CHAPTER ONE

INTRODUCTION

1.1 Introduction

The last decade has seen significant growth in concern of Artificial Intelligence and Machine Learning. In the broadest sense, these fields aim to `learn something useful' about the environment within which the organism operates. How gathered information is processed leads to the development of algorithms {how to process high dimensional data and deal with uncertainty. In the early stages of research in Machine Learning and associated areas, similar techniques were discovered in relatively isolated research communities [1].

Machine learning is a set of methods that can automatically discover patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decision making under uncertainty such as planning how to collect more data.

Broadly speaking the main two subfields of machine learning are supervised learning and unsupervised learning. In supervised learning the focus is on precise prediction, whereas in unsupervised learning the aim is to find accurate compact descriptions of the data. Particularly in supervised learning, one is interested in methods that perform well on previously unseen data. That is, the method `generalises' to unseen data. In this sense, one distinguishes between data that is used to train a model, and data that is used to test the performance of the trained model.

The kernel based machine learning methods are considered one of the most used machine learning methods particularly in the biological data, these methods include support vector machine (SVM) and kernel logistical regression (KLR). Most of the biological data needs preprocessing for the machine learning method to give accurate results. In this research we will use clustering approach as preprocessing step to enhance the accuracy of the result we will use the kernel based approaches SVM and KLR in our research. Moreover we will test the effect of clustering when the dataset are very large or there is a rear event.

Proteins play a key role in almost all biological processes. They take part in, for example, maintaining the structural integrity of the cell, transport and storage of small molecules, catalysis, regulation, signaling and the immune system. Linear protein molecules fold up into specific three-dimensional structures, and their functional properties depend intricately upon their structures. As a result, there has been much effort, both experimental and computational, in determining protein structures. Protein

structures are determined experimentally using either x-ray crystallography or nuclear magnetic resonance (NMR) spectroscopy [8]. While both methods are increasingly being applied in a high-throughput manner, structure determination is not yet a straightforward process. X-ray crystallography is limited by the difficulty of getting some proteins to form crystals, and NMR can only be applied to relatively small protein molecules. As a result, whereas whole-genome sequencing effort have led to large numbers of known protein sequences, their corresponding protein structures are being determined at a significantly slower pace. On the other hand, despite decades of work, the problem of predicting the full three-dimensional structure of a protein from its sequence remains unsolved. Nevertheless, computational methods can provide a first step in protein structure determination, and sequence-based methods are routinely used to help characterize protein structure.

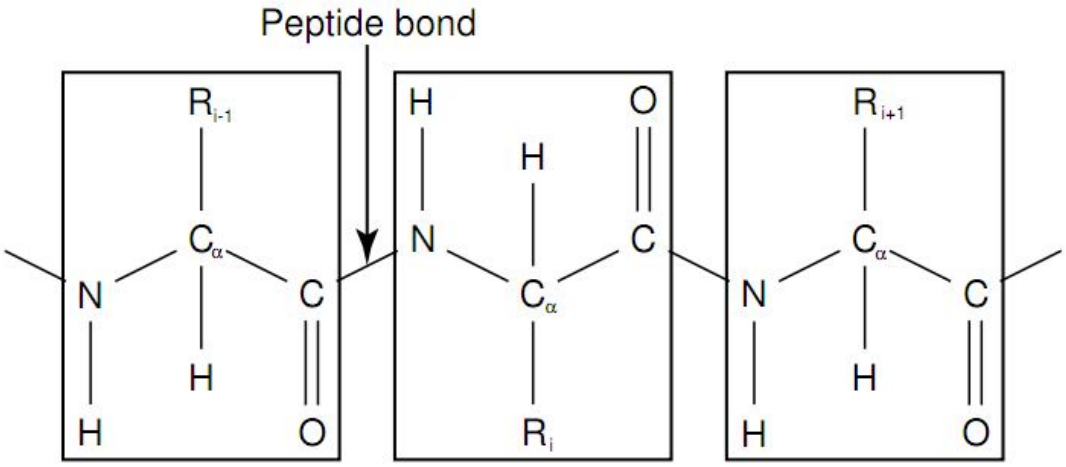

Figure 1: Proteins are polymers of amino acids.

Each amino acid has the same Fundamental structure (boxed), differing only in the atoms making up the side chain. Here, the i_{th} side chain in the protein sequence is designated by R_i. The carbon atom to which the amino group, carboxyl group, and side chain are attached is called the alpha carbon (Cα). Two amino acids i − 1 and i are linked linearly through a peptide bond between the carboxyl group of amino acid i − 1 and the amino group of amino acid i; a water molecule is removed in the process of bond formation. Aspects of protein structure [8]. A protein molecule is formed from a chain of amino acids. Each amino acid consists of central carbon atom (Cα), and attached to this carbon are a hydrogen atom, an amino group (NH2), a carboxyl group

(COOH) and a side chain that characterizes the amino acid. The amino acids of a protein are connected in sequence with the carboxyl group of one amino acid forming a peptide bond with the amino group of the next amino acid (Figure1) Successive bonds make up the protein backbone, and the repeating amino-acid units (all called residues) within the protein consist of both the main-chain atoms that comprise the backbone as well as the side-chain atoms. [8]

1.2 Research scope

The scope of this research is in the area of machine learning methods concerning the biological data. The idea is to use the preprocessing approach for the biological data to calculate their features and the clustering approach such as k-mean clustering as preprocessing steps to prepare the data for the SVM and KLR. The biological data that will be used in this research is the data of proteins. The goal is to predict their secondary structure which can be alpha-helix, beta-sheets, or coils. The datasets that will be used is the RS126 is set of 126 nonhomologous globular protein chains, used in the experiment of Rost and Sander and referred to as the RS126 set, was used to evaluate the accuracy of the predictors.

1.3 Problem statement

Determining the secondary structure of protein in the laboratory is very costly and consumes a lot of time. Therefore computerized methods are needed to determine the structure of protein. There are different methods designed for protein structure prediction. In this research we will use data preprocessing using the clustering approach to solve unbalanced data set problem for protein secondary structure prediction.

1.4 Objective of the research

We proposed to design ever effective machine learning method that can be used in the protein structure features prediction. To test the effectiveness of data preprocessing such as clustering for solving unbalanced data set problem.

1.5 Research methodology and tools

RS126 dataset is downloaded from the web. The features of this data set will be obtained using NCBI blast program. We accomplished this with writing a program in VB.net to calculate the features of all the protein automatically using a local copy of NCBI blast. The preprocessing for the data will be done using K-means clustering to prepare clusters that can be used as input for a support vector machines (SVM) and kernel logistic (KLR) regression algorithms. These algorithms will be written in MATLAB.

1.6 Research questions

1. Is the machine learning methods predicting biological data accurately?
2. Is the clustering as a preprocessing step can solve problem of unbalanced data set?
3. Is the clustering affect the accuracy of the machine learning methods prediction results?
4. What are the appropriate measurements that can be used to calculate the accuracy of the results?

1.7 Research organization

This research is organized as follows: Chapter one is the Introduction which highlight a brief history of machine learning. Chapter two is the Literature review and related work. In chapter three we present Material and Methods. In chapter four results and discussion are presented. And in is the Chapter five Conclusion and Recommendation.

CHAPTER TWO

LITERATURE REVIEW AND RELATED WORK

2.1.1 Machine Learning

Machine learning is topic that studies computer algorithms for learning to solve prediction and classification problem. We might, for instance, be interested in learning to complete a task, or to make accurate predictions, or to behave intelligently. The learning that is being done is always based on some sort of observations or data. In general, machine learning is about learning to do better in the future base on what was experienced in the past. The emphasis of machine learning is on automatic methods. In other words, the goal is to devise learning algorithms that do the learning automatically without human intervention or assistance. The machine learning paradigm can be viewed as "programming by example." Often we have a specific task in mind, such as spam filtering. But rather than program the computer to solve the task directly, in machine learning, we seek methods by which the computer will come up with its own program based on examples that we provide [25].

Machine learning is a core sub-area of artificial intelligence. It is very unlikely that we will be able to build any kind of intelligent system capable of any of the facilities that we associate with intelligence, such as language or vision, without using learning to get there. These tasks are otherwise simply too difficult to solve. Further, we would not consider a system to be truly intelligent if it were incapable of learning since learning is at the core of intelligence [25].learning is two types supervised learning and unsupervised learning.

2.1.1Supervised learning

Given a set of data D = {(x^n, y^n), n = 1…….N} the task is to `learn' the relationship between the input x and output y such that, when given a new input x*the predicted output y* is accurate. To specify explicitly what accuracy means one defines a loss function L (yˆpred, yˆtrue) or, conversely, a utility function U = -L.

In supervised learning our interest is describing y conditioned on knowing x. From a probabilistic modeling perspective, we are therefore concerned primarily with the conditional distribution p (y|x,D). The term `supervised' indicates that there is a `supervisor' specifying the output y for each input x in the available data D. The output is also called a `label', particularly when discussing classification.

2.1.1 Unsupervised learning

Given a set of data D = { x^n, n = 1….N} in unsupervised learning we aim to `learn' a possible compact description of the data. An objective is used to quantify the accuracy of the description. In unsupervised learning there is no special `prediction' variable. From a probabilistic perspective we are interested in modeling the distribution p(x) **[1]**.

2.1.1Utility and Loss

To more fully specify a supervised problem we need to be clear what `cost' is involved in making a correct or incorrect prediction. In a two class problem the classes will be in the set c = {1, 2}, we assume here that everything we know about the environment is contained in a model p(x, c). Given a new input x*, the optimal prediction also depends on how costly making an error is. This can be quantified using a loss function (or conversely a utility). In forming a decision function c(x*) that will produce a class label for the new input x*, we don't know the true class, only our presumed distribution p (c|x*) **[1]**. The expected utility for the decision function is:

$$U(c(x*)) = \sum_{c^{true}} U(c^{true}, c(x^*))\, p(c^{true} \mid x^*) \qquad (1)$$

And the optimal decision is that which maximizes the expected utility.

2.1.1 Zero-one loss

A `count the correct predictions' measure of prediction performance is based on the `zero-one' utility (or conversely the zero-one loss):

$$U(c^{true}, c^*) = \begin{cases} 1 \text{ if } c^* = c^{true} \\ 0 \text{ if } c^* \neq c^{true} \end{cases} \qquad (2)$$

For the two class case, we then have

$$U(c(x^*)) = \begin{cases} p(c^{true} = 1 \mid x^*) \text{ for } c(x^*) = 1 \\ p(c^{true} = 2 \mid x^*) \text{ for } c(x^*) = 2 \end{cases} \qquad (3)$$

Hence, in order to have the highest expected utility, the decision function c (x*) should correspond to selecting the highest class probability p (c|x*):

$$c(x^*) = \begin{cases} 1 \text{ if } p(c = 1 \mid x^*) \geq 0.5 \\ 2 \text{ if } p(c = 2 \mid x^*) \geq 0.5 \end{cases} \qquad (4)$$

In the case of a tie, either class is selected at random with equal probability.

2.1.1 General loss functions

In general, for a two-class problem, we have

$$U(c(x^*)) = \begin{cases} U(c^{true}=1, c^*=1)p(c^{true}=1|x^*) + U(c^{true}=2, c^*=1)p(c^{true}=2|x^*) \text{for } c(x^*)=1 \\ U(c^{true}=1, c^*=2)p(c^{true}=1|x^*) + U(c^{true}=2, c^*=2)p(c^{true}=2|x^*) \text{for } c(x^*)=2 \end{cases}$$

And the optimal decision function c (x*) chooses that class with highest expected utility.

One can readily generalise this to multiple-class situations using a utility matrix with elements

$$U_{i,j} = U(c^{true} = i, c^{pred} = j) \quad (5)$$

Where the i, j element of the matrix contains the utility of predicting class j when the true class is i. Conversely one could think of a loss-matrix with entries $L_{ij} = -U_{ij}$. The expected loss with respect to p (c|x) is then termed the risk **[1]**.

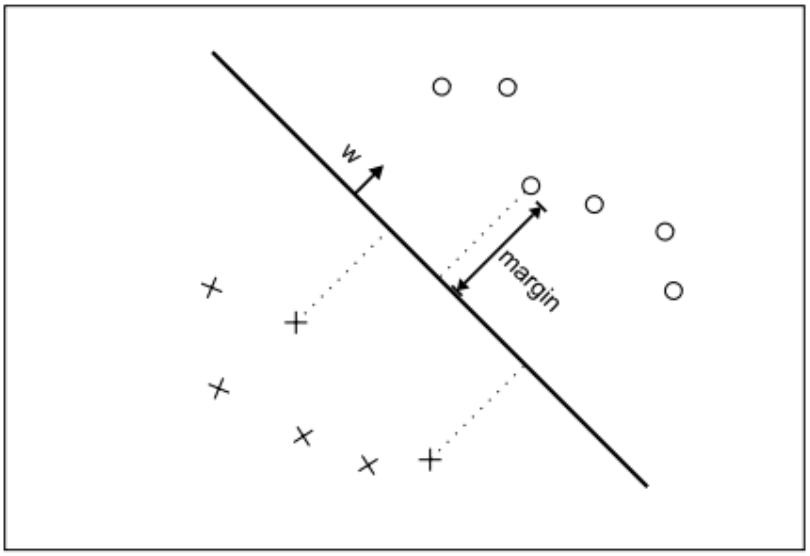

Figure 2: A simple linear support vector machine

2.1.2 Support Vector Machines

Support vector machines (Vapnik, 1982) have strong theoretical foundations and excellent empirical successes. They have been applied to tasks such as handwritten digit recognition, object recognition, and text classification [3]

2.1.2 SVMs for Induction

We shall consider SVMs in the binary classification setting. We are given training data $\{ x_1 \ldots x_n \}$ that are vectors in some space $X \times \mathbb{R}^d$. We are also given their labels $\{ y_1 \ldots y_n \}$ where $y_i \in \{-1, 1\}$. In their simplest form, SVMs are hyperplanes that separate the training data by a maximal margin (see Fig. 2a). All vectors lying on one side of the hyperplane are labeled as −1, and all vectors lying on the other side are labeled as 1. The training instances that lie closest to the hyperplane are called support vectors, More generally, SVMs allow one to project the original training data in space X to a higher dimensional feature space F via a Mercer kernel operator K [3]. In other words, we consider the set of classifiers of the form:

$$f(X) = \left(\sum_{i=1}^{n} \alpha_i K(X_i, X) \right).$$

When K satisfies Mercer's condition (Burges, 1998) we can write:

$K(u, v) = \Phi(u) \cdot \Phi(v)$ where $\Phi: X \rightarrow F$ and "·" denotes an inner product. We can then rewrite f as:

$$f(x) = w.\Phi(X), \text{ where } w = \sum_{i=1}^{n} \alpha_i \Phi(X_i).$$

Thus, by using K we are implicitly projecting the training data into a different (often higher dimensional) feature space F. The SVM then computes the α_i s that corresponds to the maximal margin hyperplane in F. By choosing different kernel functions we can implicitly project the training data from X into spaces F for which hyperplanes in F correspond to more complex decision boundaries in the original space X. Two commonly used kernels are the polynomial kernel given by $K(u, v) = (u \cdot v + 1)^p$ which induces polynomial boundaries of degree p in the original space X_1 and the radial basis function kernel also called Gaussian kernel $K(u, v) = (e^{-\gamma (u-v) \cdot (u-v)})$ which induces boundaries by placing weighted Gaussians upon key training instances.

They are will assume that the modulus of the training data feature vectors are constant, i.e., for all training instances x*i*, $\| \Phi (xi) \| = \lambda$ for some fixed λ. The quantity $\| \Phi (xi) \|$ is always constant for radial basis function kernels, and so the assumption has no effect for this kernel. For $\| \Phi (xi) \|$ to be constant with the polynomial kernels we require that $\| xi \|$ be constant [3].

2.1.3 Kernel logistic regression

KLR is the kernel version of logistic regression that allows non-linear probabilistic classification by constructing the logistic regression in higher dimensional feature space using kernel function K: $\chi \times \chi \rightarrow F$ [22]. The kernel function evaluates the inner product between the input vectors in the feature space i.e. $K(x, x') = \phi(x).\phi(x')$, where $x \in \chi \in R^D$ The KLR can be constructed in the feature space such that

$$P_r(Y = -1 \mid X = x, w) = \frac{e^{w^T \phi(X)+b}}{1+e^{w^T \phi(X)+b}}$$

$$P_r(Y = 1 \mid X = x, w) = pli = \frac{1}{1+e^{w^T \phi(X)+b}} \qquad (1)$$

Where *w* is the KLR parameters and b is the intercept term. The panelized negative log likelihood (PNLL) is normally used to infer KLR parameters and it can be defined in the primal weight space as follows [22]:

$$\min_{w,b} \frac{1}{2} w^T w + \frac{v}{2} \sum_{i=1}^{N} \log(1+\exp(-y_i(w^T \phi(x_i)+b))) \qquad (2)$$

where v is the penalty term. One of the most popular techniques used to find the maximum-likelihood estimation (MLE) for the parameters of the LR model is the iteratively re-weighted least squares (IRLS) method, which use Newton Raphson algorithm The same method can be used for KLR in the primal weight space in which the solution for *w* on the $(c+1)_{th}$ iteration using Newton-Raphson update can be given as in the following equation, given that we normally start from initial parameter w^0

$$_{w}(c+1)_{=w}(c)_{+s}(c+1) \qquad (3)$$

Where $s^{(c+1)}$ in each iteration is determined by the following minimization problem

$$q^{c+1} = \min_{S(c+1)} \frac{1}{2}(S^{(c+1)} + w^{(c)})^T (S^{(c+1)} + w^{(c)}) + \frac{V}{2}\sum_{i=1}^{N} g_i^{(c+1)}(e_i^{(c+1)})^2 ,$$

Such that

$$(S^{(c+1)})^T \phi(x_i) + b = z_i^{(c+1)} - e_i^{(c+1)}, i = 1,2,...N \qquad (4)$$

$$Where \quad z_i^{(c+1)} = \frac{(p_{yi\,i}^{(c+1)} - 1)yi}{g_i^{(c+1)}} \qquad and\ g_i^{(c+1)} = p_{1i}^{(c+1)}(1 - p_{1i}^{(c+1)}).$$

The solution to equation (4) at iteration (c+1) is given by the following dual problem

$$\left(\frac{\Omega + \frac{1}{V}(W^{(c+1)})^{-1}}{I_N^T} \;\middle|\; \frac{1_N}{0}\right)\begin{pmatrix}\alpha^{(c+1)} \\ b^{(c+1)}\end{pmatrix} = \begin{pmatrix}z^{(c-1)} + \Omega\alpha^{(c)} \\ 0\end{pmatrix} \qquad (5)$$

Where

$$1_N = (1,1,...,1)^T, 1_N \in R^N, z^{(c+1)} = (z_1^{(c+1)}, z_2{}^{(c+1)}, ..., z_N{}^{(c+1)})^T,$$
$$\alpha^{(c+1)} = (\alpha_1^{(c+1)}, \alpha_2^{(c+1)}, ..., \alpha_N^{(c+1)})^T, \Omega_{ij} = K(x_i, x_j), w^{(c+1)} = diag(g_1^{(c+1)}, g_2^{(c+1)}), ..., g_N^{(c+1)})^T$$

The IRLS method for large scale problem is computationally expensive, because the linear system in equation (5) must be solved for each Newton's iteration. To reduce the computation cost of IRLS we can adopt Eigen decomposition of the kernel matrix K in the form.

$$\boldsymbol{K = P\Lambda P'} \quad \boldsymbol{(6)}$$

Where $\Lambda = diag(\lambda_i), \lambda_1 \geq \lambda_2 \geq ..... \geq 0$ the Eigen values of the matrix K and *P* is the matrix of the eigenvectors that correspond to the eigenvalues. We can select the first *p*

eigenvectors and eigenvalues from the matrices P and Λ respectively, where $p \ll N$ to approximate the Eigen decomposition matrix given in equation (6). This approximation is motivated by its widely usage e.g. principal component analysis. Using this approximation the computational cost can be reduced dramatically. However computing the Eigen decomposition itself is also computationally expensive. Nystrom method can be used to reduce the computation cost of computing the eigendecomposition by selecting small sample of size $M \ll N$ from the training data to create the Eigen problem of equation (6). Then the required eigenvectors and eigenvalues at all N points can be approximated as:

$$\tilde{\lambda}_i(N) = \frac{N}{M}\lambda_i(M), \tilde{P}_i(N) = \sqrt{\frac{M}{N}}\frac{1}{\lambda_i^{(M)}} K_{N \times M} p_i(l) \qquad (7)$$

The selected $M \ll N$ from the features matrix X should minimize the mean squares error or in another words it should contain as much information as possible. Since the Nystrom low-rank approximation depends crucially on the quantization error induced by encoding the sample set with landmark points one, can simply use the clusters obtained with K-means algorithm with outliers' removal as a selected vectors. The computation time of the KLR using Nystrom and K-mean clustering scales to O (NM^2) whereas the computation time of the SVMs is O (N^3)[22] .

2.1.4 K-means Clustering Algorithm

The k-mean clustering technique is simple, and we begin with description of the basic algorithm .we first choose k initial centroids , where k is a user-specified parameter, namely, the number of clusters desired each point is then assigned to the closest centroid, and each collection of points assigned to a centroid is a cluster . The centroid of each cluster is then updated based on the points assigned to the cluster. We repeat the assignment and update steps until no point change cluster, or equivalently, until the centroid remain the same-mean is formally described by the following algorithm:

Algorithm basic k-means algorithm

1. Select k point as initial centroids.
2. Repeat

3. Form k cluster by assigning each point to its closest centroid.
4. Recomputed the centroid of each cluster.
5. Until centroids do not change.

2.1.4 Assigning points to the closest centroid

Assign a point to the closest centroid; we need proximity measure that quantifies the notion of "closest" for the specific data under consideration. Euclidean Distance is often used for data points in Euclidean space, while cosine similarity is more appropriate for document. However, there may be several types of proximity measure that are appropriate for a given data .for example, Manhattan Distance can be used for Euclidean data, while the Jaccard measure is often employed for document. Usually, the similarity measures used for k-mean are relatively simple since the algorithm repeatedly calculates the similarity of each point to each centroid.in some case, however, such as when the data is in low-dimensional Euclidean space. It is possible to avoid computing many of the similarities, thus significantly speeding up the k-mean algorithm [7].

2.1.4 Centroid and objective function

Step 4 of k-mean algorithm was stated rather generally as" recomputed the centroid of each cluster", since the centroid can vary, depending on the proximity measure for the data and the goal of the clustering. The goal of the clustering is typically expressed by an objective function that depends on the proximities of the points to one another or to cluster centroids; e.g., minimize the squared distance of each point to its closest centroid. Data in Euclidean space consider data whose proximity measure is Euclidean distance. For our objective functions, which measure the quality of clustering, we use the sum of the squared error (SSE), which is also known as scatter. In other words, we calculate the error of each data point, i.e., its Euclidean distance to the closest centroid, and then compute the total sum of the squared error. Given two different sets of clusters that are produced by two different runs of k-means, we prefer the one with the smallest squared error since this means that the prototypes (centroids) of this clustering are a better representation of the point in their cluster [7]. Using the notation in the following table, the SSE is formally defined as follows:

$$SSE = \sum_{i=1}^{k} \sum_{x \in ci} dist(c_i, x)^2$$

Where dist is the standard Euclidean distance between two objects in Euclidean space

Symbol	Description
X	An object
C_i	The i_{th} cluster
c_i	The centroid of cluster c_i
c	The centroid of all points
m_i	The number of object in the i_{th}cluster
m	The number of objects in the data set
k	The number of cluster

Table 1 : Description of Symbol of the sum of the squared error (SSE)

Given these assumptions, the centroid that minimizes the SSE of the cluster is the mean, using the notation in Table1 the centroid(mean) of i^{th} cluster is defined by the following Equation

$$c_i = \frac{1}{m_i} \sum_{x \in C_I} X$$

2.1.5 Majority Voting Vector

Majority voting is the simplest method for combining several SVMs. Let f_k(k=1, 2,…., K) be a decision function of the k_{th} SVM in the SVM ensemble and C_j (j=1,2,…..,C) denote a label of the j_{th} class. Then, let N_j = # {k|fk (x) = C_j }, i.e. the number of SVMs whose decisions are known to the j_{th} class [26]. Then, the final decision of the SVM ensemble f_{mv}(x) for a given test vector x due to the majority voting is determined by

$$f_{mv}(X) = \arg_j \max N_j .$$

2.1.6 The Biological Background of the PSSP

A protein sequence is a linear array of amino acids. Each amino acid consists of 3 consecutively ordered DNA bases (A, T, C, or G). An amino acid carries various kinds of information determined by its DNA combination. An amino acid is a basic unit of a protein sequence and is called a residue. There are altogether 20 types of amino acids and each type of amino acids is denoted by an English character. For example, the character "A" is used to represent the type of amino acid named Alanine. Thus, a protein sequence in the alphabetical representation is a long sequence of characters. Given a protein sequence, various evolutionary environments may induce mutations, including insertions, deletions, or substitutions, to the original protein, thereby producing diversified yet biologically similar organisms [11].

2.1.7 Types of Protein Secondary Structures

Secondary structures are formed by hydrogen bonds between relatively small segments of protein sequences. There are three common secondary structures in proteins, namely α-helix, β-sheet (strand) and coil. Figure 3 visualizes protein secondary structures. In Figure3, the dark ribbons represent helices and the gray ribbons are sheets. And the strings in between are coils that bind helixes and sheets [11].

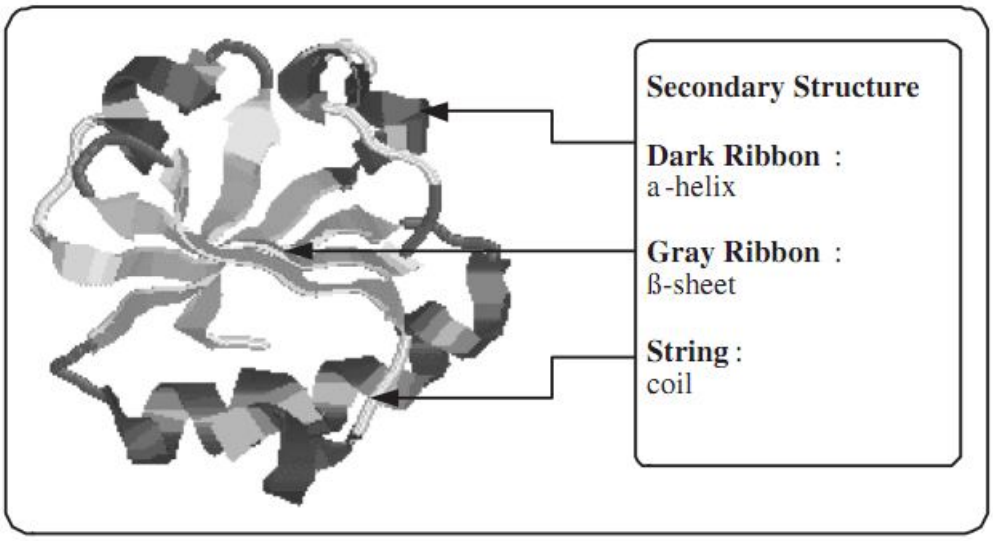

Figure 3: Three types of protein secondary structures: α-helix, β-strand, and coil

2.1.8 Secondary structure

Most commonly, the secondary structure prediction problem is formulated as follows: given a protein sequence with amino acids r1r2 . . . r_n, predict whether each amino acid $r_i$is in a α−helix (H), a β−strand (E), or neither (C). Predictions of secondary structure are typically judged via the 3-state accuracy (Q3), which is the percent of residues for which a method's predicted secondary structure (H, E, or C) is correct. Since residues in known protein structures are approximately 30% in helices, 20% in strands and 50% in neither, a trivial algorithm that always predicts C has a 3-state accuracy of 50%. The 3-state accuracy measure does not convey many useful types of information. [8]

2.1.9 DSSP algorithm

The DSSP program was designed by Wolfgang Kabsch and Chris Sander as the standard method for assigning secondary structure to the amino acids of a protein, given the atomic-resolution coordinates of the protein. DSSP is a database of secondary structure assignments for all protein entries in the Protein Data Bank (PDB). DSSP is also the program that calculates DSSP entries from PDB entries. DSSP has eight types of protein secondary structure, depending on the pattern of hydrogen bond [12]. The list bellows shows the different types of protein secondary structure in DSSP:

i) H = alpha helix

ii) B = residue in isolated beta-bridge

iii) E = extended strand, participates in beta ladder

iv) G = 3-helix (3/10 helix)

v) I = 5 helix (pi helix)

vi) T = hydrogen bonded turn

vii) S = bend

viii) L = others

These eight types are usually assigned into three larger groups: helix (G, H and I), strand (E and B) and loop (all others). In this research, DSSP used as feature class are

from the three classes, which is helix (H), strand (E) and coil (C). DSSP dataset can be obtained from the RS126 sequence data which contain secondary structures and will be implemented as the feature class to fit into SVM for prediction [12]. In this research we used this type for define secondary structure of protein.

2.1.10 STRIDE

The secondary Structural Identification method by Frishman and Argos uses an empirically derived hydrogen bond energy and phi psi torsion angle criteria to assign secondary structure. Torsion angles are given alpha-helix and beta-sheet propensities According to how close they are to their regions in Ramachandranplots .The parameters are optimized to mirror visual assignments made by crystallographers for a set of proteins. By construction, the STRIDE assignments agreed better with the expert assignments than DSSP, at least for the data set used to optimize the free parameters [26].

•Like DSSP, STRIDE assigns the shortest alpha-helix ('H') if it contains at least two consecutive *i - i+4* hydrogen bonds. In contrast to DSSP, helices are elongated to comprise one or both edge residues if they have acceptable phi-psi angles, similarly a short helix can be vetoed. Hydrogen bond patterns may be ignored if the phi-psi angles are unfavorable. The sheet category does not distinguish between parallel and Anti-parallel sheets. The minimal sheet ('E') is composed of two residues. The dihedral angles are incorporated into the final sheet assignment criterion as was done for the alpha-helix.

2.1.11 DEFINE

An algorithm by Richards and Kundrot which assigns secondary structures by matching Cα-coordinates with a linear distance mask of the ideal secondary structures. First, strict matches are found which subsequently are elongated and/or joined allowing moderate irregularities or curvature. The algorithm locates the starts and ends of α- and 310-helices, beta-sheets, turns and loops. With these classifications the authors are able to assign 90-95% of all residues to at least one of the given secondary structure classes [26].

2.1.12 PSSM Generation

To obtain Position-Specific Scoring Matrix (PSSM) profile for each protein sequence, we performed PSI-BLAST (Position-Specific Iterative Basic Local Alignment Search Tool) against a non-redundant sequence (NR) data base. BLAST is the most widely used sequence similarity tool. PSI-BLAST is an iterative database searching method that uses homologous proteins found in an iteration to build a profile to be used for searching in the next iteration. This profile incorporates sequence weighting so that several closely-related homologs detected in the database do not overwhelm the contribution of more remote homologs. The PSSM for each protein sequence has 20×L elements where L is the length of target sequence and each element represents the log-likelihood of particular residue substitution based on a weighted average of BLOSUM62 **[19]**.

2.2 Related work

In 1993 Rost and Sander [13] improved the accuracy of the 3-state secondary structure prediction by developing (PHD) method, which was based on a multi-layer back-propagation neural network. Using the 126 protein sequences (RS126) developed by themselves, achieved the overall accuracy as high as 72%.

In 1995 Salamov and Solovyev is a scored nearest neighbor secondary structure prediction (NNSSP) method by considering the position of N and C terminal in α-helices and β-strands. Its prediction accuracy on the RS126 data set was 72.7% [15].

PSIPRED [Jones, 1999] used a position-specific scoring matrix generated by PSI-BLAST to predict protein secondary structure and achieved 78.3[16].

DSC [King and Sternberg, 1996] achieved 71.1% prediction accuracy in the RS126 data set by exploring amino acid profiles, conservation weights, indels, and hydrophobicity, based on linear discrimination, [17].

In 2001, Hua and Sun [18] proposed an SVM approach. This was an early application of the SVM to the PSSP problem. In their work, they first constructed 3 one-versus-one and 3 one-versus-all binary classifiers. Three tertiary classifiers were designed based on these binary classifiers through the use of the largest response, the decision tree and

votes for the final decision. By making use of the Rost's data encoding scheme, they achieved the accuracy of 71.6% and the segment overlap accuracy of 74.6% for the RS126 data set.

George Karypis 2005 they developed a new secondary structure prediction algorithm called YASSPP that uses a pair of cascaded models constructed from two sets of binary SVM-based models. YASSPP uses an input coding scheme that combines both position-specific and non-position specific information, utilizes a kernel function designed to capture the sequence conservation signals around the local window of each residue, and constructs a second-level model by incorporating both the three-state predictions produced by the first-level model and information about the original sequence. Experiments on dataset RS126, show that YASSPP is capable of producing the highest Q3 =77.08 when they use window length 7 and produce Q3= 76.89 when they use window length 9. [9]

Jung-Ying Wang (2002) applies SVM for protein secondary structure prediction. We worked on similar data and encoding schemes as those in Rost and Sander (referred here as RS130). The performance accuracy is verified by a seven-fold cross validation [4]. Results indicate that SVM easily returns comparable results as neural networks.

Murtada Khalafallah Elbashir et al .they used KLR to obtain sparse B-turns prediction in short evolution time. Secondary structure information and position-specific scoring matrices (PSSMs) are utilized as input features. They achieved Qtotal of 80.7% and MCC of 50% on BT426 dataset. These results show that KLR method with the right algorithm can yield performance equivalent to or even better than NNs and SVMs in B-turns prediction. In addition, KLR yields probabilistic outcome and has a well-denied extension to multiclass case. [5]

Elbashir et al. Proteome Science 2013 they proposed an approach that combines support vector machines (SVMs) and logistic regression (LR) in a hybrid prediction method, which we call (H-SVM-LR) to predict β-turns in proteins. Fractional polynomials are used for LR modeling. We utilize position specific scoring matrices (PSSMs) and predicted secondary structure (PSS) as features. Our simulation studies show that H-SVM-LR achieves Qtotal of 82.87%, 82.84%, and 82.32% on the BT426, BT547, and BT823 datasets respectively. These values are the highest among other β-turns

prediction methods that are based on PSSMs and secondary structure information. H-SVM-LR also achieves favorable performance in predicting β-turns as measured by the Matthew's correlation coefficient (MCC) on these datasets. Furthermore, H-SVM-LR shows good performance when considering shape strings as additional features. [2]

CHAPTER THREE

MATERIAL AND METHODS

3.1 RS126 Dataset

The dataset used in this study is RS126. The purpose of the research is to obtain the protein sequence datasets in order to predict protein secondary structure. RS126 is one of the oldest dataset with the longest history in protein secondary structure prediction evaluation. The scheme is created by Rost and Sander. RS126 being the most commonly used datasets to predict protein structure are applied in most of the study including this research. It contains 23,347 residues with an average protein sequence length of 185. 32% of RS126 are alpha helix, 21% as beta strand and 47% as coil. RS126 dataset can be collected from various supplementary data files in previous research or study. Besides that, it can also be obtained from online database such as Protein Data Bank (PDB) [21].

PDB ID protein		
1acx	1wsyb	3hmga
1azu	256ba	3hmgb
1bbpa	2aat	3icb
1bds	2ak3a	3pgm
1bmv1	2alp	3rnt
1bmv2	2cab	3tima
1cbh	2ccya	4bp2
1cc5	2cyp	4cms
1cdta	2fox	4cpai
1crn	2fxb	4cpv
1csei	2gbp	4gr1
1eca	2gcr	4pfk
1etu	2glsa	4rhv1
1fc2c	2gn5	4rhv3
1fdlh	2hmza	4rhv4
1fdx	2i1b	4rxn

1fkf	2lhb	4sdha
1fnd	2ltna	4sgbi
1fxia	2ltnb	4ts1a
1gd1o	2mev4	4xiaa
1gdj	2mhu	5cytr
1gp1a	2or1l	5er2e
1hip	2paba	5hvpa
1il8a	2pcy	5ldh
1l58	2phh	5lyz
1lap	2rspa	6acn
1lmb3	2sns	6cpa
1mcpl	2sodb	6cpp
1mrt	2stv	6cts
1ovoa	2tgpi	6dfr
1paz	2tmvp	6hir
1ppt	2tsca	6tmne
1pyp	2utga	7cata
1r092	2wrpr	7icd
1rbp	3ait	7rsa
1rhd	3b5c	8abp
1s01	3blm	8adh
1sh1	3cd4	9apia
1tgsi	3cla	9apib
1tnfa	3cln	9insb
1ubq	3ebx	9pap
1wsya	3gapa	9wgaa

Table 2: List of 126 Proteins in RS126 dataset used in secondary structure prediction

3.2 Position Specific Scoring Matrices (PSSMs)

It has been shown that PSSMs contributed significantly to the accuracy of protein secondary structure prediction [2]. They are in the form of M*20, where M represents the sequence length. The PSSMs are generated using three rounds of the iterative PSI-BLAST program [23] against National Center for Biotechnology Information (NCBI) non- redundant (nr) sequence database with the default parameters. The PSSMs values are scaled to values between 0 and 1 using the following function.

$$f(x) = \frac{1}{1+e^{-x}}$$

Where x is the PSSM's element that stands for the likelihood of the particular residue substitution at that position, normally the feature are obtained from the PSSM in a window has manner as shown in figure

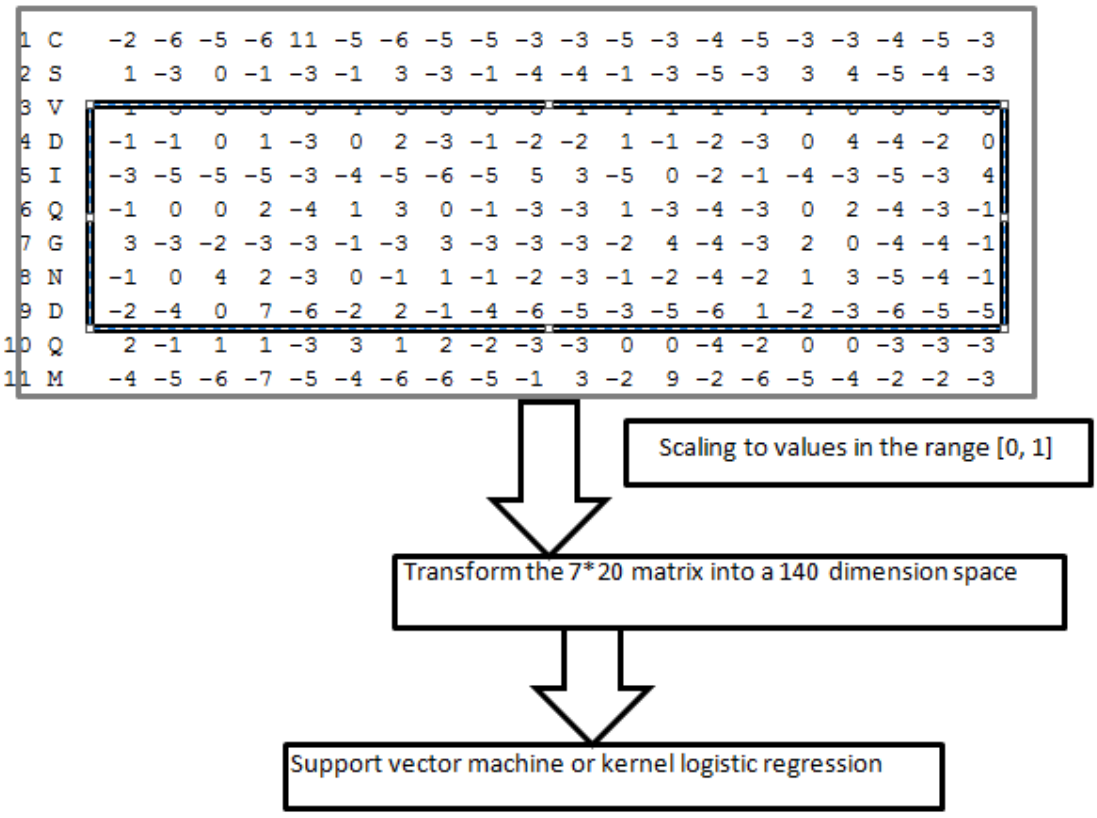

Figure4 : The architecture of the SVM and KLR inputs in form of PSSMs.

Algorithm: PSIBLAST

Input: protein file in FASTA format

Output: PSSMS

Step1: read all line in protein file of FASTA format

Step2: Perform initial alignment with BLAST

Step3: Construct a multiple alignment from hits.

Step4: Prepare a position specific scoring matrix (PSSM).

Step5: Use PSSM profile as the scoring matrix for a second BLAST (run against database).

Step6: Repeat steps 2-4 until convergence.

Step7: stop the process.

3.3 Support Vector Machine (SVM)

The SVM is a state-of-the-art supervised learning model with associated learning algorithm for analyzing and classifying data. It transfers the data from low dimensional space to high or infinite dimensional space and then construct a hyper-plane or hyper-planes in this higher dimensional space to classify the transformed data. Normally the training data are represented as points in a vector space. The hyper-plane with the largest distance to the nearest training data point is considered to be the good separator. Given a training set $\{ x_i , \ \}$ i = 1,….,*l*, where x_i is a vector of features, and $y_i \in \{-1, 1\}$. SVM solves the following primal problem [2].

$$min \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} \xi_{i'}$$

Subject to

$$y_i(w.x_i + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1,....., l,$$

Where *w* is the normal vector to the hyper-plane, b is the offset from the origin, and C is the error penalty parameter. The kernel function, which maps the input space into a higher-dimensional space, can be applied to create SVM classifier for non-linear problem. SVM finds a linear separating hyper plane with the maximal margin in this high dimensional space. Three typical kernel functions can be used for SVM as follows:

Polynomial: $K(x_i, x_j) = (x_i x_j + 1)^d$

Radial basis kernel function $K(x_i, x_j) = \exp(-g\|x_i - x_j\|^2)$

Sigmoid: $K(x_i, x_j) = \tanh(b(x_i, x_j) + r)$

Where d, g, and r are the kernel parameters.

3.3.1 Training and testing

We used LIBSVM package [20] to train and build the SVMs prediction models. The radial basis kernel function was used to transfer the data from a low dimension space to a higher-dimensional space nonlinearly for all the SVMs. The default grid search approach was used to find the optimal values for the LIBSVM's parameters C and gamma. Most of the state-of-the-art secondary structure prediction methods use seven-fold cross validation to assess their prediction performances .Therefore; we used seven-fold cross validation and different window sizes (7, 9,11,13,15 and 17) to assess the performance of our prediction method. We first started by dividing the dataset into subsets that contain equal numbers of proteins. And then use one set to acts the testing set and all other sets are used to acts training set.

3.4 kernel Logistic Regression (KLR)

KLR is the kernel version of logistic regression that allows non-linear probabilistic classification by constructing the logistic regression in higher dimensional feature space using kernel function K: $\chi \times \chi \rightarrow F$. The kernel function evaluates the inner product between the input vectors in the feature space, i.e. $K(x, x') = \phi(x).\phi(x')$, where $x \in \chi \in R^D$. (Mentioned in chapter2)

3.4.1Training and testing

To test the accuracy of secondary structure prediction, different windows sizes (7, 9, 11, 13, 15, and 17) was implemented on all the datasets. That is, these datasets were arbitrarily divided into subsets, each having equaled number of proteins. Each set is an unbalanced set that maintains the naturally occurring amount of (extended Beta-sheet an non-Beta-sheet, α-helix an non- α-helix and coil an non-coil). Ten fold cross validation was performed on all the datasets. Eight of the ten subsets were merged together to form a training set that will be used to train the KLR model. The KLR model is validated for minimum error on the ninth subset to avoid over-training. The last subset is used for testing. This procedure was continual ten times to test. The prediction result for each testing set. The final prediction results are taken as the average of the results from the ten testing sets.

3.5 The proposed approach

The general framework of our proposed approach is shown in Figure 5. In this framework we used local copy of NCBI blast an non redundancy database to convert the protein sequence into their equivalent PSSM. Then we tried different sizes of windows the sizes are (7, 9, 11, 13, 15, and 17) to prepare the feature as ready features for training and testing. We used scaling factor according to the following equation to scale the values of the PSSM to the range of (0-1) to avoid extremes values effect on the training. In the window normally the prediction is made for the central residue. The framework is shown in Figure 6; the SVM classifiers and KLR are constructed using inputs from the clustered model. Then these SVMs classifiers KLR model are integrated with aggregation method called majority voting (mentioned in chapter2).

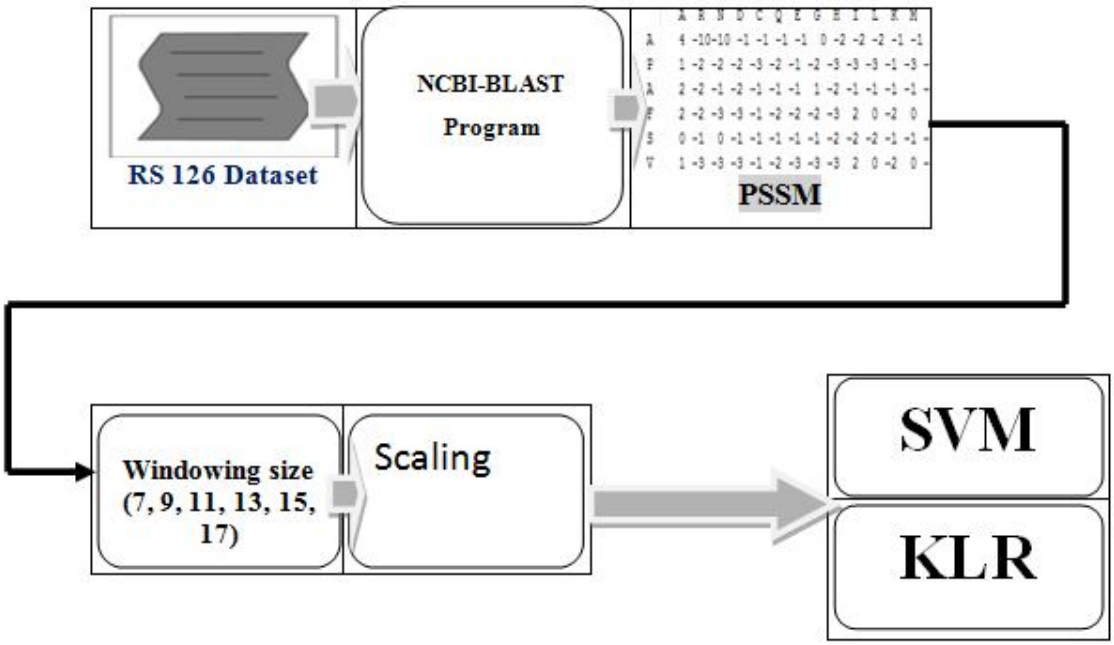

Figure5 : general framework of our proposed approach

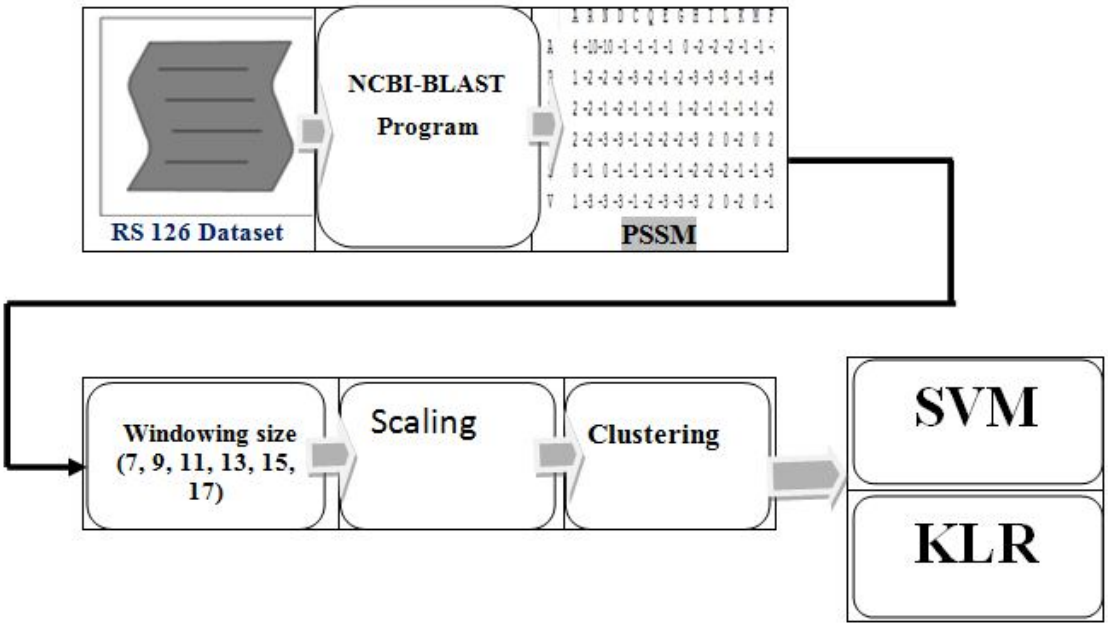

Figure6 : framework of our proposed approach with clustering as preprocessing step

3.6 Clustered model

The ratio of (extended Beta-sheet to non-Beta-sheet 1:3) approximately, (α-helix to non- α-helix and coil to non-coil 1:2) approximately, Thus, the training sets used for secondary structure prediction are imbalanced sets. In our trail experiments, we found that if the non-Beta-sheet set is divided into a three subsets by a suitable clustering algorithm, each non-Beta-sheet subset with the whole extended Beta-sheet set will form approximately stable training set. This stable training set is more likely to be separable in the feature space. That is because the distribution of the negative (non-Beta-sheet, non-coil and non-α-helix) samples in a subset is centralized and compacted. In other words, the Beta-sheet set can be easily separated from each non-Beta-sheet cluster by a different hyper-plane. That means good performance would be expected when constructing localized SVMs and KLR model using each non- Beta-sheet cluster against the Beta-sheet as for the α-helix and coil. But, each of these SVMs and KLR model alone is certainly not a good global classifier. It proposes that it is possible to construct a better classifier than the SVM or KLR trained with the whole data by combining these SVMs and KLR model effectively. Particularly, a localized SVM classifier and KLR model can be constructed for each sub training set, this way the localized SVMs and KLR models will not be affected by the heterogeneity of the whole training set. To outperform the SVM that is trained with the whole data, we need to combine these localized SVMs and KLR models effectively into global one without ignoring their local advantages. Majority voting is one of the methods that are used to combine several classifiers. At the very beginning, the whole negative examples are divided into the number of clusters by a k-means clustering algorithm using original variables. We merged the whole positive examples with each cluster to form number of sub-training sets. These sub-training sets are used to build SVMs and KLR models. The SVMs and KLR models will not be used directly in the prediction, but they will be used as variable generators. During training and prediction stages, these models are unchanged and all the samples enter all of these models, these sub-training sets are used to build KLR model, the result of the SVMs and KLR models are will be aggregated using majority voting vector.

3.7 Performance measures

The quality of prediction is evaluated using four measures, these measure are the prediction accuracy, Qpredicted, Qobserved, and MCC. These measures are the most commonly used measures to evaluate the secondary structure prediction methods. They are calculated using the four values (i) true positive (TP), which is the number of the residues that are correctly classified as(α-helix ,coil or extended B-sheet) ,(ii)true negative(TN), which is the number of the residues that are correctly classified as (non α-helix , non-coil or non-extended Beta-sheet), (iii) false positive (FP), which is the number of residues that have (non α-helix , non-coil or non-extended Beta-sheet) structure and incorrectly classified as having(α-helix ,coil or extended B-sheet) structure, and (iv) false negative (FN), which is the number of residues that have (α-helix ,coil or extended B-sheet) structure and incorrectly classified as having (non α-helix , non-coil or non-extended Beta-sheet) structure. The prediction accuracy (also known as Qtotal) refers to the percentage of correctly classified residues and is calculated as follows:

$$\text{Qtotal}= \frac{TP+TN}{TP+TN+FP+FN}\times 100$$

Qpredicted (also known as the predicted positive value (PPV) or the probability of correct prediction) refers to the percentage of the residues that are correctly predicted as (α-helix, coil or extended B-sheet) among the predicted ones and is calculated as follows [2]:

$$\text{Qpredicted} = \frac{TP}{TP+FP}\times 100$$

Qobserved (also known as sensitivity or coverage) refers to the percentage of the residues that are correctly predicted to have (α-helix, coil or extended B-sheet) structure among those observed as having (α-helix, coil or extended B-sheet) structure. In other words, it represents the fraction of the total positive samples that are correctly predicted and it is calculated as follows: [2]

$$\text{Qobserved} = \frac{TP}{TP+FN}\times 100$$

Matthew's correlation coefficient (MCC) [24] is an important, robust and reliable performance measure. The MCC can be obtained using the following formula:

$$\text{MCC}= \frac{TP * TN - FP * FN}{\sqrt{(TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)}}$$

Normally, the value of MCC is greater than or equal to -1 and less than or equal to 1. If the value of MCC is close to 1 then there is a perfect positive correlation, if it is close to -1 then there is a perfect negative correlation, and a value close to 0 indicates no correlation. The receiver operating characteristic (ROC) curve is implemented in this research as a threshold independent measure. The ROC curve provides the effectiveness of (α-helix, coil or extended B-sheet) prediction method. The area under the ROC curve (AUC) is an important index that reflects the prediction reliability. A good classifier has an area close to1, while a random classifier has an area of 0.5.

CHAPTER FOURE

RESULT AND DISCUSSION

Results and discussion

The methods that are applied on data of the Protein Secondary Structure Prediction used different windows sizes on the PSSMs .These windows size are tested in our proposed approach and results for the RS126 dataset are shown in Table 3, Table4, Table5 for Beta-sheet, coil and α-helix Secondary Structure respectively, From the results we found that the performance of SVM method on coil secondary structure achieved highest prediction accuracy or Qtotal =77.46% when using window size of fifteen, highest MCC=0.5217, Qpredicted =74.298% when using window size of seven and highest Qobserved=72.76 when using window size of seventeen. For beta_ sheet secondary structure achieved highest prediction accuracy or Qtotal=84.86% on window size of seventeen, highest Qpredicted=70.14 %, and highest MCC=0.6374 on window size of seven and highest Qobserved =81.40 when using window size of fifteen. For helix secondary structure we achieved highest prediction accuracy or Qtotal=83.27%, highest MCC=0.64503, and highest Qpredicted=75.13 on window size of fifteen and highest Qobserved =82.01 when using window size of eleven. From these results, it is clear that the performances of the methods are affected by the window sizes. Also we apply the KLR method on the data of the Protein Secondary Structure Prediction by using different window sizes on the PSSMs .These window sizes are tested in our proposed approach and the results for the RS126 dataset are shown in Table 8, Table 10, Table 12 for α-helix, coil, and Beta-sheet, Secondary Structure respectively. To test the effect of the clustering, we applied it on the dataset before creating the SVM and KLR models. The results for the SVM method are shown in Table 6, Table 7 for coil, α-helix respectively, we observed that the clustering has no effect on the result of b-sheet as for the results of KLR method are displayed in Table 9, Table 11, and Table13 for α-helix, coil and beta-sheet respectively, we achieved high prediction accuracy Qtotal of 86.5%, 77.6%, highest MCC of 0.719 and 0.550, and highest Qobserved of 83.2%, and 73.28%, and highest Qpredicted of 83.3%, and 77.9%, on α-helix and coil secondary structure respectively when using the clustering algorithm as preprocessing steps for the SVM method. And we achieved Qtotal of 82.18%, 75.3%and 82.9% , highest MCC of 0.583, 0.483 and 0.508 , highest Qobserved of 84.78 ,79.2 and 85.44, and highest Qpredicted of 98%,82.3 and 96% on α-helix, coil and extended beta-sheet secondary structure respectively when using the clustering algorithm as preprocessing steps for the KLR method. After applying the clustering on dataset, we found that the clustering has

significant effect on the accuracy of Protein Secondary Structure Prediction when using both SVM and KLR methods. In the SVM method the clustering has significant effect on coil and helix secondary structure. For the KLR the clustering has significant effect on coil, helix and extended beta-sheet.

Secondary structure	Size of window	Qobserved	Qpredicted	Qtotal	MCC
Beta-sheet	7	79.83	70.14	84.40	0.6374
	9	20.53	6.76	52.80	-0.1502
	11	74.23	41.85	82.95	0.4677
	13	77.16	47.80	83.86	0.5187
	15	81.40	49.50	79.12	0.5119
	17	76.65	46.75	84.86	0.5175

Table 3 : Result of Prediction for beta-sheet using different slide windows on SVM

Secondary structure	Size of sliding window	Qobserved	Qpredicted	Qtotal	MCC
Coil	7	72.83	74.29	76.39	0.5217
	9	67.82	69.79	74.97	0.4792
	11	69.28	70.28	76.68	0.5089
	13	69.04	70.04	77.28	0.5142
	15	70.51	68.50	77.38	0.5154
	17	72.76	68	75.33	0.4933

Table 4: Result of Prediction for coil using different slide windows on SVM

Secondary Structure	sliding window	Qobserved	Qpredicted	Qtotal	MCC
α- helix	7	75.60	45.03	81.99	0.4845
	9	13.33	9.46	39.91	-0.3345
	11	82.01	69.76	82.58	0.6253
	13	80.82	70.09	82.02	0.6151
	15	80.66	75.13	83.27	0.6450
	17	80.45	73.97	82.15	0.6265

Table 5 : Result of Prediction for α- helix using different slide windows on SVM

Secondary Structure	sliding window	Qobserved	Qpredicted	Qtotal	MCC
coil	7	72.5	75.8	76.6	0.5280
	9	72.4	76.5	76.7	0.5296
	11	72.9	77.3	77.1	0.5398
	13	73.28	77.9	77.6	0.5509
	15	73	77.7	77.6	0.5483
	17	72.5	75.4	76.2	0.5214

Table 6: Result of Prediction for coil using different slide windows with clustering on SVM

Secondary Structure	sliding window	Qobserved	Qpredicted	Qtotal	MCC
α –helix	7	79.3	73.3	84.8	0.6518
	9	80.3	73.8	85.6	0.6661
	11	81.9	74.6	86.1 1	0.6809
	13	82.7	81.5	86.2	0.7085
	15	79.3	72.5	84.99	0.6499
	17	83.2	83.3	86.5	0.7193

Table 7 Result of Prediction for α –helix using different slide windows with clustering on SVM

Secondary Structure	sliding window	Qobserved	Qpredicted	Qtotal	MCC
α- helix	7	82.43	91.87	80.79	0.5257
	9	81.61	90.40	80.47	0.5486
	11	81.90	90.43	80.13	0.5148
	13	84.23	91.57	81.81	0.5220
	15	79.51	92.11	78.46	0.4600
	17	78.75	90.95	77.92	0.4591

Table 8: Result of Prediction for α –helix using different slide windows on KLR

Secondary Structure	sliding window	Qobserved	Qpredicted	Qtotal	MCC
α- helix	7	82.98	98.08	81.69	0.5552
	9	82.39	97.49	81.79	0.5657
	11	82.59	97.73	82.18	0.5837
	13	84.78	95.99	82.03	0.5680
	15	83.30	94.71	81.65	0.4748
	17	82.89	94.66	81.39	0.4724

Table 9 : Result of Prediction for α –helix using different slide windows with clustering on KLR

Secondary Structure	sliding window	Qobserved	Qpredicted	Qtotal	MCC
Coil	7	75.32	80.02	74.62	0.4858
	9	77.02	78.46	74.78	0.4880
	11	77.157	79.45	74.76	0.4829
	13	78.70	76.40	74.23	0.4733
	15	75.35	79.55	73.68	0.4582
	17	76.02	78.23	72.96	0.4400

Table 10: Result of Prediction for coil using different slide windows on KLR

Secondary Structure	sliding window	Qobserved	Qpredicted	Qtotal	MCC
Coil	7	75.67	81.81	75.30	0.4882
	9	79.12	80.21	75.12	0.4826
	11	79.17	82.31	75.71	0.4837
	13	79.09	81.39	75.29	0.4797
	15	76.12	80.72	75.004	0.4754
	17	77.78	79.99	74.63	0.4695

Table 11: Result of Prediction for coil using different slide windows with clustering on KLR

Secondary Structure	sliding window	Qobserved	Qpredicted	Qtotal	MCC
Beta-sheet	7	82.87	94.42	81.20	0.4395
	9	84.86	93.60	81.82	0.3883
	11	84.73	94.30	82.44	0.4398
	13	83.59	94.73	81.59	0.3985
	15	81.99	94.95	80.41	0.4042
	17	82.18	95.34	80.62	0.3789

Table 12: Result of Prediction for Beta-sheet using different slide windows on KLR

Secondary Structure	sliding window	Qobserved	Qpredicted	Qtotal	MCC
Beta-sheet	7	84.58	95.72	82.55	0.4585
	9	85.45	95.99	82.82	0.3977
	11	84.87	94.82	82.91	0.5079
	13	84.62	95.50	82.93	0.4783
	15	84.50	95.72	82.88	0.4613
	17	83.83	95.57	82.31	0.4561

Table 13: Result of Prediction for Beta-sheet using different slide windows with clustering on KLR

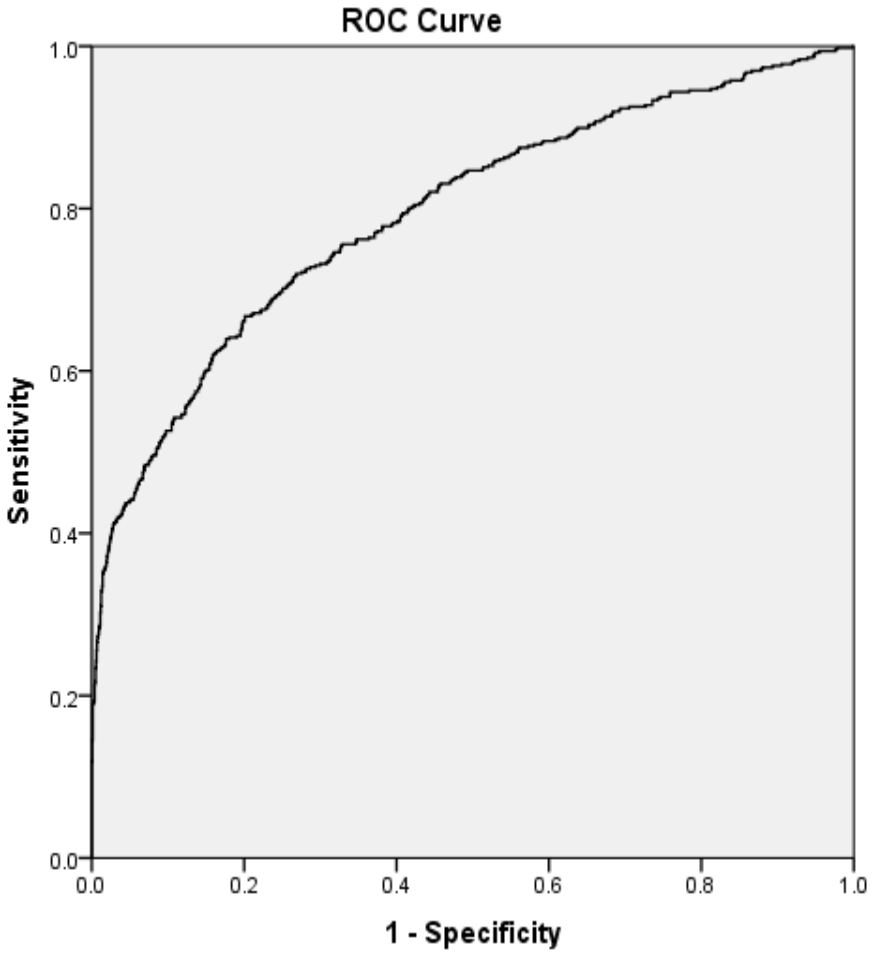

Figure7 : The ROC curve on KLR without clustering

Area Under the Curve

Area
.795

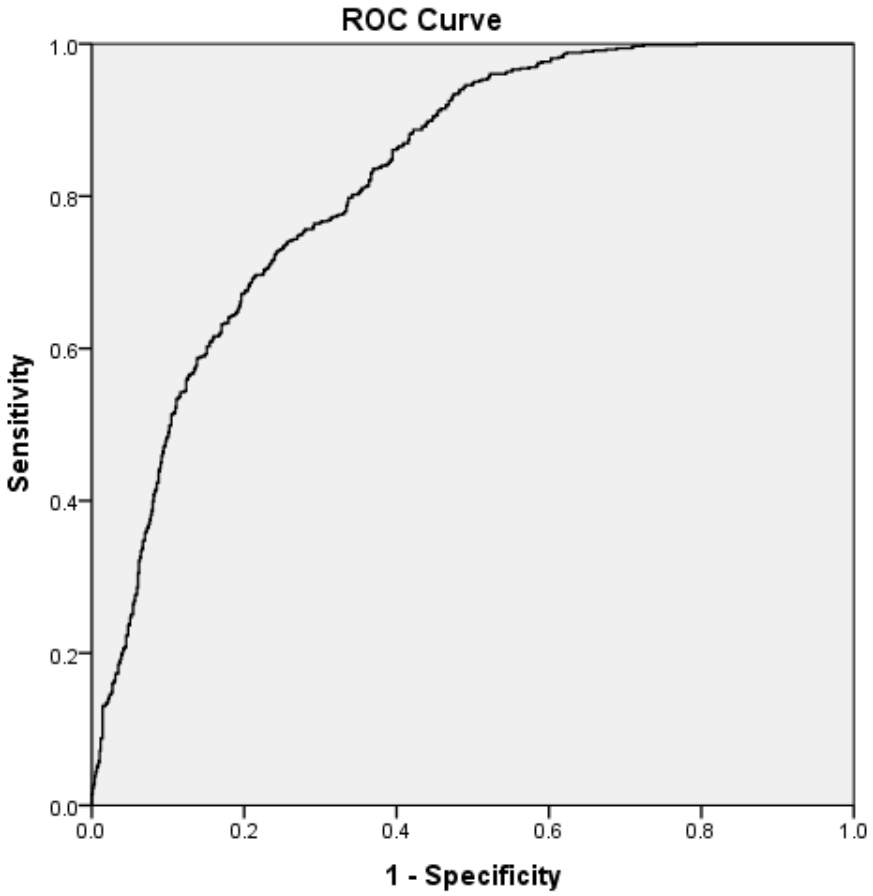

Figure8 : The ROC curve on KLR without clustering.

Area Under the Curve

Area
.835

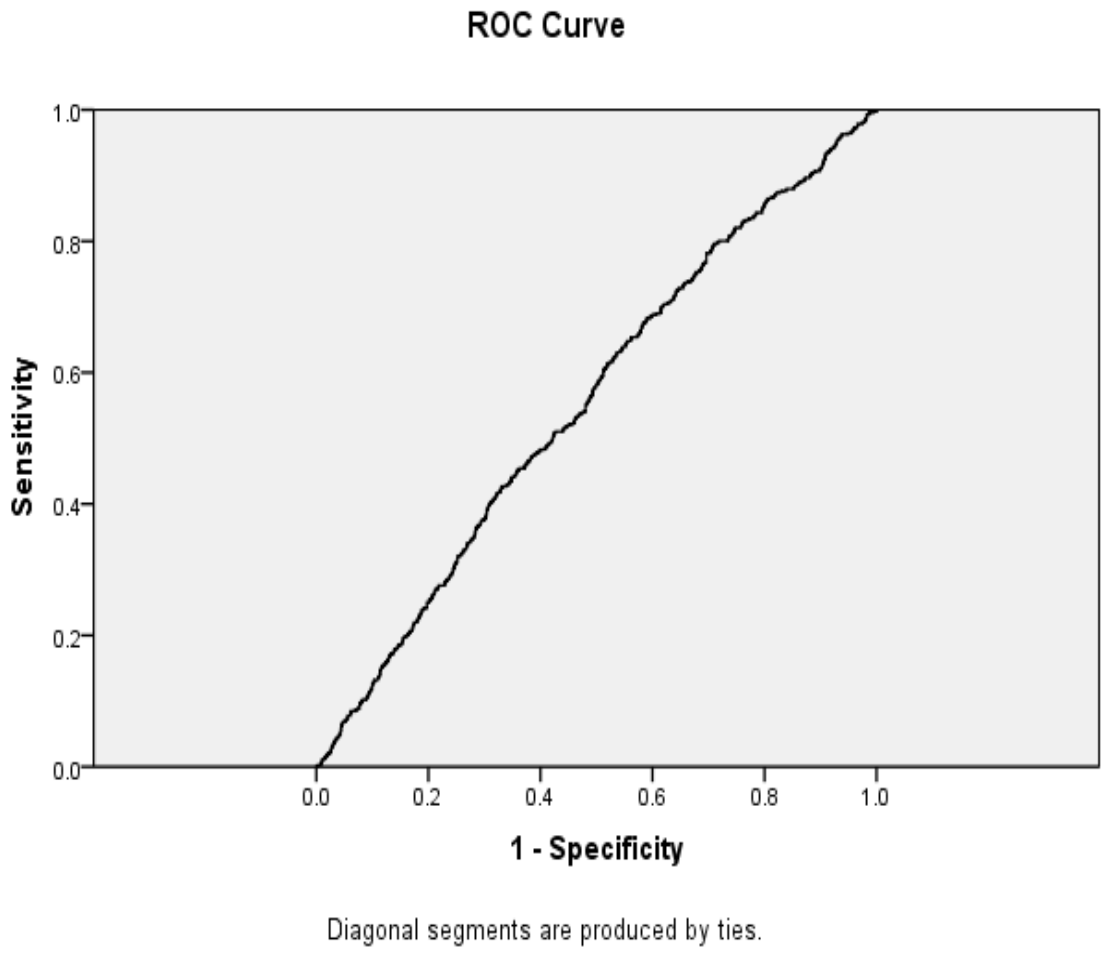

Figure9 : The ROC curve of SVM without clustering

Area Under the Curve

Area
.555

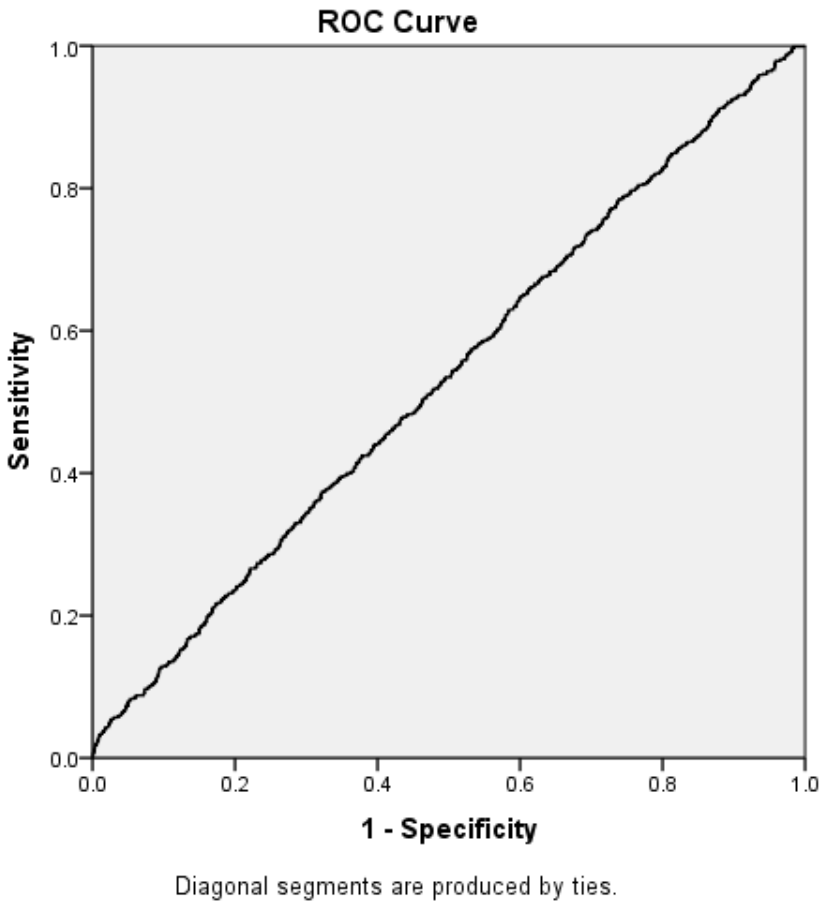

Figure10 : ROC curve of SVM when using the Clustering

Area Under the Curve

Area
.533

.

Conclusions and Recommendation

In this research we proposed an approach that uses the clustering algorithm as preprocessing step for machine learning method to predict Protein secondary structure and compare the result when we use the clustering algorithm with the result without using it in the prediction. We utilized protein profile in the form of PSSMs, we divided the non-Beta-sheet class into three partitions and non(α-helix , coil) classes into two partitions using k-means clustering algorithm and then each partition is combined with the Beta-sheet , α-helix and coil class to form approximately balanced sub-training sets. SVM classifier and KLR models is used for each sub-training set. Using this procedure, the problem of imbalanced class can be overcome, and the SVM computational time can be reduced. The majority voting is used to aggregate the decisions of the SVMs and KLR models. In this study we achieved high prediction accuracy Qtotal of 86.5%, 77.6%, highest MCC of 0.719 and 0.550 , highest Qobserved of 83.2%, and 73.28%, and highest Qpredicted of 83.3%, and 77.9%, on α-helix and coil secondary structure respectively when using the clustering algorithm as preprocessing steps for the SVM method and we achieved Qtotal of 82.18%, 75.3%and 82.9% , highest MCC of 0.583, 0.483 and 0.508 , highest Qobserved of 84.78,79.2 and 85.44, and highest Qpredicted of 98%,82.3 and 96% on α-helix, coil and extended beta-sheet secondary structure respectively when using the clustering algorithm as preprocessing steps for the KLR method. Our recommendation for future work is t to use data reduction methods such as principal component analysis to enhance the performance of learning while keeping the prediction accuracy and other measures as high as possible.