

Variables often have considerably different numerical ranges. When a variable be in a large range its variance become large, and vice versa. Since PCA is a maximum variance method, it leads that a variable with a large variance is more likely to be expressed in a modeling. In this regard, all the data would be scaled in advance in order to provide the same contribution for them to the model. Another advantage is to avoid numerical difficulties during the calculation. Since kernel values usually depend on the inner products of feature vectors, e.g. the linear kernel and the polynomial kernel, large attribute values might cause numerical problems [45]. For this purpose, Eq. (8) is used for linearly scaling, where X indicates the original data, $X_{Normalized}$ is the normalized data, X_{max} and X_{min} are the maximum and minimum values of X, respectively.

$$X_{Normalized} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (8)$$

Table 1. Average and variance of accuracy for different number of features extracted by PCA on KUPS.

Principal Component Analysis (PCA)		Accuracy
Number of selected Features		
50	Average	70.725
	Variance	±0.475
100	Average	72.537
	Variance	±0.12
150	Average	73.708
	Variance	±0.136
200	Average	73.152
	Variance	±0.21
250	Average	74.348
	Variance	±0.157
300	Average	73.546
	Variance	±0.142
350	Average	73.498
	Variance	±0.014
400	Average	72.125
	Variance	±0.11

3.3. From Regression to Classification

While in the literature of machine learning, classification and regression problems are addressed as two different problems differentiated by categorical or continuous dependent variable, there have been some attempts to use regression methods to solve the classification problems and vice versa [34].

In this paper, the support vector machine regression method is used to solve the classification problem (PPI prediction). Since in the regression problem, the class labels are real-valued rather than binary-valued, a solution is needed to map the real-valued class label to binary-valued for classifying. Therefore, if a perfect mapping method is applied, the classification problem can be solved by regression methods. The most important aspect of rounding values is the selection of mapping point (MP). The following pseudo code presents the overall procedure for choosing MP.

Algorithm 1 – steps of MP algorithm

```
Train () {
Best MP=0;
Best accuracy=0;
For each combination of c , γ and ε
  Execute SVR algorithm (generate Output)
  For step=0.1:0.1:1
    MP=step;
    For all output
      If (Output < MP)
        Output = 0; // non-interaction
      Else
        Output = 1; // interaction
      End if
    End for
  Calculate accuracy
  If (accuracy > best accuracy)
    Best accuracy = accuracy;
    Best MP = MP;
    Best c = c;
    Best γ = γ;
    Best ε = ε;
  End if
End for
Test ( Best c , Best γ , Best ε , Best MP)
}

Test ( c , γ , ε , MP) {
Learn SVR with c , γ and ε parameters
```

```
For all output line
      If (Output line < MP)
        Output line=0;
      Else
        Output line=1;
      End if
End for
}
```

3.3.1. Parallel Hierarchical Cube Search (PHCS)

PHCS method is employed for tuning SVR kernel function parameters. It should be noted that the selection of the best values for kernel function parameters is an NP complete problem, so the selected parameter values are not necessarily the best overall calculation.

There are some methods to find these parameters properly. They mostly differ in the way the search the parameter space. Among them, greedy search, pattern search and GA are mostly used in different applications. PHCS is the extended version of PHGS introduced in [50]. PHGS method is used for tuning SVM kernel function parameters (c, γ). Therefore, it has a grid search space of two dimensions. While the PHCS is applied for tuning SVR kernel function parameters (c, γ, ε), with the three-dimensional search space. Moreover, it is able to find Mapping Point (MP). So that for mapping the real-valued class label would be mapped in to binary-valued one. In this work, Cross Validation Score (CVS) has been used to validate the hierarchical cube search effectively. A_θ is considered support vector machine regression learning algorithm where θ is a vector of SVR parameters with RBF kernel function. A_θ is employed on dataset D, A_θ (D); the result will be a classifier. Given a set Θ, assessment of CVS of the best accessible classifier A_θ^*(D) is desired, where $\theta^* \in \Theta$ is the best assignment for D.

In order to calculate the CVS, the following k-fold cross validation procedure is applied, which returns the cross validation score of k different classifier that are learned by the algorithm on different folds of dataset. The cross validation procedure consists of the following steps:

1. Data permutation and split. Randomly permute the whole data and then split it into k non-overlapping equally sized subsets D_i which is called folds. Each times k-1 folds are assumed as train and one fold for validation.
2. Train classifiers over folds. Algorithm repeats k times while in each iteration; one subset is tested using the classifier trained on the remaining k-1 subsets. Finally, each instance of the whole training set is predicted.
3. Calculate cross validation score. CVS is obtained by Equation. (9).

$$Cross\ Validation\ Score\ (CVS) = \frac{\#\ Recordes\ true\ predicted}{\#\ Total\ Recordes} \qquad (9)$$

K-fold cross validation minimizes the bias associated with the random sampling of the training. Because of this property, it is widely used among researchers. Now, the proposed PHCS method would be described in details. There are three main parameters for SVR kernel function: c, γ and ε. The c parameter trades off misclassification of training examples against simplicity of the decision surface. A low value of c makes the decision surface smooth, while a high value of c aims at classifying all training examples correctly. As a result, Parameter c controls the balance between the complexity of the machine and the number of separable points. The γ parameter defines how far the influence of a single training example reaches. Low values are meant as 'far' and high values as 'close'. On the other hand, the value of parameter ε is very crucial in support vector condition and hence in the model performance. Choosing some large values for ε, the number of support vectors is decreased, in this way ε bond becomes wider and the range of accepa error increases. In addition, very small values of ε makes more support vectors and increases the risk of over-training.

The best values of these parameters depend on the nature of the problem. Selecting the best values for parameters is a vital step that has a direct effect on the performance and overall capability of SVR learning algorithm. Grid search is one of the popular techniques for finding the optimal values for SVM kernel function parameters. This method is very popular and reliable for selecting the best value on parameter ranges. However, this approach suffers from dimensionality, grid granularity and high computational time [49, 51].

In SVR with RBF kernel function, there are three parameters that should be found out in 3D search space. In order to find the best parameter values, a hierarchical cube search method is used. Although this method saves time, but it is still time consuming. Since all points on the cube are independent from each other, hierarchical cube search can be implemented in parallel. With parallel implementation of a hierarchical cube search, the required time to find the best parameters will be significantly reduced.

In this paper, exponentially growing sequences of $c, \gamma\ and\ \varepsilon$ and search the optimal values of these parameters are considered in the space of $(c = 2^{-5} \dots 2^{15}, \gamma = 2^{-15} \dots 2^{3}, \varepsilon = 2^{-15} \dots 2^{0})$.

In order to find the best values for (c, γ, ε) in user defined boundaries, the whole search space must be searched. N is assumed the number of available CPUs, consequently c, γ and ε values divided into N interval. Then, each interval is assigned to one CPU. Interval division task and assigning each interval to one CPU are managed by one CPU as master. Each CPU performs the cube search on the total space that belongs to it. For each triple (c, γ, ε) in an interval, the CPU calculates the CVS for all them. Then, based on the maximum value of CVS, the best (c, γ, ε) is selected as the best local optimum for each CPU. Fig. (**2**) presents all i CPUs that find the best local (c, γ, ε) values in parallel and in independent manner. As it is shown, all triple of (c, γ, ε)

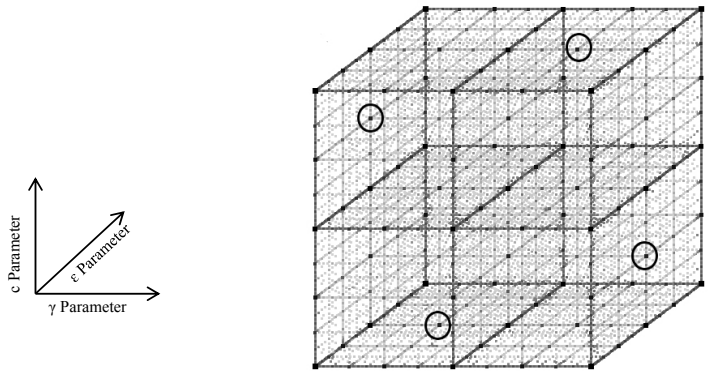

Fig. (2). Parallel selection of the best triple of .

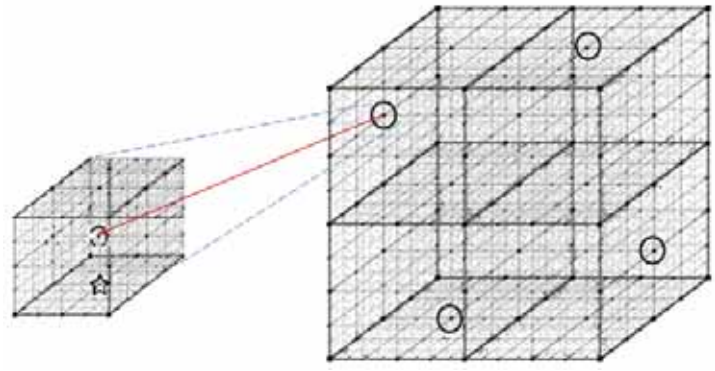

Fig. (3). Construct hierarchical virtual window at parallel.

that have the maximum CVS in each CPU, have been marked.

When the candidate values of (c, γ, ε) are found for each CPU, all the N candidate values are compared and the best one is chosen. So the local search procedure will be continued for the chosen candidate point. In this regard, the local search is done in the neighborhood of the selected candidates with smaller steps to find the best possible result.

In the next iteration, in order to find the best values of (c, γ, ε), a virtual cube around the best local optimum point of the last iteration is defined. This virtual cube denotes new search space which is divided in to N new intervals. Each CPU begins to search the new determined search space to find a better triple of (c, γ, ε). Then, the best CVS in the new space will be searched to find the optimum values again. Fig. (**3**) represents the hierarchical constructing virtual cube and finding the best new local (c, γ, ε). As it is shown, the new best local (c, γ, ε) is marked as star.

By increasing the iterations of parallel hierarchical cube search, the accuracy will be increased. However, it leads to more processing time. Therefore, a trade-off between accuracy and processing time should be considered to solve the problem.

Finally, all the best i local values of CVS will be compared to select the best (c, γ, ε) as global optimum. Then, SVR algorithm is performed on train and test dataset using the best global (c, γ, ε)values.

The overall process of the proposed SVR-PHCS method is illustrated in Fig. (**4**).

4. PERFORMANCE EVALUATION

4.1. Metrics

Confusion matrix contains information about the actual and predicted class of samples that are classified by the classification method.

The performance of supervised machine learning techniques can be evaluated by confusion matrix. Parameters used in the confusion matrix are:

TP: The number of interacting proteins that are correctly classified.

FN: The number of interacting proteins that are wrongly classified non-interactive.

TN: The number of non-interacting protein pairs that are correctly classified.

FP: The number of non-interacting protein pairs that are incorrectly classified as interactive.

In the following, a series of evaluation metrics that have been used in this work is presented:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (10)$$

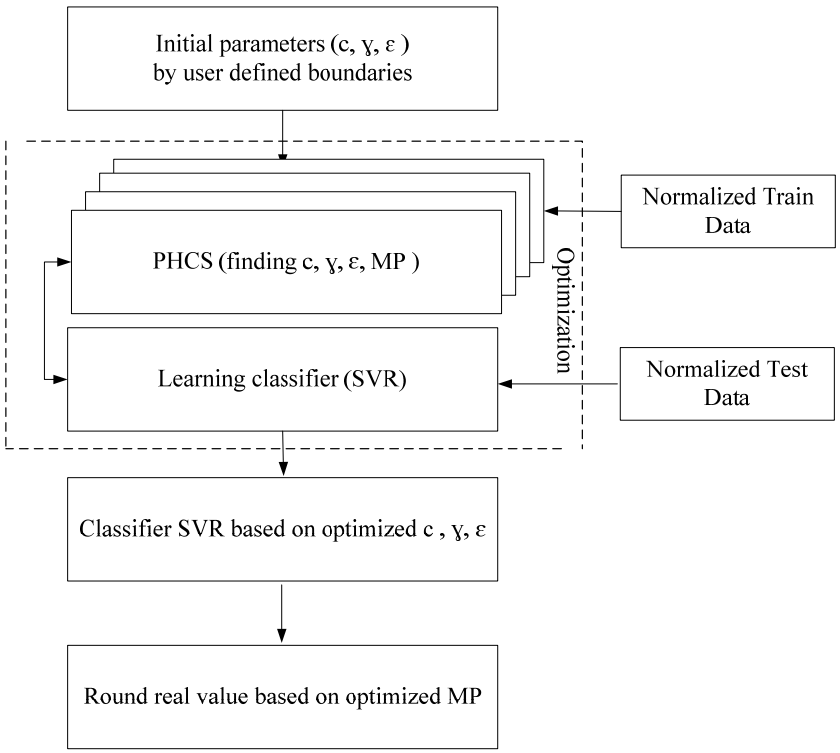

Fig. (4). SVR-PHCS integration.

Table 2. The ten highest combination of c , γ and ε values from SVR-PHCS.

c Value	γ Value	ε Value	The Accuracy Rate of Classification (%)
2	0.125	0.0002441406	74.7
1	0.125	0.25	74.472
1	0.25	0.0002441406	74.434
2	0.125	0.03125	74.377
2	0.125	0.0004882812	74.348
2	0.125	0.0000610352	74.32
2	0.125	0.0625	74.282
4	0.25	0.0000305176	74.263
4	0.25	0.0004882812	74.225
4	0.25	0.00390625	74.196

$$precision = \frac{TP}{TP + FP} \quad (11)$$

$$\mathrm{Re}call = \frac{TP}{TP + FN} \quad (12)$$

$$F - Measure = \frac{2.\mathrm{Pr}ecision.\mathrm{Re}call}{(\mathrm{Pr}ecision + \mathrm{Re}call)} \quad (13)$$

4.2. Experimental Result

Table **2** shows the 10 best combinations of c, γ and ε values that have been obtained from SVR-PHCS.

The results of performance evaluation metrics by using SVR and the extension of SVR (SVR-PHCS) are presented in Tables **3** and **4**, respectively. As it is indicated, the performance of SVR-PHCS is better than the classical SVR. Various performance evaluation metrics can be considered as

Table 3. Performance metrics for SVR on KUPS dataset.

Predicted	Actual	
	Negative	**Positive**
Positive	1651	3297
Negative	3607	1961
Precision	66.632	
Recall	62.704	
F-Measure	64.608	
Classification Accuracy	65.652	
Selected C	1	
Selected γ	0.004	
Selected ε	0.1	

Table 4. Performance metrics for SVR-PHCS on KUPS dataset.

Predicted	Actual	
	Negative	**Positive**
Positive	790	3367
Negative	4468	1891
Precision	80.995	
Recall	64.035	
F-Measure	71.523	
Classification Accuracy	74.505	
Selected C	2	
Selected γ	0.125	
Selected ε	0.0002441406	

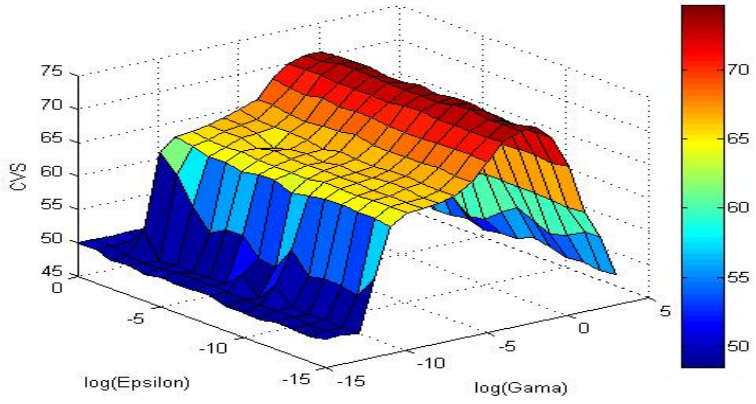

Fig. (5). Cross Validation Score (CVS) changes for all combination of γ and ε on the best c value (c=2).

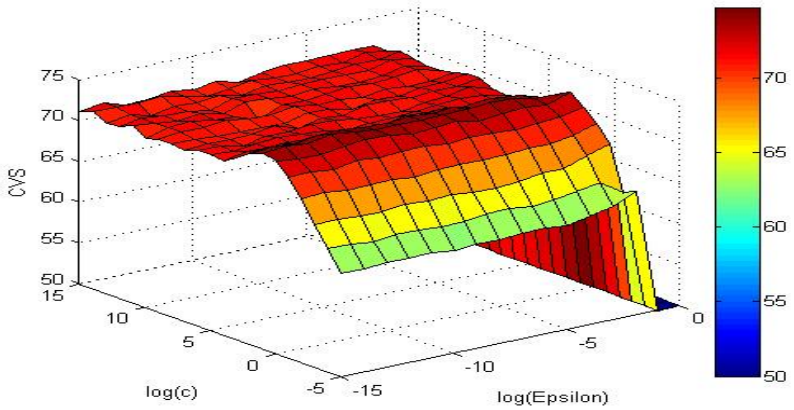

Fig. (6). Cross Validation Score (CVS) changes for all combination of c and ε on the best γ value (γ =0.125).

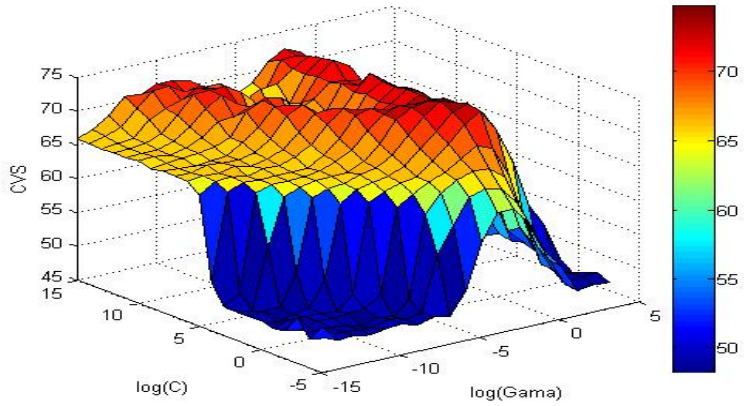

Fig. (7). Cross Validation Score (CVS) changes for all combination of c and γ on the best ε value (ε=0.0002441406).

CVS. In this paper, the CVS is used as accuracy. The results of the search space for the best c, γ and ε are presented in Fig. (**5**-**7**), respectively. In Fig. (**5**), the combination of γ and ε for the best c value have been shown; all the combination of c and ε for the best γ value have been indicated in Fig. (**6**). Moreover, in Fig. (**7**), all the combination of c and γ for the best ε value has been plotted.

In order to find out the effect of the MP value on performance evaluation metrics, the predicted outputs of test samples are mapped into two binary classes using various MP. Fig. (**8**) shows Accuracy, Precision, Recall and F-Measure value while MP changes from 0.1 to 1 with interval 0.1.

4.3. Comparison with other Works

The proposed method is compared with other well-known prediction methods based on KUPS (The University of Kansas Proteomics Service) dataset. This dataset contains PPI of various organisms which is aggregated from seven data sets including, MINT, IntAct, HPRD, Gene Ontology, Uniprot, AAindex and PSSM [28]. The dataset is composed of training and testing sets, where training set has 10518 protein pairs and testing set has 10516 protein pairs. Each protein pair in KUPS is composed of 400 features. To compare the results, accuracy and F-Measure have been used as a proper metric. The results of the proposed method and other classification methods on KUPS dataset are showed in Table **5**.

Precision measures the exactness of a classifier, Whereas, *Recall* measures the completeness, or sensitivity of a classifier. Improving Recall often decreases precision and vice versa. Precision and Recall are combined to produce a single metric known as *F-measure*, which is the weighted harmonic average of Precision and Recall. In this paper, the results are compared with other results by accuracy and f-measure metrics.

CONCLUSION

There are many classification techniques to predict Protein-Protein Interactions in literature. Using regression methods is a new approach to solve classification problems. In this paper, a new approach is proposed using PCA and Support Vector Regression (SVR) which has been improved by a new Parallel Hierarchical Cube Search (PHCS) method. The major challenge of applying SVR is how to

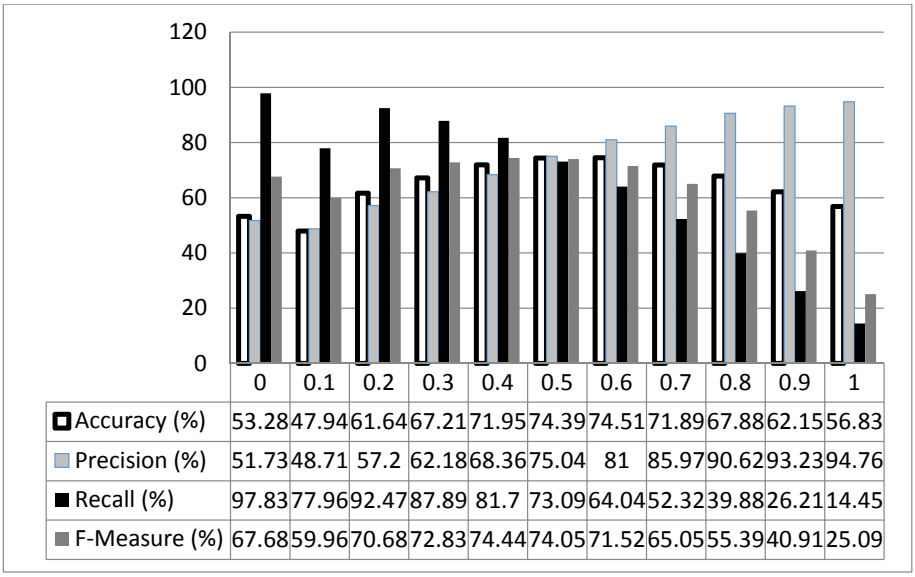

Fig. (8). Various values of Accuracy, Precision, Recall and F-Measure for different MP, X axis indicates as MP.

Table 5. Classification accuracy of SVR-PHCS and other methods on KUPS dataset.

	Accuracy	Precision	Recall	F-Measure
Naive Bayes [28]	57.6%	55.7%	73.7%	63.45%
Decision tree (c4.5) [28]	58.9%	58.8%	59.4%	59.1%
SVM [28]	70.8%	73.1%	65.8%	69.26%
Random Forest [28]	71.5%	72.7%	69.0%	70.8%
STRING [52]	NA	59%	59%	59%
PPI Finder [52]	NA	65%	47%	55%
$Domain_{m1}$ [52]	NA	88%	29%	43%
$Domain_{m2}$ [52]	NA	81%	43%	57%
ATRP [52]	NA	93%	49%	64%
SVR-PHCS	**74.505%**	77.062%	70.349%	**73.552%**

tune and set the parameters (to achieve the best performance) for a given dataset and how to map the regression output to classification label. In this regard, the PHCS is applied to tune SVR parameters (c, γ and ε) and select the mapping point. The proposed method has been employed on KUPS dataset that is an aggregating of multiple data source and highly dimensional. Some features of the dataset may have no effect at all, or contain a high level of noise. Deletion of such features increases the search speed and the accuracy rate, therefore PCA has been used to select the appropriate features.

According to the experimental results, SVR-PHCS prediction system obtains very promising results in classifying the protein pairs. The results indicate 74.705% accuracy, which is one of the best results reported for this dataset.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflicts of interest.

ACKNOWLEDGEMENTS

Declared none.

REFERENCE

[1] R. Roslan, R.M. Othman, Z.A. Shah, S. Kasim, H. Asmuni, J. Taliba, R. Hassan, and Z. Zakaria, "Utilizing shared interacting domain patterns and Gene Ontology information to improve protein–protein interaction prediction," *Comput. Biol. Med.*, vol. 40, pp. 555-564, 2010.

[2] 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, pp. 86-90, 1999.

[3] 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. Biophys. Res. Commun.*, vol. 380, pp. 318-322, 2009.

[4] L. Hu, T. Huang, X. Shi, and W.C. Lu, "Predicting functions of proteins in mouse based on weighted protein-protein interaction network and protein hybrid properties," *PLoS ONE*, vol. 6, p. e14556, 2011.

[5] B.Q. Li, T. Huang, and L. Liu, "Identification of colorectal cancer related genes with mRMR and shortest path in protein-protein interaction network," *PLoS ONE*, vol. 7, p. e33393, 2012.

[6] A. Gitter, J. Klein-Seetharaman, A. Gupta, and Z. Bar-Joseph, "Discovering pathways by orienting edges in protein interaction networks," *Nucleic Acids Res.*, vol. 39, pp. e22-e22, 2011.

[7] T.-P. Nguyen, W.-c. Liu, and F. Jordán, "Inferring pleiotropy by network analysis: linked diseases in the human PPI network," *BMC Syst. Biol.*, vol. 5, p. 179, 2011.

[8] L. Giot, J.S. Bader, C. Brouwer, A. Chaudhuri, B. Kuang, Y. Li, Y.L. Hao, C.E. Ooi, B. Godwin, E. Vitols, G. Vijayadamodar, P. Pochart, H. Machineni, M. Welsh, Y. Kong, B. Zerhusen, R. Malcolm, Z.Varrone, A. Collis, M. minto, S. Burgess, L. McDaniel, E. Stimpson, F. Spriggs, J. Williams, K. Neurath, N. Liome, M. Agee, E. Voss, K. Furtak, R. Renzulli, N. Aanensen, S. Carrolla, E. Bickelhaupt, Y. Lanzovatsky, A. DaSilva, J. Zhong, C.A. Stanyon, R.L. Finley Jr, K.P. White, M. Braverman, T. Jarvie, S. Gold, M. Leach, J. Knight, R.A. Shimkets, M.P. McKenna, J. Chany, and J.M. Rothberg, "A protein interaction map of Drosophila melanogaster," *Science*, vol. 302, no. 5651, pp. 1727-1736, 2003.

[9] A.C. Gingras, R. Aebersold, and B. Raught, "Advances in protein complex analysis using mass spectrometry," *J. Physiol.*, vol. 563, pp. 11-21, 2005.

[10] H. Zhu, M. Bilgin, R. Bangham, D. Hall, A. Casamayor, P. Bertone, N. lan, R. Jansen, S. Bidlingmaier, T. Houfek, T. Mitchell, P. Miller, R.A. Dean, M. Gerstein, and M. Snyder, "Global analysis of protein activities using proteome chips," *Science*, vol. 293, pp. 2101-2105, 2001.

[11] T. Mohamed, J. Carbonell, and M. Ganapathiraju, "Active learning for human protein-protein interaction prediction," *BMC Bioinformatics*, vol. 11, p. S57, 2010.

[12] J. Yu, and F. Fotouhi, "Computational approaches for predicting protein–protein interactions: a survey," *J. Med. Syst.*, vol. 30, pp. 39-44, 2006.

[13] T. Dandekar, B. Snel, M. Huynen, and P. Bork, "Conservation of gene order: a fingerprint of proteins that physically interact," *Trends Pharmacol. Sci.*, vol. 23, pp. 324-328, 1998.

[14] E.M. Marcotte, M. Pellegrini, H.-L. Ng, D.W. Rice, T.O. Yeates, and D. Eisenberg, "Detecting protein function and protein-protein interactions from genome sequences," *Science*, vol. 285, pp. 751-753, 1999.

[15] 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. Natl. Acad. Sci. USA*, vol. 96, pp. 4285-4288, 1999.

[16] F. Pazos, and A. Valencia, "Similarity of phylogenetic trees as indicator of protein–protein interaction," *Protein Eng.*, vol. 14, pp. 609-614, 2001.

[17] L. Licamele, and L. Getoor, "*Predicting protein-protein interactions using relational features*," University of Maryland Institute for Advanced Computer Studies, 2007.

[18] U. Ogmen, O. Keskin, A.S. Aytuna, R. Nussinov, and A. Gursoy, "PRISM: protein interactions by structural matching," *Nucleic Acids Res.*, vol. 33, pp. W331-W336, 2005.

[19] J.-F. Xia, X.-M. Zhao, and D.-S. Huang, "Predicting protein–protein interactions from protein sequences using meta predictor," *Amino Acids*, vol. 39, pp. 1595-1599, 2010.

[20] M.-G. Shi, J.-F. Xia, X.-L. Li, and D.-S. Huang, "Predicting protein–protein interactions from sequence using correlation coefficient and high-quality interaction dataset," *Amino Acids*, vol. 38, pp. 891-899, 2010.

[21] C.-Y. Yu, L.-C. Chou, and D.T. Chang, "Predicting protein-protein interactions in unbalanced data using the primary structure of proteins," *BMC Bioinformatics*, vol. 11, p. 167, 2010.

[22] D.T. Chang, Y.-T. Syu, and P.-C. Lin, "Predicting the protein-protein interactions using primary structures with predicted protein surface," *BMC Bioinformatics*, vol. 11, p. S3, 2010.

[23] K.C. Chou, and Y.D. Cai, "Predicting protein-protein interactions from sequences in a hybridization space," *J. Proteome Res.*, vol. 5, pp. 316-322, 2006.

[24] X.-W. Chen, and M. Liu, "Prediction of protein–protein interactions using random decision forest framework," *Bioinformatics*, vol. 21, pp. 4394-4400, 2005.

[25] J.L. Morrison, R. Breitling, D.J. Higham, and D.R. Gilbert, "A lock-and-key model for protein–protein interactions," *Bioinformatics*, vol. 22, pp. 2012-2019, 2006.

[26] M. Singhal, and H. Resat, "A domain-based approach to predict protein-protein interactions," *BMC Bioinformatics*, vol. 8, p. 199, 2007.

[27] M.S. Scott, and G.J. Barton, "Probabilistic prediction and ranking of human protein-protein interactions," *BMC Bioinformatics*, vol. 8, p. 239, 2007.

[28] X.-w. Chen, J.C. Jeong, and P. Dermyer, "KUPS: constructing datasets of interacting and non-interacting protein pairs with associated attributions," *Nucleic Acids Res.*, vol. 39, pp. D750-D754, 2010.

[29] Y. Qi, J. Klein-Seetharaman, Z. Bar-Joseph, Y. Qi, and Z. Bar-joseph, "Random Forest Similarity for Protein-Protein Interaction Prediction," *Pac. Symp. Biocomput.*, pp. 531-42, 2005.

[30] Y. Qi, Z. Bar-Joseph, and J. Klein-Seetharaman, "Evaluation of different biological data and computational classification methods for use in protein interaction prediction," *Proteins*, vol. 63, pp. 490-500, 2006.

[31] Y. Qi, J. Klein-Seetharaman, and Z. Bar-Joseph, "A mixture of feature experts approach for protein-protein interaction prediction," *BMC Bioinformatics*, vol. 8, p. S6, 2007.

[32] Y. Zhan, and H. Cheng, "A robust support vector algorithm for harmonic and interharmonic analysis of electric power system," *Electr. Power Syst. Res.*, vol. 73, pp. 393-400, 2005.

[33] G. Nalbantov, P.J. Groenen, and J.C. Bioch, "Support Vector Regression Basics," *Medium Econometrische Toepassingen*, vol. 13, pp. 16-19, 2005.

[34] M.H. Zangooei, and S. Jalili, "Protein secondary structure prediction using DWKF based on SVR-NSGAII," *Neurocomputing*, vol. 94, pp. 87-101, 2012.

[35] A.L. Oliveira, P.L. Braga, R.M. Lima, and M.L. Cornélio, "GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation," *Inform. Softw Tech.*, vol. 52, pp. 1155-1166, 2010.

[36] R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N.J. Krogan, S. Chung, A. Emili, M. Snyder, J.F. Greenblatt, and M. Gerstein, "A Bayesian networks approach for predicting protein-protein interactions from genomic data," *Science*, vol. 302, no. 5644, pp. 449-453, 2003.

[37] C. Xing, and D.B. Dunson, "Bayesian inference for genomic data integration reduces misclassification rate in predicting protein-protein interactions," *PLoS Comput. Biol.*, vol. 7, p. e1002110, 2011.

[38] J. Shen, J. Zhang, X. Luo, W. Zhu, K. Yu, K. Chen, Y. Li, and H. Jiang, "Predicting protein–protein interactions based only on sequences information," *Proc. Natl. Acad. Sci. USA*, vol. 104, no. 11, pp. 4337-4341, 2007.

[39] 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, pp. 3025-3030, 2008.

[40] L.V. Zhang, S.L. Wong, O.D. King, and F.P. Roth, "Predicting co-complexed protein pairs using genomic and proteomic data integration," *BMC Bioinformatics*, vol. 5, p. 38, 2004.

[41] J. Wang, C. Li, E. Wang, and X. Wang, "Uncovering the rules for protein–protein interactions from yeast genomic data," *Proc. Natl. Acad. Sci. USA*, vol. 106, no. 10, pp. 3752-3757, 2009.

[42] N. Lin, B. Wu, R. Jansen, M. Gerstein, and H. Zhao, "Information assessment on predicting protein-protein interactions," *BMC Bioinformatics*, vol. 5, p. 154, 2004.

[43] T. Huang, L. Chen, and Y.D. Cai, "Classification and analysis of regulatory pathways using graph property, biochemical and physicochemical property, and functional property," *PLoS ONE*, vol. 6, p. e25297, 2011.

[44] J.-F. Xia, K. Han, and D.-S. Huang, "Sequence-based prediction of protein-protein interactions by means of rotation forest and autocorrelation descriptor," *Protein Pept. Lett.*, vol. 17, pp. 137-145, 2010.

[45] L. Nanni, and A. Lumini, "An ensemble of K-local hyperplanes for predicting protein–protein interactions," *Bioinformatics*, vol. 22, pp. 1207-1210, 2006.

[46] K. Smets, B. Verdonk, and E.M. Jordaan, "Evaluation of performance measures for SVR hyperparameter selection," in *Neural Networks, 2007. IJCNN 2007. International Joint Conference* , 2007, pp. 637-642.

[47] M.-G. Shi, D.-S. Huang, and X.-L. Li, "A protein interaction network analysis for yeast integral membrane protein," *Protein Pept. Lett.*, vol. 15, pp. 692-699, 2008.

[48] S. Asur, D. Ucar, and S. Parthasarathy, "An ensemble framework for clustering protein–protein interaction networks," *Bioinformatics*, vol. 23, pp. i29-i40, 2007.

[49] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "*A practical guide to support vector classification*," 2003.

[50] M.H. Zangooei, and S. Jalili, "PSSP with dynamic weighted kernel fusion based on SVM-PHGS," *Knowl. Based Syst.*, vol. 27, pp. 424-442, 2012.

[51] J. Wang, X. Wu, and C. Zhang, "Support vector machines based on K-means clustering for real-time business intelligence systems," *Int. J. Business Intell. Data Mining*, vol. 1, pp. 54-64, 2005.

[52] Y.-T. Tang and H.-Y. Kao, "Augmented transitive relationships with high impact protein distillation in protein interaction prediction," *Biochim Biophys. Acta (BBA)-Proteins and Proteomics*, vol. 1824, pp. 1468-1475, 2012.

Received: February 13, 2014 Revised: February 28, 2014 Accepted: October 28, 2014