Evaluation of Normalization and PCA on the Performance of Classifiers for Protein Crystallization Images

İmren Dinç[1], Madhav Sigdel[1], Semih Dinç[1], Madhu S. Sigdel[1], Marc L. Pusey[2], Ramazan S. Aygün[1]
[1]Computer Science Department
[1]University of Alabama in Huntsville
[1]Huntsville Alabama 35899
[2]iXpressGenes Inc., 601 Genome Way, Huntsville, Alabama 35806
Email: [1]{id0002, ms0023, sd0016, mss0025, aygunr} @uah.edu, [2]marc.pusey@ixpressgenes.com

***Abstract*—In this paper, we investigate the performance of classification of protein crystallization images captured during protein crystal growth process. We group protein crystallization images into 3 categories: noncrystals, likely leads (conditions that may yield formation of crystals) and crystals. In this research, we only consider the subcategories of noncrystal and likely lead protein crystallization images separately. We use 5 different classifiers to solve this problem and we applied some data preprocessing methods such as principal component analysis, min-max normalization and z-score (ZS) normalization methods to our datasets in order to evaluate their effects on classifiers for the noncrystals and likely-leads datasets. We performed our experiments on 1606 noncrystal and 245 likely lead images independently. We had satisfactory results for both datasets. We reached 96.8% accuracy for noncrystal dataset and 94.8% accuracy for likely leads dataset. Our target is to investigate the best classifiers with optimal preprocessing techniques on both noncrystal and likely leads datasets.**

Keywords—protein crystallization; classification; normalization; principal component analysis

I. INTRODUCTION

Each protein has its own particular 3D structure. The structure of a protein is significantly important, since it provides information about the functionality of proteins. Protein crystallization is the trial process of growing proteins deployed into numerous solutions with varying conditions [1]. Some of these trials may lead to a successful crystallization and this may help crystallographers identify the structure of a protein crystal using X-ray diffraction [1]. The protein crystallization is a challenging activity because these experiments may take so long time or so many attempts to grow a protein crystal due to their sensitivity to thermodynamic (pH, temperature, etc.) and hard to control kinetic (equilibration rates, molecular association, etc.) factors [2].

Today's robotic systems have started to perform the protein crystallization experiments in an automated way. Those robotic systems may execute thousands of experiments per day and the systems can record the images of each plate well periodically. This enables experts to track the growth of the protein structure over a period of time. The drawback of these robotic systems is that each image needs to be reviewed by a human expert to label it [3]. Evaluation of these results automatically is as important as performing these experiments since there are thousands of images to examine.

There has been some research on the classification of protein crystallization images in the literature, and there are many attempts to split these images into different number of categories. Sigdel et al. [4] have classified protein crystallization images into 3 main categories (crystals, likely leads, and noncrystals). Bern et al. [5] used 5 categories (empty, clear, precipitate, microcrystal hit, and crystal) in their study. Spraggon et al. [6] have divided the images into 6 categories (experimental mistake, clear drop, homogenous precipitant, inhomogeneous precipitant, microcrystals, and crystals). Cumbaa et al. [3] have offered to classify the proteins images into 3 or 10 categories.

Our dataset comprises of protein crystallization images that were collected at iXpressGenes, Inc. We have categorized images into three groups at UAH with the help of Dr. Pusey at iXpressGenes, Inc. In this paper, we only focus on classification of noncrystals and likely leads subcategories separately. To solve these classification problems, we compare the results of Random Forest (RF), Bayesian (BYS), Support Vector Machines (SVM), Neural Networks (NNW), Decision Tree (ID3), and Linear Discriminant Analysis (LDA) classifiers to determine how they perform for our dataset. Before the classification process, we also apply some data preprocessing methods such as normalization and data reduction since these kinds of preprocessing techniques may improve the accuracy and the effectiveness of the classifiers [7]. In this study, we apply min-max (MM), z-score (ZS) normalization and principal component analysis (PCA) to our datasets to evaluate the performance of classifiers.

This paper is organized as follows. The categories and features of the protein crystallization images are described in Section 2 and Section 3, respectively. Brief explanations of the classification and data preprocessing techniques are given in Section 4. The experiments results are presented in Section 5. Finally, the last section concludes our paper.

II. BACKGROUND

We classify protein crystallization images into three main categories: noncrystals, likely leads, and crystals. In this paper, only non-crystals and likely leads data are analyzed. We focus on these two categories since a) these may give information about early stages of crystal formation and b) crystal sub-category classification may require a different feature set that may indicate the shapes of crystals. The subcategories of these classes are described below.

A. Noncrystals

This category consists of images under the following protein crystallization phases: clear drop, phase separation, and regular granular precipitates. The conditions corresponding to these images do not have crystals and are observed in the early stages of protein crystal growth whether successful or not. Images under phase separation occur very rarely. The phase separation category is not considered in our experiments because of limited data. Therefore, we classify only into the two sub-categories of the non-crystals: clear drop and regular granular precipitates in this paper.

1) Clear Drop: Clear drop category corresponds to the initial state of the crystallization process. Fig. 1 shows a few sample images under this category.

2) Regular Granular Precipitate: This category consists of images where the precipitation has just begun. Precipitates are seen in the form of clouds. Fig. 2 shows some sample images under this category.

B. Likely Leads

This category contains images of protein crystallization that are in intermediate phase between noncrystals and crystals. This means that those images are good candidates to complete their structural growth successfully. This category consists of two subsets: granular precipitate (microcrystals) and unclear bright regions. These subcategories are briefly described below.

1) Granular Precipitate or Microcrystals: This category contains the images of numerous microcrystals. This indicates that the precipitates have started to form crystals. Fig. 3 provides a few sample images under this category.

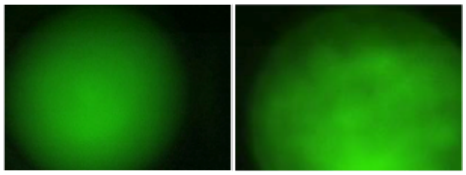

Fig. 1. Clear Drop

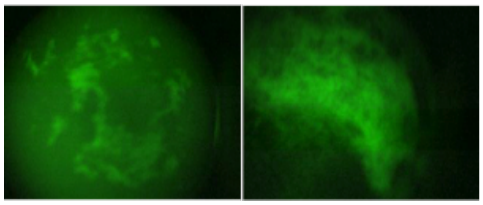

Fig. 2. Regular Granular Precipitate

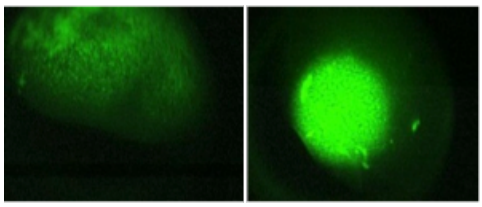

Fig. 3. Granular Precipitate or Microcrystals

2) Unclear Bright Regions: This category includes images with bright regions suggesting formation of crystals. However, the shapes of crystals are not clear. The images could be affected by conditions such as improper focusing, and lighting, etc. Since high intensity might indicate the presence of crystals, those images should be double checked by experts. Fig. 4 shows some sample images in this category.

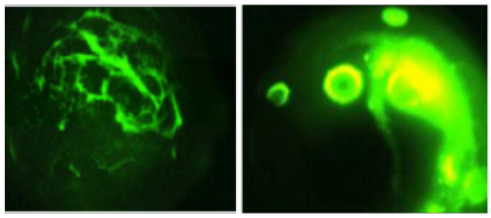

Fig. 4. Unclear Bright Regions

III. OVERVIEW OF FEATURES

For feature extraction, we follow the image processing steps described in [4]. Initial pre-processing steps include image resizing to 320x240 pixels, applying median filter, image thresholding, skeletonization of binary images and binary large object (blob) detection. Feature extraction depends on the quality and correctness of the binary (or thresholded) images. Different thresholding techniques can provide good results for different images. Hence, combining the results from multiple thresholding techniques is helpful. For each image, we apply 3 thresholding techniques and obtain 3 binary images. From each binary image, we extract 6 intensity related features and 9 blob related features. Therefore, we extract a total of 3*(6+9) = 45 features per image.

A. Intensity features

- Threshold intensity
- Number of white pixels in the binary image
- Average image intensity in the foreground region

- Standard deviation of intensity in the foreground region
- Average image intensity in the background region
- Standard deviation of intensity in the background region

B. Region (Blob) features

- Number of blobs
- Area of the largest blob
- The largest blob fullness
- The largest blob boundary pixel count
- The largest blob boundary uniformity measure
- The largest blob uniformity measure
- The largest blob measure of symmetry
- Average area of the top 5 largest blobs excluding largest blob
- Average fullness of the top 5 largest blobs excluding largest blob

IV. PREPROCESSING AND CLASSIFICATION TECHNIQUES

Our study includes an extensive investigation and comparison of various techniques to reach the optimal classification results. These techniques are categorized into 2 groups. The first group is the preprocessing techniques that consist of two normalization methods and PCA data reduction technique. The classifiers are described in the second group. We select some state-of-art classification methods in the literature from different approaches such as probabilistic, categorical, linear, and ensemble classifiers. In this way, we aimed to cover all possible solutions and find the best one for this specific problem. The following parts of this section describe the normalization methods, principal component analysis and classification techniques that are used in our experiments.

A. Normalization

Normalization is a data transformation method that maps the data into a specific range. That transformation may affect the efficiency and the accuracy of classifiers. While some classifiers such as neural networks benefit from normalization significantly, the normalization of data may not affect some classifiers such as naive Bayesian and decision trees. Particularly, the algorithms that are using distance measures may produce reasonable results with normalization, because the distance metrics may produce meaningful values after normalization. Moreover, the normalization may improve the classification accuracy of the neural networks (NNW) since it accelerates the training stage [7].

In our study, we use min-max and z-score normalization in order to evaluate their effects on classifiers for our dataset. Min-max normalization maps the original data (OD) range $[\min, \max]$ into new range as $[\min', \max']$ in a linear fashion. We map our dataset into range of $[0,1]$ in our experiments. In min-max normalization, the new value (v') of current data (v) is calculated as in (1) [8]:

$$v' = \frac{\mathrm{v}-\min}{\max-\min}(\max' - \min') + \min' \quad (1)$$

In z-score normalization, the data is normalized with respect to its mean (μ_v) and standard deviation (σ_v). The new value $(\mathrm{new_v})$ of original data is calculated as in (2) [8]:

$$v' = \frac{\mathrm{v}-\mu_v}{\sigma_v} \quad (2)$$

B. Data Reduction using Principal Component Analysis

In many classification problems, there exist one or more features in the dataset that do not have distinctive properties for separation. Some features may be highly correlated or completely irrelevant to the sample. That is why, data reduction techniques are offered to eliminate these useless features. PCA is one of the famous approaches to reduce the dimensionality [9]. Basically, it transforms the complete dataset into a new space of linearly uncorrelated attributes using orthogonal transformation. PCA is done by eigenvalue decomposition of a correlation matrix such that the eigenvector of the highest eigenvalue captures the largest possible information or variance about the dataset. In this manner, a subset of most informative eigenvectors (or principal components) is selected. Using this subset, the original dataset is transformed into a lower dimensional space in which every data sample is represented by a smaller feature vector.

In this study we reduce the number of features from 45 to 3, 5, 7, 9, and 11, respectively. And then we classify the dataset using these numbers of features.

C. Classifiers

The results of the classification really depend on the structure of the dataset. Different types of datasets may require different types of classifiers [10]. For this reason, in this study, we examined 6 different classifiers to determine which one offers the best classification results for our datasets. Selected classifiers are described below with their characteristics.

1) Random Forest (RF): RF is an ensemble classifier that includes many number of decision tree classifiers that are generated from different subsets of features and number of training samples. By combining the results of all decision trees RF predicts the final class based on a voting mechanism. We included this method into the paper because it is one of the most powerful classifiers and it can assign importance values to the features [10].

2) Naïve Bayesian Classifier (BYS): BYS is a probabilistic classifier technique that decides the class of the instance by providing the probability of membership to the classes. The class with the highest probability is predicted as the result class. Naïve Bayesian classifier is selected since it is robust to the noise, training stage is fast and

classification is independent from the range of the feature values [10].

3) Support Vector Machines: SVM is one of the most powerful classification techniques in the literature. In the training stage, it tries to find the samples that maximize the decision boundary between classes. These samples are called "support vectors" that help to decide the position of the classification border. In this paper, SVM is selected because it is a robust linear classifier [10].

4) Artificial Neural Network (NNW): NNW is a computational training model that is originally inspired from human neural system (particularly brain). The NNW model is formed by different number of interconnected neurons that are communicated with each other. We select NNW in this study because it can learn most datasets effectively in the training stage even if the classes are nonlinearly distributed. Although there are some drawbacks, NNW is commonly used technique for various classification problems [10].

5) Decision Tree: ID3 is a classification technique that uses a tree-based graph of features to separate the classes of the samples. In the training stage, ID3 creates leaf nodes based on the feature outcomes. Therefore, it provides good performance for categorical data types. It can be utilized as a rule based classifier that requires relatively less time to create training model. Furthermore, testing is also quite fast after building the decision tree [10].

6) Linear Discriminant Analysis: LDA is also one of the fundamental techniques in the literature for dimensionality reduction and classification problems. Similar to PCA, LDA transforms the data into a new space. But unlike PCA, it takes the class labels into account. It targets to maximize in-class similarity while minimizing inter-class similarity in the new space. After transforming the data, a linear decision border is applied for classification. LDA is relatively weak approach compared to the other classification methods but it can be considered as a benchmark classifier since it is commonly used technique in many areas [10].

V. EXPERIMENTS

In this section, the classification operation is performed for specific portion of our protein crystallization images dataset. Our dataset contains 2250 images of 3 major classes (67% noncrystal, 18% likely leads, and 15% clear crystal images) of proteins. However, in this study, we consider the classification of subcategories of noncrystals and likely leads, independently. Since the crystal category may require a different feature set, it is not covered in this paper. Although noncrystals consist of 3 subcategories, which are mentioned, in Section 2, we will use 2 subcategories of noncrystals, because one of its subclasses is quite rare. Therefore, both datasets have two subsets to classify. Since there are only two classes to classify for both datasets, we may adequately use binary classifiers to solve this problem.

In this classification problem, we apply the classifiers mentioned in the previous section. An extensive comparison has been presented for different cases. We rerun the experiments by using different normalization techniques and different number of principal components (PCs). Since we use random and stratified sampling, we repeat the experiments 5 times for each case to provide more reliable results. The minimum (min), maximum (max) and average accuracies of each classifier are presented in the tables in the following subsections. At the end of this section, a summary of the results is provided.

A. Noncrystal Classification

Our noncrystal dataset consists of 1606 observations and 87% of those samples belong to clear drop and the remaining part belongs to regular granular precipitate. The dataset is fragmented into two parts: training set and testing set. The training set contains 75% of the samples and the testing set consists of 25% of the samples. We repeat our experiments 5 times on original, min-max and z-score normalized form of dataset. Table I shows minimum, maximum and mean values of accuracies for each classification technique with respect to data transformation (Data Trans.) and Fig. 5 presents the changes of classifiers with the normalization.

According to the results in the Table I, random forest gives the best accuracy in all classifiers. It can reach 97% accuracy on the average. On the other hand, Fig. 5 shows that normalization of data improves support vector machine, random forest, and neural network accuracies, but neural network shows the most significant improvement by normalizing the data. It gives the best results with z-score normalization. In addition, naïve bayesian classifier is not affected by normalization. Results of decision tree and linear discriminant analysis are inconsistent with respect to data normalization.

We also apply PCA technique to our dataset in order to evaluate the effects of the feature reduction on different classifiers. The results are shown in Table II, with respect to the number of principal components.

TABLE I. COMPARISON OF DIFFERENT NORMALIZATION TECHNIQUES ON CLASSIFIERS FOR NONCRYSTAL DATASET

Data Trans.		**RF**	**BYS**	**SVM**	**NNW**	**ID3**	**LDA**
OD	**min**	**0.953**	0.870	0.913	0.868	0.933	0.913
	max	**0.975**	0.870	0.933	0.870	0.958	0.925
	mean	**0.962**	0.870	0.925	0.870	0.945	0.921
MM	**min**	**0.953**	0.870	0.925	0.870	0.925	0.910
	max	**0.970**	0.870	0.940	0.930	0.943	0.925
	mean	**0.962**	0.870	0.931	0.909	0.935	0.919
ZS	**min**	**0.960**	0.870	0.910	0.943	0.935	0.910
	max	**0.973**	0.870	0.945	0.955	0.968	0.928
	mean	**0.968**	0.870	0.931	0.950	0.949	0.919

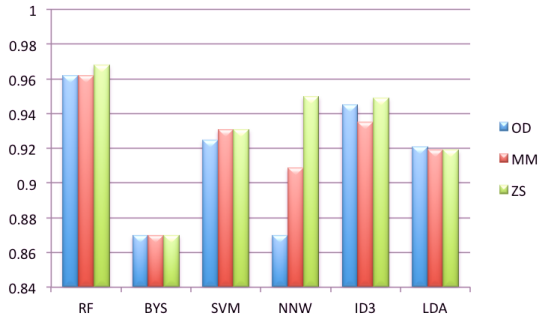

Fig. 5. Effects of data normalization on classifiers for noncrystal dataset

TABLE II. COMPARISON OF DIFFERENT NUMBER OF PRINCIPAL COMPONENTS ON CLASSIFIERS FOR NONCRYSTAL DATASET

#PC	RF	BYS	SVM	NNW	ID3	LDA
3	0.894	**0.860**	0.542	0.872	0.877	**0.835**
5	0.894	0.852	0.680	0.861	**0.880**	0.828
7	**0.896**	0.845	0.737	0.782	0.868	0.830
9	0.889	0.753	0.752	0.873	0.849	0.808
11	0.892	0.707	**0.778**	**0.876**	0.858	0.817

According to the results in Table II, random forest gives the best results with 89% accuracy on average. However, the increasing number of principal components does not affect random forest consistently; it improves support vector machine and it affects naïve Bayesian adversely. The results for the rest of the classifiers do not yield consistent results with respect to the principal component analysis. We had the best results for random forest with the 7 principal components (PCs), for Bayesian with the 3 PCs, for support vector machine with 11 PCs, for neural network with the 11 PCs, for decision tree with the 5 PCs, and for linear discriminant analysis with the 3 PCs. Our results imply that there is no consistent number of PCs that provide the best accuracy for all classifiers.

B. Likely Leads Classification

In our likely leads dataset, we have 245 observations. 70% of those observations belong to unclear bright images category and 30% of those belong to granular precipitate or microcrystals category. Similar to noncrystals, the likely leads dataset is split into two subsets as training set and testing set. We use 75% of samples as training set and 25% of samples as testing set. We rerun our experiments 5 times on original, min-max and z-score normalized datasets. The minimum, maximum and mean values of accuracies of each classifier are shown in Table III and Fig. 6 represents the changes in the classifier with the normalization.

According to Table III, random forest gives the best result with 93% accuracy. Fig. 6 shows that normalization of the data improves the accuracy of random forest and neural network classifiers. Even if neural network classifier does not have the best accuracy, it shows better improvement than random forest. We can say that the normalization does not affect the rest of classifiers consistently.

TABLE III. COMPARISON OF DIFFERENT NORMALIZATION TECHNIQUES ON CLASSIFIERS FOR LIKELY LEADS DATA SET

Data Trans		RF	BYS	SVM	NNW	ID3	LDA
OD	**min**	**0.902**	0.721	0.869	0.295	0.852	0.525
	max	**0.967**	0.885	0.918	0.770	0.902	0.820
	mean	**0.928**	0.810	0.895	0.649	0.875	0.721
MM	**min**	**0.902**	0.787	0.852	0.852	0.836	0.590
	max	**0.967**	0.918	0.934	0.967	0.951	0.754
	mean	**0.934**	0.846	0.892	0.911	0.895	0.689
ZS	**min**	**0.918**	0.770	0.869	0.902	0.803	0.656
	max	**0.984**	0.852	0.984	0.984	0.934	0.852
	mean	**0.948**	0.803	0.911	0.941	0.862	0.734

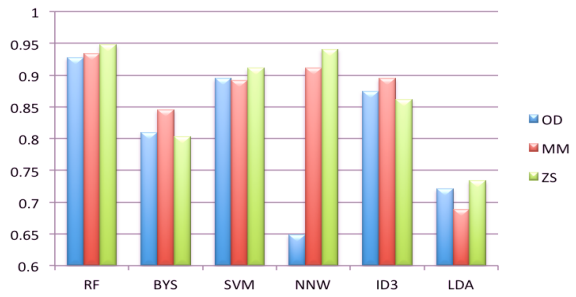

Fig. 6. Effects of data normalization on classifiers for likely leads dataset

To evaluate effects of number of principal components on different classifiers for likely leads dataset, we present our results of experiments with respect to the number of principal components in Table IV.

TABLE IV. COMPARISON OF DIFFERENT NUMBER OF PRINCIPAL COMPONENTS ON CLASSIFIERS FOR LIKELY LEADS DATASET

#PC	RF	BYS	SVM	NNW	ID3	LDA
3	0.754	0.590	0.689	0.721	0.754	0.623
5	**0.803**	0.639	0.721	**0.770**	**0.820**	0.656
7	0.787	0.525	**0.787**	0.770	0.803	**0.689**
9	0.754	**0.656**	0.787	0.770	0.754	0.574
11	0.738	0.639	0.705	0.672	0.721	0.689

According to the our results in Table IV, we obtained the best accuracy for random forest with the 5 PCs, for bayesian with the 9 PCs, for support vector machine with the 7 PCs, for neural network with 5 PCs, for decision tree with 5 PCs and for linear discriminant analysis with 7 PCs. The results state that we get the best accuracy with 5 or 7 PCs for all classifiers except for naive Bayesian classifier. The results also show the increasing the number of PCs improves accuracy of support vector machine but it does not affect the results of the rest of the classifiers consistently.

C. Summary

According to the results in Tables I and II, random forest gives satisfactory results for likely leads and noncrystal dataset with those features mentioned in Section 3. Normalization improves random forest and neural network accuracies for both datasets, but the improvement of neural network is more significant than the improvement of random forest. Random forest and neural network give the best result with z-score normalized data for both datasets. If we use

Bayesian, decision tree as classifiers, normalization does not show consistent improvement on the results of those classifiers.

According to Tables III and IV, applying PCA to both dataset does not improve results in Tables I and II, but PCA results for noncrystals are better than likely leads. Random forest gives the best results with 89.6% accuracy on 7-PC noncrystal dataset and decision tree could reach 82% accuracy on 5-PC likely leads dataset. However, we cannot recommend any specific number of PCs for both dataset due to inconsistent results of classifiers.

We have received the best accuracy with random forest classifier using z-score normalization for both datasets. Z-score normalization improved 0.6% to 96.8% for non-crystal dataset and 2% to 94.8% for the likely leads dataset. On the other hand, z-score normalization improved accuracy of neural networks 8% to 95% for non-crystal dataset and improved 29.2% to 94.1%.

PCA did not improve our results. This may be due to our feature set. This may be an indication that our feature set has little to none linear correlation among features.

VI. Conclusion

We performed our experiments with 6 different classifiers on original, normalized and reduced datasets. The experimental results suggested choosing random forest generally gives satisfactory results with normalization on both noncrystal and likely leads dataset. The experiments in this paper also demonstrate a) normalization of data improve the accuracies of random forest, neural network, and support vector machines on the noncrystals dataset, b) it improves random forest and neural network on the likely leads data set, and c) z-score normalization gives the best results for those classifiers. Although neural network does not give the highest accuracy, it shows a significant improvement on both dataset with normalization. We have acquired fairly good accuracy results for sub-categories of non-crystal and likely leads. After applying principal component analysis, RF and ID3 give the best results for noncrystals and likely leads datasets, respectively. Nonetheless, these classifiers do not show any consistent improvement with respect to increasing PCs. Increasing number of PCs may improve support vector machines on both datasets. PCA results may indicate that our feature set has minimal linear correlation, and we may rely on our feature set for protein crystallization classification. We plan to work on the crystal sub-categories as future work. Crystal categories may require a different set of features.

VII. Acknowledgement

VIII. References

[1] S. Pan et al., "Automated classification of protein crystallization images using support vector machines with scale-invariant texture and Gabor features," Acta Crystallographica Section D, vol. 62, no. 3, pp. 271-279, March 2006.

[2] B. Rupp and J. Wang, "Predictive Models for Protein Crystallization," Methods, vol. 34, no. 3, pp. 390-407, 2004.

[3] C. A. Cumbaa and I. Jurisica, "Protein crystallization analysis on the World Community Grid," Journal of Structural and Functional Genomics, vol. 11, no. 1, pp. 61-69, Jan. 2010.

[4] M. Sigdel, M. L. Pusey, and R. S. Aygun, "Real-Time Protein Crystallization Image Acquisition and Classification System," Crystal Growth & Design, vol. 13, no. 7, pp. 2728-2736, 2013.

[5] M. Bern, D. Goldberg, R. C. Stevens, and P. Kuhn, "Automatic classification of protein crystallization images using a curve-tracking algorithm," Journal of Applied Crystallography, vol. 37, no. 2, pp. 279--287, April 2007.

[6] G. Spraggon, S. A. Lesley, A. Kreusch, and J. P. Priestle, "Computational analysis of crystallization trials," Acta Crystallographica Section D, vol. 58, no. 11, pp. 1915-1923, November 2002.

[7] L.A. Shalabi, Z. Shaaban, and B. Kasasbeh, "Data Mining: A Preprocessing Engine," Journal of Computer Science, vol. 2, no. 9, pp. 735-739, 2006.

[8] J. Shlens, "A tutorial on principal component analysis," Systems Neurobiology Laboratory, University of California at San Diego, 2005.

[9] Pang-Ning T., Michael S., and Vipin K., Introduction to Data Mining. Boston, MA, USA: Addison-Wesley Longman Publishing Co. Inc., 2005.

[10] Jiawei H., Micheline K., and Jian P., Data mining: concepts and techniques. San Francisco, CA, USA: Morgan kaufmann, 2006.