VAWKUM Transactions on Computer Sciences
http://vfast.org/journals/index.php/VTCS@ 2021, ISSN(e): 2308-8168, ISSN(p): 2411-6335
Volume 9, Number 1, January-December , 2021 *pp:47-71*

Analyzing updates in Amino Acid Composition and Translation Algorithm towards Predicting Membrane Proteins using Machine Learning Approaches

Abdulsalam Mohammed Alfarsi[1], Abdulrahman Mohammed Alghanmi [2]

Department of Information Technology, Faculty of Computing and Information Technology in Rabigh, King Abdelaziz University. Jeddah, KSA
*Corresponding author email address: snoke.7rb@gmail.com

ABSTRACT

Membrane proteins are of different types that take on different functions. Classification of protein sequences in a data set is very important for understanding cell functions, disease prevention, and drug discovery. Initially, traditional methods were used for transmembrane protein classification. However, due to advanced technology and new research, it increases the transmembrane protein datasets by thousands which are almost impossible to obtain accurate results based on traditional methods. Computational methods are very useful for membrane protein classification. Several methods such as Pseudo Amino Acid Composition (PseAAC) can extract many silent features of a protein sequence. In this work, we intended to modify an existing algorithm of amino acid composition and translation to extract membrane protein features with better accuracy. To validate our algorithm, we will use the Support Vector Machine SVM and KNN.

KEYWORDS
Membrane protein, feature extraction, SVM, KNN, Amino Acid Composition and Translation

JOURNAL INFO
HISTORY: Received: March 25, 2021 Accepted: June 15, 2021 Published: December 14, 2021

INTRODUCTION

Cell is the Unit of life, group of cells form tissues. Cell membrane is outer layer of a cell, it is the composition of three major molecules: phospholipid, cholesterol and proteins membrane.

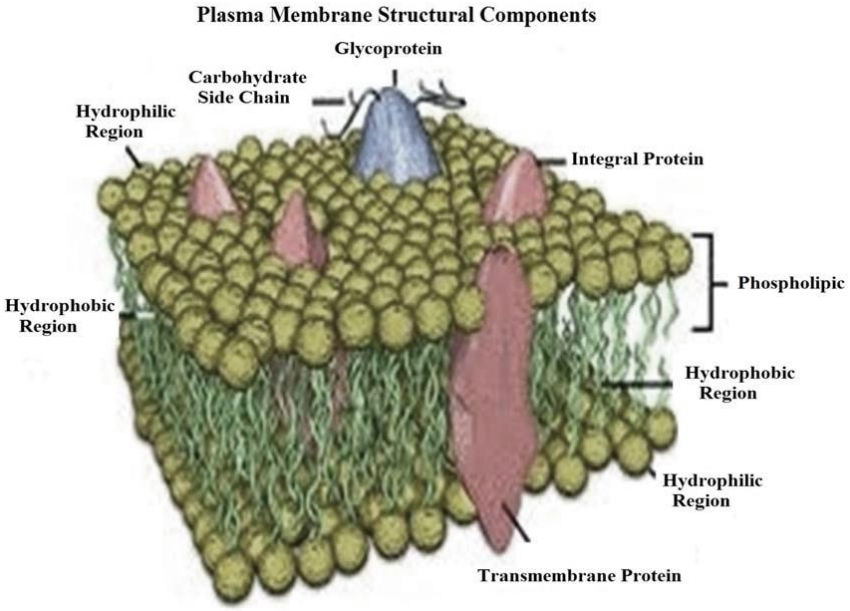

Figure 1. Fluid mosaic model[15]

Protein membrane plays a very important role in cell functions. Proteins are composed of 20 different amino acids. Membrane protein is the type of protein is an essential part of cell which manage the intra and extracellular process of cell. proteins membrane works like receptor to transport molecules in and out of the cell. It can move in all directions in cell membrane. It is documented that membrane protein encode about 20-30% genes of eukaryotic organisms [9][10][11]. Furthermore, human genomes are encoded from approximately 8000 membrane proteins [12]. Moreover, membrane proteins

constitute about 50% of potential targets for the primary target of drugs [12][13][14]. We can categorized membrane protein in integral (intrinsic) and peripheral (extrinsic) membranes, however, some have both qualities as shown in the Figure 1 and 2.

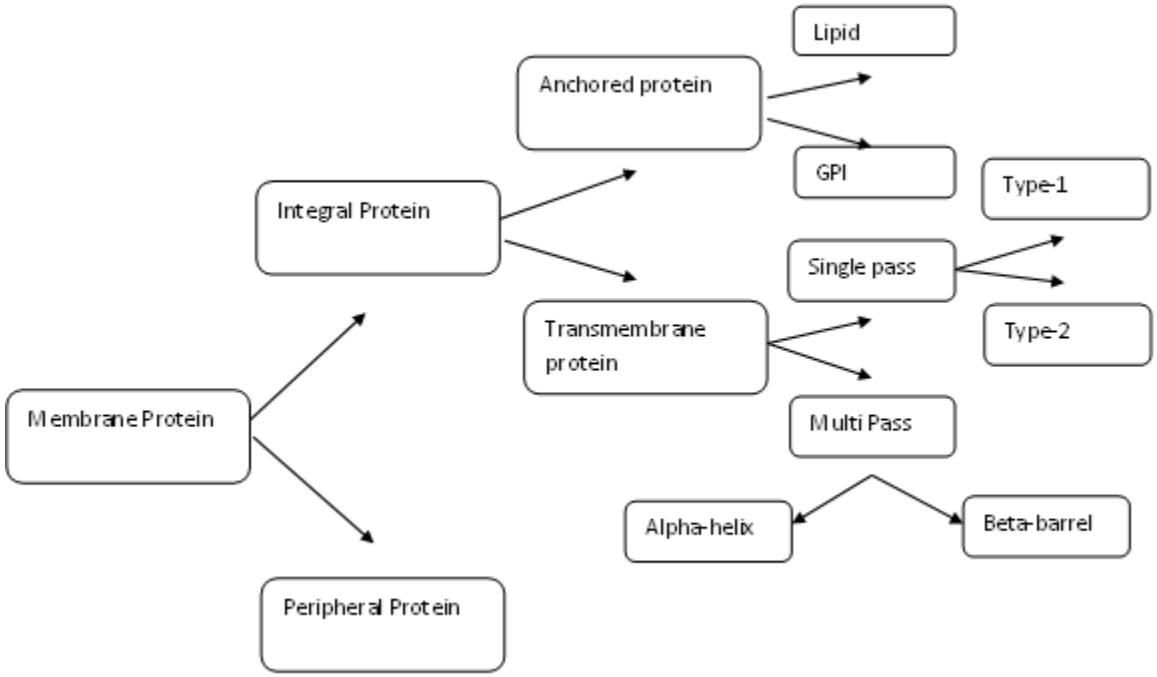

Figure 2. Classification of Membrane Protein

PERIPHERAL MEMBRANE PROTEINS

Peripheral membranes proteins are loosely hang out on both side of a biological membrane and don't cross the membrane as shown in the Figure 3.

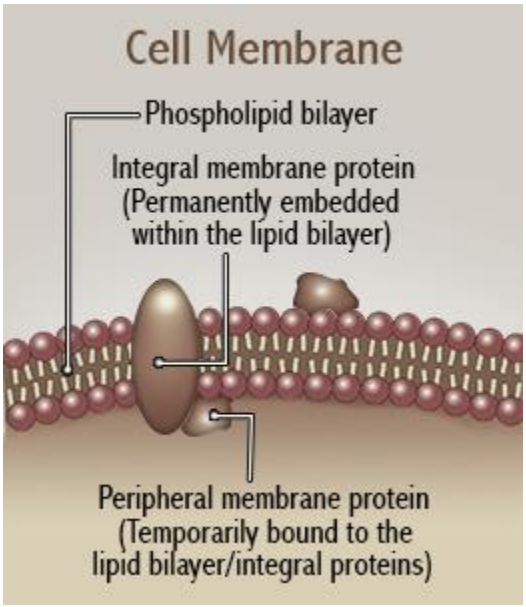

Figure 3. peripheral and Integral protein

INTEGRAL MEMBRANE PROTEINS

The integral membrane is the type of protein attached permanently with biological membrane The integral protein are membrane spanning peptide because of it has both hydrophobic and hydrophilic regions of the polar end and non- polar end as shown in the Figure 3. An integral protein consisting of head group amino NH3 and tail carboxyl group COOH. The integral

protein has classic alpha-helix structure. The portion of amino acid in bilayer are non-polar, while on either side of the membrane are charged. The membrane protein can be classified in transmembrane proteins and anchored membrane proteins.

TRANSMEMBRANE PROTEIN:

It covers the entire surface of biological membrane, contains one or more hydrophobic segment which can differentiate it from non-membrane protein. This situated in three regions of cell i.e., intracellular region (inside the cell region), extracellular region (outside of the cell region) and one part is embedded in the lipid bilayer membrane is non-polar region.

Transmembrane protein has several biological functions like transportation of molecules across membrane, signaling to the cell about the external environment, associated with controlling the exchange of material across the membrane. Nerve cell propagate signals by the flow of ions through membrane protein [3]. Transmembrane has two type of proteins: Alpha-helix single pass and multipass alpha or beta

Transmembrane Alpha Helix single pass is further divided in various types: Type-1 and Type-2 etc., Transmembrane Alpha Helix single pass type-1 is of type in which the amino group is in the extracellular region and carboxyl group is in the intracellular group. Transmembrane Alpha Helix single pass type-2 is opposite to the type-1 as the amino group of the protein is in the intracellular fluid and carboxyl group is in the extracellular region. In both the cases the protein crosses the bilayer only once and contain 20-25 hydrophobic amino acids. Alpha-helix multipass membrane protein passes the lipid bilayer several times as shown in Figures 4 and 5. In case when the protein has even number of alpha-helix then both the terminal amino and coboxel are on the same side of the membrane in other case are in opposite side as shown in the figure 5(a). The Beta-barrel transmembrane proteins usually constitute the outer membranes of Gram- negative bacteria, chloroplasts, and mitochondria [16].

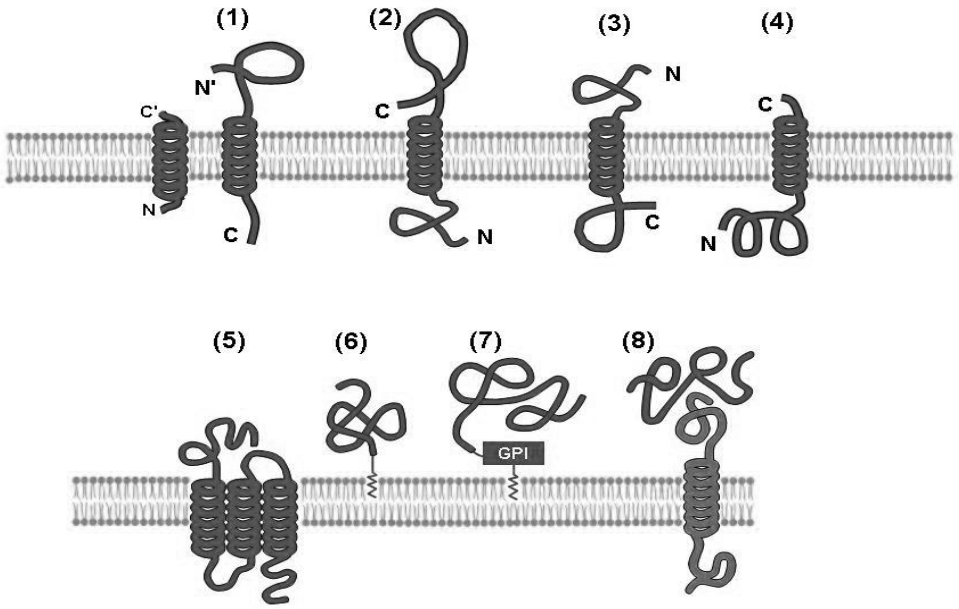

Figure 4. Graphical illustration shows the eight types of membrane proteins

These proteins are constructed from even anti-parallel strands where each strand has hydrogen bonds facing towards its neighbor strand in the primary sequence [17] as shown in the Figure 5(b). The stands around and make a barrel shape, the sidechains on the outside of the barrel are hydrophobic, whereas, the interior of the barrel is hydrophilic. This alternative pattern reveals that one amino acid is non-polar and hydrophobic whereas the second is polar and hydrophilic in topogenic sequence. In addition, approximately 2-3% of genomes are encoded from beta-barrel membrane proteins [18].

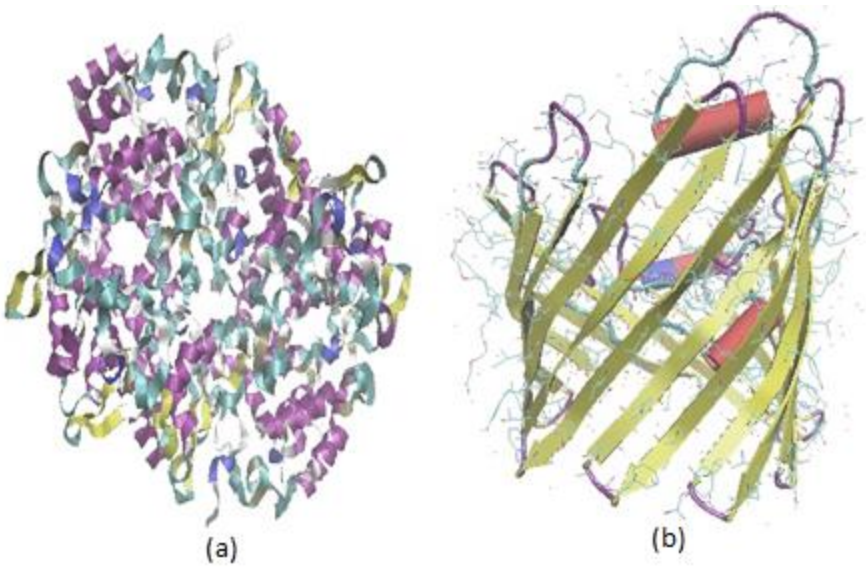

Figure 5. (a) shows the alpha helix and (b) represents beta-barrel transmembrane protein

ANCHORED MEMBRANE PROTEINS:

Anchored membrane proteins have two types: lipid chain-anchored membrane proteins and glycophosphatidylinositol (GPI) anchored membrane proteins are depicted in Figure

- Lipid chain-anchored membrane protein is attached to the bilayer only
- GPI- anchored membrane protein is adhered to the membrane by a GPI-anchor. GPI-anchored membrane proteins have lack of transmembrane region, and have no cytoplasmic tail; therefore, they are located on the outer surface of the plasma membrane.
- Basic Structure of Amino Acid:

The properties of protein can be known by understanding the sequence of amino acids which forms that protein. Amino acids have the same structure only differ by the R group associated with every amino acid as shown in the Figure ---. There are twenty amino acids exists with the Amino group NH2, Carboxyl group COOH, one central atom of carbon, one atom of hydrogen and R-group as shown in the Figure 6.

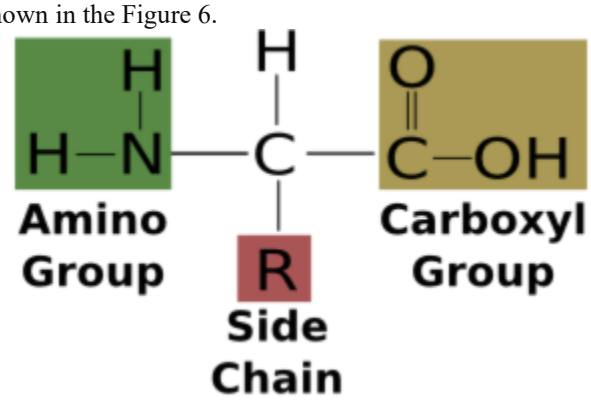

Figure 6. Structure of Amino Acid

R-group is called the side chain or chain residue of amino acid which differentiate one amino acid from another. Each amino acid is represented by unique name and formula. R can be hydrophobic or hydrophilic, charged or uncharged and small and large etc. Two amino acid can connect with onother amino acid by peptide bond. In the reaction the amino group of one amino acid connect with the carboxyl group of another by releasing the water (H2O) to form the protein as shown in the Figure

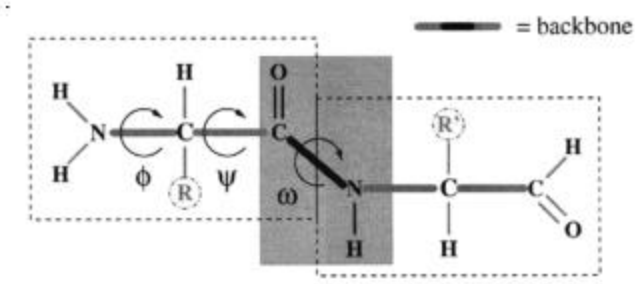

Figure 7. Bond in two Amino Acids

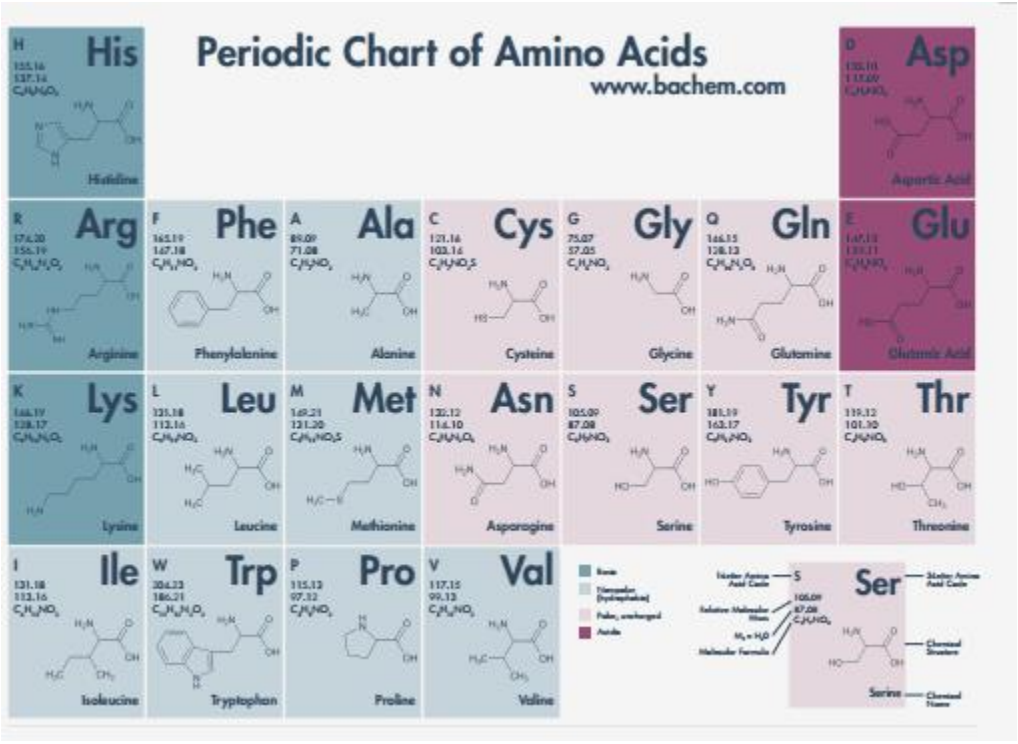

Figure 8. Periodic Chart of Amino Acids

The periodic chart of amino acid is shown in the Figure 8. The dark blue shown in the figure represts the basic amino acid, the red on the right sides shows the acidic amino acid. These two group are the charged amino acids. The light blue group in chart are the non-polar amino acids. The pink group are the polar amino acids. These sets have different properties of amino acids and further its combination predicts the properties of resultant protein like polarity, hydrophobicity and hydrophilicity

- Hydrophobic Amino Acids: The R group of these amino acids contains hydrogen and carbon, these molecules don't interact with water. These amino acids located in the folded structure of protein usually situated in the hydrophobic region of the biological membrane. The list of Hydrophobic amino acids is{A,V,I,L,M,F,Y, W}.
- Polar Amino Acids: These amino acids contains both positive and negative charge with the overall charge is zero. These amino acids are reside in water molecule and mostly found at the surface of folded protein. The polar amino acids are Serine, Threnine, Cysteine, Proline, Asparagine and Glutamine.
- Charge Group Amino Acid: The charge group amino acid can be characterized by the overall charge of the R-group in amino acid. There are two types of charges amino acids the positive and the negative charge. The positive charged amino acids are Lysine, Arginine and Histidine, whereas, the negative charged amino acids are Aspartate and Glutamate.

MOTIVATION:

The development of advanced technology in cell biology increases very rapidly the protein sequences. The identification of sequences in huge data sets is challenging and requires accurate computational methods. For the purpose many computational approaches are developed which are documented using Machin learning approaches.

Bioinformatics is the application of Information technology which consists of three categories: techniques for processing, the application of mathematical and statistical methods to decision making and the simulation of higher order thinking through computer program which reflects to store, organize and analyse and structure the vast amount of biological data [6] [8]. The protein data is available in sequenes in one dimension, in this work it is our intension to make it multi dimensional based on their features and then its classifications for correct prediction of respective Membrane protein. It is also documented that information technology made transformation of the life sciences research [7]. It is now past when IT was considered to help only the organization to achieve its operational goals

Note: One of the still incomplete problems in biology is the protein structure prediction (protein folding) of the native 3D structure of the protein from its sequence. The scientific community consider it one of the most significant and fundamental problem in biological science that has broad economic and scientific impact and whose solution can be advanced only by applying high-performance computing technologies.

METHODOLOGY:

In the first step, we extract the features from the data set based the given features. Amino acids in protein sequenc. In the second step, we classify the trained protein types in helix transmembrane and outer membrane proteins. In cross validation our goal is to apply various classification algorithm on feature extracted dataset to find

- Accuracy of Identified Instances
- Sensitivity
- Specificity
- Precision
- Recall And
- F-Measure

RELATED WORK:

Protein is an important part of the cell membrane which performing various functions. Proteins are the composition of twenty different amino acids form different sequences in a specific patron with many other features. The sequence and the specific patron make it easy for computing approaches to predict various types of membrane protein which is an important problem of the field. The process of prediction can be divided in two steps, firstly to extract the feature of every protein sequence from the given data sets based on various features like hydrophobicity, hydrophilicity, alpha helix and beta-helix etc or some other statistical approaches. In the second phase the featured space is passed to through various methods to identify the required output. Their exist a lot of work to develop and apply an efficient and accurate methods to predict various type of membrane proteins. In 1999, Chou and Elrod [1] proposed an algorithm for protein feature extraction. The model takes amino acid composition (AAC) for feature extraction and for classification uses Covariant Discriminant algorithm (CDA). However, this model have some diffencies including of sequence order information. Further in 2001 Chou extended the existing work developed Pseudo Amino Acid Composition (PseAAC) with more extraction power of protein sequence features [2] . This model uses least Euclidean distance and least hamming distance portlock and CDA to classify membrane protein sequence with their dimension with correlation factors, . In 2004 Wang et al., applied Weighted Support Vector Machine (SVM) with PseAAC to predict the type of membrane protein [3]. The limitation of Chou et al., previous work was that these could identify only the protein sequence however these couldn't differentiate between membrane and non-membrane proteins. To overcome the problem, Chou and Cai used PseAAC and to distinguish the membrane proteins from non-membrane proteins [4]. Similarly they also use of PseAAC and Gene Ontology (GO) to identify membrane protein and their type [5]. With the passage of time some good work came good work is done like split amino acid composition (SAAC) [19], tripeptide composition [20]. For performance evaluation and prediction some techniques are used like support Vector Machine (SVM)e (SVM) [21] [22] [23], k-nearest neighbor (KNN) [22], probabilistic neural network (PNN) [24], random forest (RF) [25]. The main objectives of the above mention work is to find the accuarate method to predict the Membrane protein. We also have the similar objective is to deveolop accurate method and compare the result with the existing.

FRAMEWORK OF PROPOSED MODEL:

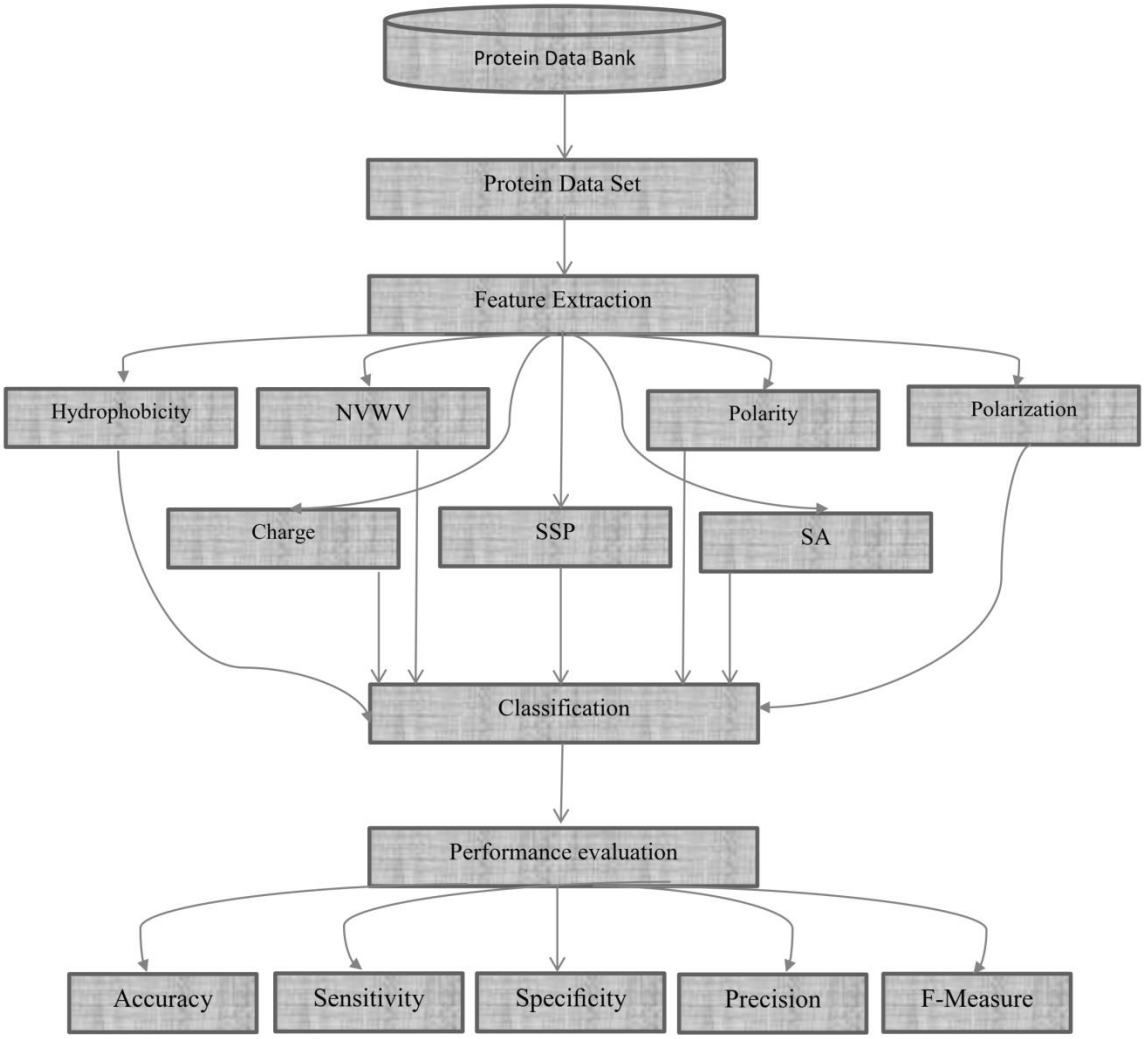

Figure 9. Framework of proposed model

In our proposed model we have the framework consists on the following steps as shown in the Figure 9

1. In order to train the model effectively and enhance the generalization power of classification model, it is mandatory to select a valid benchmark datasets from the protein databank
2. In the second step we will use algorithum to train the data set based on extracted protein features
3. The third step is to classify the train data in various classes
4. Finally performance evaluation is carried out

MEMBRANE PROTEIN DATA SET:

The protein data set is the large collection of digital sequence made from amino acids. Every sequence is the composition of amino acids which are denoted by A,C, D,E,F,G, H, I, K, L, M, N, P, Q,, R, S, T, V,W, Y and as shown in the Figure 10.

FEATURE EXTRAXTION OF PROTEIN SEQUENCE:

It is an important step in machine learning which transforms the row data set into relevant information based on selected sets of features in non-redundant form. The extracted feature seats are also called feature vectors. In protein feature extraction, many techniques are documented like: Amino Acid Composition (AAC), Split Amino Acid Composition (SAAC), Position Specific Storing Matrix (PSSM), Dipeptide Composition (DPC), Tri-peptide compositin (TPC) and Pseudo Amino

Acid Composition (PseAAC). In this work we will extend the exisiting technique AACT and then will compare it results after the classification.

AMINO ACID COMPOSITION AND TRANSLATION

This method characterizes the structure of uncharacterized protein sequence based on biochemical and physicochemical properties. To extract the features of protein sequence seven different physiochemical properties of amino acids are considered which are (1). Hydrophobicity, (2). normalized van der Waals volume, (3) polarity, (4) polarization, (5) charge, (6) secondary structure propensity and (7) solvent accessibility. The above properties are further divided in three sub categories. The Hydrophobicity of amino acid in protein sequence can be classified as polar (R, K, E, D, Q, N), neutral (G, A, S, T, P, H, Y) or hydrophobic (C, V, L, I, M, F, W). The normalized van der Waals volume has three categories based on range-1(G, A, S, C, T, P, D), range-2 (N, V, E, Q, I, L) and range-3 (M, H, K, F, R, Y, W). The polarity is classified as range-1 (L, I, F, W, C, M, V, Y), range-2 (P,A, T, G, S) and range-3 (H, Q, R, k, N, E D). The Polarization is categorized as range-1 (G, A, S, D,T), range-2 (C, P, N, v, Q, I, L) and range-3 (K, M, H, F, R, Y, M). Charge can be classified as positive (K, R), neutral (A,N,C, Q, G, H, I, L, M, F, P, S, T, W, Y, V) and negative (D, E). The secondary structure propensity can be classified as Helix with amino acids (E, A, L, M, Q,K, R, H), strand by (V, I, Y, C, W, F, T) and coil as (G, N, P, S, D) and finally solvent accessibility is categorized as exposed (R,K,Q,E,N,D), buried (L, F, C, G, I, V, W) and intermediate (M, P, S, T, H, Y). We have slight change in the existing algorithm:

ALGORITHM AMINO ACID COMPOSITION AND TRANSLATION WITH MODIFICATION:

```
clc
clear all

feature_train=[ ];
Input: 'Membrane Protein data set
Total_Seq_train=size(Nam_seq_train,2); {Total number of sequences required to be trained}
    for   (i=1 to i=Total_Seq_train)
          seq=seqdata_train(i);
          seq=cell2mat(seq);     {  cell2mat(seq) converts all cell arrays    into one ordinary array }
          feature_train=[feature_train ; Comp_Trans_feature(seq)];
    end
    Membrance_types_Composition_feature=feature_train;
    save Membrance_types_Composition_feature Membrance_types_Composition_feature;
Hydrophobicity
function [ Hyd ] = Hydropobicity(x)
len=size(x,2); Polar=0; Neutral=0; Hydrop=0; Trans=[ ]; PN=0; PH=0; NP=0; NH=0; HP=0; HN=0;    {size(x,2) means length
of the protein sequence}


f or   (i=1 to i=len)
            if(x(i)=='D'   or   x(i)=='E' or   x(i)=='R' or   x(i)=='K' or   x(i)=='Q' or   x(i)=='N')
                  Polar=Polar+1;
                  Trans(i)=1
            elseif(x(i)=='G' or   x(i)=='A' or   x(i)=='S' or   x(i)=='T' or   x(i)=='P' or   x(i)=='H' or   x(i)=='Y')
                   Neutral=Neutral+1;
                  Trans(i)=2
            elseif(x(i)=='C' or   x(i)=='L' or   x(i)=='V' or   x(i)=='I' or   x(i)=='M' or   x(i)=='F' or   x(i)=='W')
                  Hydrop=Hydrop+1
                     Trans(i)=3
        End (if)
    End (f or )
P1=(Polar/len)*100;
N1=(Neutral/len)*100;
H1=(Hydrop/len)*100;
f or   i=1:len-1
 if(transe(i)==1 && transe(i+1)==2)
            PN=PN+1;
elseif(transe(i)==1 && transe(i+1)==3)
PH=PH+1;
elseif(transe(i)==2 && transe(i+1)==1)
        NP=NP+1;
```

```
elseif(transe(i)==2 && transe(i+1)==3)
        NH=NH+1;
elseif(transe(i)==3 && transe(i+1)==1)
        HP=HP+1;
elseif(transe(i)==3 && transe(i+1)==2)
        HN=HN+1;
end
len1=len-1;
PN1=(PN/len1)*100
PH1=(PH/len1)*100;
NP1=(NP/len1)*100;
NH1=(NH/len1)*100;
HP1=(HP/len1)*100;
HN1=(HN/len1)*100;
Hyd=[P1 N1 H1 PN1    PH1 NP1    NH1 HP1 NH1]; (output Matrix)
return;
end
```

Van der Waals Volume:

```
function [ Wan] = Wandarval(x)
len=size(x,2); Range1=0; Range2=0; Range3=0; Trans=[]; PN=0; PH=0; NP=0; NH=0; HP=0; HN=0;
f or ( i=1 to i=len)
          if(x(i)=='G'    or    x(i)=='A' or    x(i)=='S' or    x(i)=='T' or    x(i)=='P' or    x(i)=='D')
                    Range1=Range1+1;
                    Trans(i)=1;
          elseif(x(i)=='N' or    x(i)=='V' or    x(i)=='E' or    x(i)=='Q' or    x(i)=='I' or    x(i)=='L' or    x(i)=='C')
                      Range2=Range2+1;
                    Trans(i)=2;
        elseif(x(i)=='M' or    x(i)=='H' or    x(i)=='K' or    x(i)=='F' or    x(i)=='R' or    x(i)=='Y' or    x(i)=='W')
                  Range3=Range3+1;
                  Trans(i)=3;
          End(if)
End(f or )
R1=Range1/len;
R2=Range2/len;
R3=Range3/len;

f or    i=1:len-1
        if(transe(i)==1 && transe(i+1)==2)
                PN=PN+1;
        elseif(transe(i)==1 && transe(i+1)==3)
                PH=PH+1;
        elseif(transe(i)==2 && transe(i+1)==1)
                NP=NP+1;
        elseif(transe(i)==2 && transe(i+1)==3)
                NH=NH+1;
        elseif(transe(i)==3 && transe(i+1)==1)
                HP=HP+1;
        elseif(transe(i)==3 && transe(i+1)==2)
        HN=HN+1;
        End(if)
End(f or )
len1=len-1;
PN1=(PN/len1)*100
PH1=(PH/len1)*100;
NP1=(NP/len1)*100;
NH1=(NH/len1)*100;
HP1=(HP/len1)*100;
```

```
HN1=(HN/len1)*100;
Wan=[R1 R2 R3   PN1   PH1   NP1   NH1   HP1   HN1]; (Output Vect or )
return;
end
```

Polarity

```
function [ Po] = Polarity(x)
len=size(x,2);
Range1=0; Range2=0; Range3=0; Trans=[ ]; PN=0; PH=0; NP=0; NH=0; HP=0; HN=0;
F or   ( i=1 to i=len)
          if(x(i)=='L'   or   x(i)=='I' or   x(i)=='F' or   x(i)=='W' or   x(i)=='C' or   x(i)=='M' or   x(i)=='V' or
            x(i)=='Y')
                    Range1=Range1+1;
                    Trans(i)=1;
            elseif(x(i)=='P' or   x(i)=='A' or   x(i)=='T' or   x(i)=='G' or   x(i)=='S')
                    Range2=Range2+1;
                    Trans(i)=2;
             elseif(x(i)=='H' or   x(i)=='Q' or   x(i)=='R' or   x(i)=='K' or   x(i)=='N' or   x(i)=='E' or   x(i)=='D')
                     Range3=Range3+1;
                    Trans(i)=3;
             End (if)
P1=Range1/len;
P2=Range2/len;
P3=Range3/len;

f or     (i=1 to len-1)
          if(transe(i)==1 && transe(i+1)==2)
                  PN=PN+1;
          elseif(transe(i)==1 && transe(i+1)==3)
                  PH=PH+1;
          elseif(transe(i)==2 && transe(i+1)==1)
                  NP=NP+1;
          elseif(transe(i)==2 && transe(i+1)==3)
                  NH=NH+1;
          elseif(transe(i)==3 && transe(i+1)==1)
                  HP=HP+1;
          elseif(transe(i)==3 && transe(i+1)==2)
                  HN=HN+1;
          End(if)
End (f or )
len1=len-1;
PN1=(PN/len1)*100
PH1=(PH/len1)*100;
NP1=(NP/len1)*100;
NH1=(NH/len1)*100;
HP1=(HP/len1)*100;
HN1=(HN/len1)*100;

Po=[P1 P2 P3     PN1   PH1   NP1   NH1   HP1   HN1]; (output vector )
return;
end
```

Polarization

```
function [ Pol] = Polarization(x)
len=size(x,2); Range1=0; Range2=0; Range3=0; Trans=[ ]; PN=0; PH=0; NP=0; NH=0; HP=0; HN=0;
f or   (i=1 to i=len)
            if(x(i)=='G'   or   x(i)=='A' or   x(i)=='S' or   x(i)=='D' or   x(i)=='T')
                    Range1=Range1+1;         %Polar
                    Trans(i)=1;
            elseif(x(i)=='C' or   x(i)=='P' or   x(i)=='N' or   x(i)=='V' or   x(i)=='E' or   x(i)=='Q' or   x(i)=='I' or   x(i)=='L')
                    Range2=Range2+1;       %Neutral
                   Trans(i)=2;
```

```
            elseif(x(i)=='K' or  x(i)=='M' or  x(i)=='H' or  x(i)=='F' or  x(i)=='R' or  x(i)=='Y' or  x(i)=='W')
                    Range3=Range3+1;        %Hydropobicity
                  Trans(i)=3;
            end
          P1=Range1/len;
          P2=Range2/len;
          P3=Range3/len;


f or  i=1:len-1
          if(transe(i)==1 && transe(i+1)==2)
                  PN=PN+1;
          elseif(transe(i)==1 && transe(i+1)==3)
                  PH=PH+1;
          elseif(transe(i)==2 && transe(i+1)==1)
                  NP=NP+1;
          elseif(transe(i)==2 && transe(i+1)==3)
                  NH=NH+1;
          elseif(transe(i)==3 && transe(i+1)==1)
                  HP=HP+1;
          elseif(transe(i)==3 && transe(i+1)==2)
                  HN=HN+1;
           End(if)
End (f or )
len1=len-1;
PN1=(PN/len1)*100
PH1=(PH/len1)*100;
NP1=(NP/len1)*100;
NH1=(NH/len1)*100;
HP1=(HP/len1)*100;
HN1=(HN/len1)*100;
Pol=[P1   P2   P3   PN1  PH1  NP1  NH1  HP1  HN1];
return;
end
```

Charge

```
function [ Car] = Charge(x)
len=size(x,2); Positive=0; Neutral=0; Negative=0; Trans=[ ]; PN=0; PH=0; NP=0; NH=0; HP=0; HN=0;
f or  (i=1 to i=len)
            if(x(i)=='K'  or  x(i)=='R')
                    Positive=Positive+1
                  Trans(i)=1;
          elseif(x(i)=='A' or  x(i)=='N' or  x(i)=='C' or  x(i)=='Q' or  x(i)=='G' or  x(i)=='H' or  x(i)=='I'  or
            x(i)=='L' or  x(i)=='M' or  x(i)=='F' or  x(i)=='P' or  x(i)=='S' or  x(i)=='T' or  x(i)=='W' or
            x(i)=='Y' or  x(i)=='V')
                    Neutral=Neutral+1;
                    Trans(i)=2;
             elseif(x(i)=='D' or  x(i)=='E')
                    Negative=Negative+1;
                    Trans(i)=3;
          End(if)
End (f or )

P1=Positive/len;
P2=Neutral/len;
P3=Negative/len;
f or  (i=1 to i=len-1)
          if(transe(i)==1 && transe(i+1)==2)
                  PN=PN+1;
          elseif(transe(i)==1 && transe(i+1)==3)
```

```
                PH=PH+1;
            elseif(transe(i)==2 && transe(i+1)==1)
                NP=NP+1;
            elseif(transe(i)==2 && transe(i+1)==3)
                NH=NH+1;
            elseif(transe(i)==3 && transe(i+1)==1)
                HP=HP+1;
            elseif(transe(i)==3 && transe(i+1)==2)
                HN=HN+1;
            End(if)
End(f or )
len1=len-1;
PN1=(PN/len1)*100
PH1=(PH/len1)*100;
NP1=(NP/len1)*100;
NH1=(NH/len1)*100;
HP1=(HP/len1)*100;
HN1=(HN/len1)*100;
Car=[P1 P2 P3     PN1    PH1    NP1    NH1    HP1    NH1]; output vect or
return;
end
```

Secondary Structure

```
function [ SS] = Secondary_Struc(x)
len=size(x,2); Helix=0; Strand=0; Coil=0; Trans=[ ]; PN=0; PH=0; NP=0; NH=0; HP=0; HN=0;
f or ( i=1 to i=len)
            if(x(i)=='E'    or    x(i)=='A' or    x(i)=='L' or    x(i)=='M' or    x(i)=='Q' or    x(i)=='K' or    x(i)=='R' or    x(i)=='H')
                Helix=Helix+1;          %Helix
                Trans(i)=1;
            elseif(x(i)=='V' or    x(i)=='I' or    x(i)=='Y' or    x(i)=='C' or    x(i)=='W' or    x(i)=='F' or    x(i)=='T')
                Strand=Strand+1;        %Strand
                Trans(i)=2;
            elseif(x(i)=='G' or    x(i)=='N' or    x(i)=='P' or    x(i)=='S' or    x(i)=='D')
                Coil=Coil+1;          %Coil
                Trans(i)=3;
            End(if)
End(f or )
P1=Helix/len;
P2=Strand/len;
P3=Coil/len;
f or    (i=1 to i=len-1)
            if(transe(i)==1 && transe(i+1)==2)
                PN=PN+1;
            elseif(transe(i)==1 && transe(i+1)==3)
                PH=PH+1;
            elseif(transe(i)==2 && transe(i+1)==1)
                NP=NP+1;
            elseif(transe(i)==2 && transe(i+1)==3)
                NH=NH+1;
            elseif(transe(i)==3 && transe(i+1)==1)
                HP=HP+1;
            elseif(transe(i)==3 && transe(i+1)==2)
                HN=HN+1;
            End(if)
End(f or )
len1=len-1;
PN1=(PN/len1)*100
PH1=(PH/len1)*100;
NP1=(NP/len1)*100;
NH1=(NH/len1)*100;
HP1=(HP/len1)*100;
HN1=(HN/len1)*100;
```

```
SS=[P1 P2 P3     PN1    PH1    NP1    NH1    HP1    NH1]; output vect or
return;
end
```

Solventy

```
function [ ST] = Solventy(x)
len=size(x,2); Buried=0; Exposed=0; Intermediate=0; Trans=[ ]; PN=0; PH=0; NP=0; NH=0; HP=0; HN=0;
f or    (i=1 to i=len)
            if(x(i)=='A'    or    x(i)=='L' or    x(i)=='F' or    x(i)=='C' or    x(i)=='G' or    x(i)=='I' or    x(i)=='V' or    x(i)=='W')
                        Buried=Buried+1;          %Polar
                        Trans(i)=1;
              elseif(x(i)=='R' or    x(i)=='K' or    x(i)=='Q' or    x(i)=='E' or    x(i)=='N' or    x(i)=='D')
                        Exposed=Exposed+1;        %Neutral
                        Trans(i)=2;
            elseif(x(i)=='M' or    x(i)=='P' or    x(i)=='S' or    x(i)=='T' or    x(i)=='H' or    x(i)=='Y')
                            Intermediate=Intermediate+1;
                        Trans(i)=3;
            End(if)
End (f or )
P1=Buried/len;
P2=Exposed/len;
P3=Intermediate/len;
f or    i=1:len-1
          if(transe(i)==1 && transe(i+1)==2)
                    PN=PN+1;
          elseif(transe(i)==1 && transe(i+1)==3)
                    PH=PH+1;
          elseif(transe(i)==2 && transe(i+1)==1)
                    NP=NP+1;
          elseif(transe(i)==2 && transe(i+1)==3)
                    NH=NH+1;
          elseif(transe(i)==3 && transe(i+1)==1)
                    HP=HP+1;
          elseif(transe(i)==3 && transe(i+1)==2)
                    HN=HN+1;
          End(if)
End(f or )
len1=len-1;
PN1=(PN/len1)*100
PH1=(PH/len1)*100;
NP1=(NP/len1)*100;
NH1=(NH/len1)*100;
HP1=(HP/len1)*100;
HN1=(HN/len1)*100;
ST=[P1 P2 P3     PN1    PH1    NP1    NH1    HP1    HN1]; output vect or
return;
end
```

EXAMPLE :.

We run the segments of the above algorithm (1). To check the Hydropobicity and translation for the input sequence D,R,G,A,C,L (2). Van der waals volume and translation for the sequence D,N,K,Q,T,R (3). Polarity and translation for the sequence D,N,K,Q,T,R (4). Polarization and translation for the sequence G,C,M,V,T,R (5). Charge and translation for the sequence A,L,Y,Q,E,V (6). Secondary structure for the sequence E,I,P,C,S,Q and (7). Solvent and translation for the sequence A,R,P,C,G,S.

Hydrophobicity:

D,R,G,A,C,L

I	Polar	Neutral	Hydrop	Trans
1	1			Trans (1)=1
2	2			Trans (2)=1
3		1		Trans (3)=2
4		2		Trans (4)=2
5			1	Trans (5)=3
6			2	Trans (6)=3

P1=Polar/len1 = 2/6= 0.33
N1=Neutral/len 1= 2/6= 0.33
H1=Hydrop/len1 = 2/6= 0.33

I	PN	PH	NP	NH	HP	HN	Trans(i)	Trans(i+1)
1	0	0	0	0	0	0	1	1
2	1	0	0	0	0	0	1	2
3	0	0	0	0	0	0	2	2
4	0	0	0	1	0	0	2	3
5	0	0	0	0	0	0	3	3

PN1=(PN/lem1)*100=(1/5)*100=20%
PH1=(PH/len1)*100=(0/5)*100=0%
NP1=(NP/len1)*100=(0/5)*100=0%
NH1=(NH/len1)*100=(1/5)*100=20%
HP1=(HP/len1)*100=(0/5)*100=0%
HN1=(HN/len1)*100=(0/5)*100=0%
Wan=[0.33 0.33 0.33 20% 0% 0% 20% 0% 0%]
Van der waals volume:
D,N,K,Q,T,R

	I	Range1	Range2	Range3	Trans
	1	1			Trans (1)=1
	2		1		Trans (2)=2
	3			1	Trans (3)=3
	4		2		Trans (4)=2
	5	2			Trans (5)=1
	6			2	Trans (6)=3

R1=Range1/len1 = 2/6= 0.33
R2=Range2/len 1= 2/6= 0.33
R3=Range3/len1 = 2/6= 0.33

I	PN	PH	NP	NH	HP	HN	Trans(i)	Trans(i+1)
1	1	0	0	0	0	0	1	2
2	0	0	0	1	0	0	2	3
3	0	0	0	0	0	1	3	2
4	0	0	1	0	0	0	2	1
5	0	1	0	0	0	0	1	3

PN1=(PN/lem1)*100=(1/5)*100=20%
PH1=(PH/len1)*100=(1/5)*100=20%
NP1=(NP/len1)*100=(1/5)*100=20%

NH1=(NH/len1)*100=(1/5)*100=20%
HP1=(HP/len1)*100=(0/5)*100=0%
HN1=(HN/len1)*100=(1/5)*100=20%
Wan=[0.33 0.33 0.33 20% 20% 20% 20% 0% 20%]
Polarity : D,N,K,Q,T,R

I	Range1	Range2	Range3	Trans
1			1	Trans (1)=3
2			2	Trans (2)=3
3			3	Trans (3)=3
4			4	Trans (4)=3
5		1		Trans (5)=2
6			5	Trans (6)=3

P1=Range1/len 1= 0/6 = 0
P2=Range2/len1 = 1/6 = 0.16
P3Range3/len 1= 5/6 = 0.83
PN1=(PN/lem1)*100=(0/5)*100=0%

I	PN	PH	NP	NH	HP	HN	Trans(i)	Trans(i+1)
1	0	0	0	0	0	0	3	3
2	0	0	0	0	0	0	3	3
3	0	0	0	0	0	0	3	3
4	0	0	0	0	0	1	3	2
5	0	0	0	1	0	0	2	3

PH1=(PH/len1)*100=(0/5)*100=0%
NP1=(NP/len1)*100=(0/5)*100=0%
NH1=(NH/len1)*100=(1/5)*100=20%
HP1=(HP/len1)*100=(0/5)*100=0%
HN1=(HN/len1)*100=(1/5)*100=20%
Po=[0 0.16 0.83 0% 0% 0% 20% 0% 20%]
Polarization :
G,C,M,V,T,R

I	Range1	Range2	Range3	Trans
1	1			Trans (1)=1
2		1		Trans (2)=2
3			1	Trans (3)=3
4		2		Trans (4)=2
5	2			Trans (5)=1
6			2	Trans (6)=3

P1=Range1/len1 = 2/6 = 0.33
P2=Range2/len 1= 2/6 = 0.33
P3=Range3/len 1= 2/6 = 0.33

I	PN	PH	NP	NH	HP	HN	Trans(i)	Trans(i+1)
1	1	0	0	0	0	0	1	2
2	0	0	0	1	0	0	2	3

3	0	0	0	0	0	1	3	2
4	0	0	0	0	1	0	2	1
5	0	1	0	0	0	0	1	3

PN1=(PN/lem1)*100=(1/5)*100=20%
PH1=(PH/len1)*100=(1/5)*100=20%
NP1=(NP/len1)*100=(0/5)*100=0%
NH1=(NH/len1)*100=(1/5)*100=20%
HP1=(HP/len1)*100=(1/5)/*100=20%
HN1=(HN/len1)*100=(1/5)*100=20%
Pol=[0.33 0.33 0.33 20% 20% 0% 20% 20% 20%]
Charge:
A,L,Y,Q,E,V

I	Positive	Neutral	Negtive	Trans
1		1		Trans (1)=2
2		2		Trans (2)=2
3		3		Trans (3)=2
4		4		Trans (4)=2
5			1	Trans (5)=3
6		5		Trans (6)=2

P1=Positive/len 1= 0/6 = 0
P2=Neutral/len 1= 5/6 = 0.83
P3=Negative/len 1= 1/6 = 0.16

I	PN	PH	NP	NH	HP	HN	Trans(i)	Trans(i+1)
1	0	0	0	0	0	0	2	2
2	0	0	0	0	0	0	2	2
3	0	0	0	0	0	0	2	2
4	0	0	0	1	0	0	2	3
5	0	0	0	0	0	1	3	2

PN1=(PN/lem1)*100=(0/5)*100%=0%
PH1=(PH/len1)*100=(0/5)*100%=0%
NP1=(NP/len1)*100=(0/5)*100%=0%
NH1=(NH/len1)*100=(1/5)*100%=20%
HP1=(HP/len1)*100=(0/5)*100%=0%
HN1=(HN/len1)*100=(1/5)*100%=20%
Car=(0 0.83 0.16 0% 0% 0% 20% 0% 20%
Secondary structure:
E,I,P,C,S,Q

I	Helixe	Strand	Coli	Trans
1	1			Trans (1)=1
2		1		Trans (2)=2
3			1	Trans (3)=3
4		2		Trans (4)=2
5			2	Trans (5)=3
6	2			Trans (6)=1

P1=Helixe/len 1= 2/6 = 0.33
P2=Strnad/len1 = 2/6 = 0.33
P3=Coli/len 1= 2/6 = 0.33

PN1=(PN/lem1)*100=(1/5)*100%=20%

I	PN	PH	NP	NH	HP	HN	Trans(i)	Trans(i+1)
1	1	0	0	0	0	0	1	2
2	0	0	0	1	0	0	2	3
3	0	0	0	0	0	1	3	2
4	0	0	0	2	0	0	2	3
5	0	0	0	0	1	0	3	1

PH1=(PH/len1)*100=(0/5)*100%=0%
NP1=(NP/len1)*100=(0/5)*100%=0%
NH1=(NH/len1)*100=(2/5)*100%=40%
HP1=(HP/len1)*100=(1/5)*100%=20%
HN1=(HN/len1)*100=(1/5)*100%=20%
SS=[0.33 0.33 0.33 20% 0% 0% 40% 20% 20%]
Solvent:
A,R,P,C,G,S

I	Burid	Exposed	Intermediate	Trans
1	1			Trans (1)=1
2		1		Trans (2)=2
3			1	Trans (3)=3
4	2			Trans (4)=1
5	3			Trans (5)=1
6			2	Trans (6)=3

P1=Burid/len1 = 3/6 = 0.5
P2=Exposed/len1 = 1/6 = 0.16
P3=Intremediate/len1 = 2/6 = 0.33

I	PN	PH	NP	NH	HP	HN	Trans(i)	Trans(i+1)
1	1	0	0	0	0	0	1	2
2	0	0	0	1	0	0	2	3
3	0	0	0	0	1	0	3	1
4	0	0	0	0	0	0	1	1
5	0	1	0	0	0	0	1	3

PN1=(PN/lem1)*100=(1/5)*100%=20%
PH1=(PH/len1)*100=(1/5)*100%=20
NP1=(NP/len1)*100=(0/5)*100%=0%
NH1=(NH/len1)*100=(1/5)*100%=20%
HP1=(HP/len1)*100=(1/5)*100%=20%
HN1=(HN/len1)*100=(0/5)*100%=0%
ST[0.5 0.16 0.33 20% 20% 0% 20% 20% 0%]

CLASSIFICATION:

Classification is the method, it makes sub-classes of a dataset based on its know divided categories as observed by its training. Classifiaction algorithm are very important in bioinformatics which divided the dataset in various classes based on its observed different charecteristics. The methods used for classifications are classifier, in protein research we can use many classifiers as Support Vector Machine (SVM), K-nearest neighbor (KNN) and Random Forest (RF) etc. In this work we restricted ourselves to SVM to classify the trained dataset based on our algorithm.

Support Vector Machine (SVM): It is a classifier divided the trained dataset based on the extracted features. It draw a line between different sub-classes called the hyperplane. We give the mechanisum of SVM over the dataset with two feature class as mentioned below:

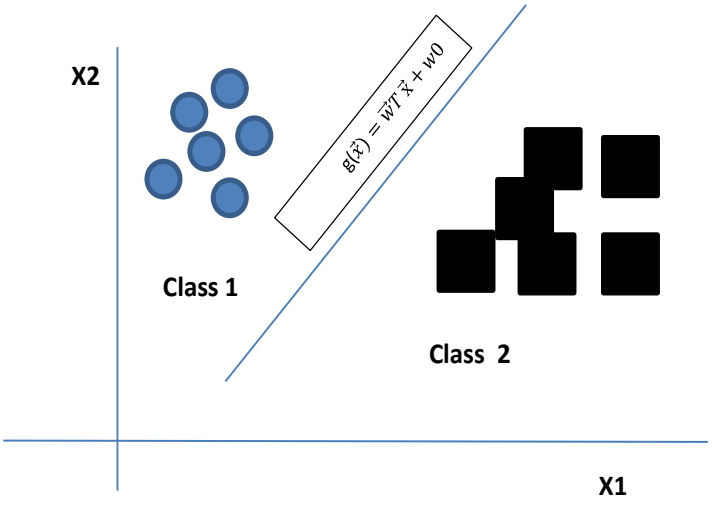

Figure 10. Support Vector Machine Mechanisum

The goal of the SVM algorithm is develop a hyperplane which make partition of the trained vectors in two classes. The hyperplane is the best choice plane which leaves maximum margin from both the classes class-1 and class 2. Hence the hyperplane is represented by the equation:

$g(\vec{x}) = \underset{w}{\rightarrow}^{t} \underset{x}{\rightarrow} + w^0$, Where $\underset{w}{\rightarrow}$ is the weight vector, further the equation must satisfy the following two conditions:

$g(\vec{x}) \geq 1 \quad \forall x \in \text{class } 1$

$g(\vec{x}) \leq -1 \quad \forall x \in \text{class } 2$

This means that the distance of the hyperplane must be greater than or equal to 1 from both the classes. Hence the minimum margins between the two classes is equal to 2 that is the margin is greater than or equal to 2. The total margin between two classes can be computed by

$$\frac{1}{\left\|\underset{w}{\rightarrow}\right\|} + \frac{1}{\left\|\underset{w}{\rightarrow}\right\|} = \frac{2}{\left\|\underset{w}{\rightarrow}\right\|}$$

The maximum of w will make minimum the distance between the two classes. Mininmizing the value of w is a non-linear optimization task solved by Karush-Kuhn-Tucker (KKT) conditions using lagrange multiplier λi

$$w = \sum_{i=1}^{n} \lambda_i y_i x_i$$

$$\sum_{i=1}^{n} \lambda_i y_i = 0$$

PERFORMANCE EVALUATION:

Performance evaluation is the test to check the validity of unseen data classified by the classifier. To asses the performance of the classifier some evaluation criterias needed to be defined.

- accuracy
- sensitivity
- specificity
- precision
- recall and
- F-measure

To evaluate the above we consider classification positively or Negative prediction, implies, the following classification outcomes .

- True Negative (TN) (actually negative and predicted negative)
- True Positive (TP) (actually positive and predicted positive)
- False Negative (FN)(actually positive and predicted negative)

- False Positive (FP) (actually negative and predicted positive)

The outcomes can be formatted by a matrix

Actual class	Predicted class	
	Negative	Positive
Negative	TN	FP
Positive	FN	TP

ACCURACY:

It is the percentage of accurate classification of the data set. It is the total number true positive and true negative out of the total number of cases.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

SENSITIVITY:

This is the total number of positive correct classification. It is defined as

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

PRECISION:

It's the the proportion of correct positive classification (True positive) from cases that are predicted positive.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FN}}$$

SPECIFICITY:

It's the the proportion of negative classification out of actual negative. It is defined as:

$$\text{Specificity} = \frac{\text{TN}}{\text{FP} + \text{TN}}$$

MATHEWS CORRELATION COEFFICIENT (MCC):

MCC is introduced by Brain W. Mathews, is a correlation coefficient use to predict and observe the classification test. It returns values from -1 to 1. If the value of the coefficient is 1, it represents the correct prediction and if 0, it indicates random prediction and if -1 then the prediction is not correct. The MCC is defined mathematically as:

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

THE F-SCORE:

The F-score (or F-measure is a single measure of a classification procedure's usefulness. The F-score considers both the Precision and the Recall of the procedure to compute the Score. The higher the F-score, the better the predictive power of the classification procedure. A score of 1 means the classification procedure is perfect. The lowest possible F-score is 0.

$$0 \leq \text{F} \leq 1$$

The F-score is the harmonic mean of Precision and Recall.

$$\text{F} = \frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}$$

Alternatively

$$\text{F} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Whereas the Harmonic Mean is a kinds of average can be expressed as the reciprocal of the average of the reciprocals.

SYSTEM IMPLEMENTATION

In this section we will make implementation of the updated algorithm of feature extraction of protein sequence and its result comparison with the existing algorithms of feature extraction based on its classification accuracy. For the purpose we take three datasets S1, S2 and S3 as available in the appendix A. B and C. S1 is the dataset consisting on 274 sequences with four classes: single pass=32 sequences, multi-pass=192, peripheral=30 and lipid-anchor=20 sequences. The dataset S2 contains two classes one class contains 274 sequences of Membrane protein and another contains 290 sequences of nonMembrane sequences. The data set S3 has total sequences 4874 which are divided in two classes: ESM Membrane protein sequnces which are 410 and 4464 non-ECM proteins sequences. For the prediction we will use Matlab 13.3 for feature extraction and Weka for their classifications and performance evaluation..

PREDICTION PERFORMANCE OF TRAINED DATASET S1 ON EXISTING ALGORITHM

The prediction results of classifier on dataset S1, of Mycobacterium tuberculosis, are listed in Table 1. In this work, SWM, KNN and RF are used on the trained dataset produced by Amino Acid composition and translation (AACT) algorithm. In result the RF has the highest result of 78.46 % whereas the SVM has 71.89 % and KNN has 76.27% as shown in the Table 1.

Table 1.Classification of trained dataset S1 based on existing algorithm

Method	Accuracy	Sensitivity	Specificity	MCC	F-Measure
SVM	71.89%	0.7189	0.9076	0.6266	0.7198
KNN	76.27%	0.7627	0.9211	0.6838	0.7627
RF	78.46%	0.7846	0.9282	0.7128	0.7846

PREDICTION PERFORMANCE OF TRAINED DATASET S1 ON EXISTING ALGORITHM WITH SMOTH FILTER

Since, the dataset has four classes with non-unifrm number of sequences in the classes. Therfore smote filter is used to make the data uniform using Weka smoth filter on the trained dataset. Further, SWM, KNN and RF are applied for classification and performance measurement. The results are increased by 10% in all as 80.54% by SVM. 86.21% by KNN and 84.86 % by RF which is shown in the Table 2.

Table 2. Classification of trained dataset S1 based on existing algorithm with smoth filter

Method	Accuracy	Sensitivity	Specificity	MCC	F-Measure
SVM	80.54%	0.8054	0.9345	0.7399	0.8054
KNN	86.21%	0.8621	0.9548	0.8170	0.8621
RF	84.86%	0.8486	0.9501	0.7987	0.8486

PREDICTION PERFORMANCE OF TRAINED DATASET S1 ON PROPOSED ALGORITHM

Now we apply the classifiers on trained datset by our proposed algorithm as shown on page 27. The algorithm is applied on the dataset-1 which has 274 sequences and 4 classes as mentioned above. The classifiers SVM, KNN and RF are applied and obtained the result as 69.7%, 74.08% and 77.37% respectively as performance evaluation as shown in the Table 3. In comparison of results we obtained 2% less in SVM and KNN and 1% less in RF as we compared to the result of existing algorithm of Amino Acid composition and translation as shown in the Table 1.

Table 3. . Classification of trained dataset S1 based on proposed algorithm

Method	Accuracy	Sensitivity	Specificity	MCC	F-Measure
SVM	69.708%	0.6970	0.8990	0.5961	0.6970
KNN	74.0876%	0.7490	0.9169	0.6660	0.7490
RF	77.3723%	0.7737	0.9242	0.6979	0.7737

PREDICTION PERFORMANCE OF TRAINED DATASET S1 ON PROPOSED ALGORITHM WITH SMOTH FILTER

Similarly to compare our algorithm with the addition of smote filter, we got 80.54%, 86.21% and 84.86% as shown in the Table 4 . The algorithm worked but the result is less than by 2% approximately as compared to the existing algorithm with smoth filter in all cases of Table 2.

Table 4. Classification of trained dataset S1 based on proposed algorithm with smoth filter

Method	Accuracy	Sensitivity	Specificity	MCC	F-Measure
SVM	74.5098%	0.7450	0.9150	0.6609	0.7450
KNN	82.6797%	0.8267	0.9424	0.7692	0.8267
RF	80.0654%	0.8006	0.9344	0.7350	0.8006

PREDICTION PERFORMANCE OF TRAINED DATASET S2 ON EXISTING ALGORITHM

The prediction results of classifier on dataset S2, with total sequences 564 which having two classes Membrane protein sequnces 274 and 290 are nonMembrane sequences, is listed in Table 5. In this work, SWM, KNN and RF used

on the trained dataset produced by the algorithm Amino Acid composition and translation (AACT) with resut 92.56 %, 90.79% and 91.85% respectively. In the result the SVM has the highest result as compared to KNN and RF.

Table 5. Classification of trained dataset S2 based on existing algorithm

Method	Accuracy	Sensitivity	Specificity	MCC	F-Measure
SVM	92.5664%	0.4894	0.4285	-0.0429	0.4894
KNN	90.7965%	0.4678	0.6538	0.0706	0.4678
RF	91.8584%	0.4816	0.5217		0.4816

PREDICTION PERFORMANCE OF TRAINED DATASET S2 ON EXISTING ALGORITHM WITH SMOTH FILTER

Since, the dataset has two classes with non-unifrm number of sequences. Therfore smote filter is used to make the data uniform using Weka smoth function on the trained dataset. Further, SWM, KNN and RF are applied for classification and performance measurement. The result of SVM is increased by approximately 2%, KNN increased by 5% and RF is increased by 3%. Based on this combination of algoritum KNN has highest accuracy as compared to SVM and RF as shown in the Table 6.

Table 6. Classification of trained dataset S2 based on existing algorithm with smoth filter

Method	Accuracy	Sensitivity	Specificity	MCC	F-Measure
SVM	94.8749%	0.6771	0.2094	-0.0538	0.6771
KNN	95.8284%	0.6716	0.2285	-0.0426	0.6716
RF	94.8749%	0.6708	0.3255		0.6708

PREDICTION PERFORMANCE OF TRAINED DATASET S2 BASED ON PROPOSED ALGORITHM

Now we apply the classifiers on trained datset by our proposed algorithm as shown in the Figure ----, page -----. The algorithm is applied on the dataset-2 which has 564 sequences in 2 classes as mentioned above. The classifiers SVM, KNN and RF are applied and obtained 92.92%, 91.68% and 92.03% respectively as performance evaluation as shown in the Table 7. In comparison of results we obtained 0.4% greater result in SVM, 1% in KNN and 2.6% greater in RF as we compared to the result of existing algorithm of Amino Acid composition and translation as shown in the Table 5. The overalall maximum value is achieved by our updated algorithm with **SVM which is 92.9206.**

Table 7. Classification of trained dataset S2 based on proposed algorithm

Method	Accuracy	Sensitivity	Specificity	MCC	F-Measure
SVM	92.9204%	0.4971	0.325	-0.0912	0.4971
KNN	91.6841%	0.4710	0.6382	0.0605	0.4710
RF	92.0354%	0.4826	0.5111	0.841	0.4826

PREDICTION PERFORMANCE OF TRAINED DATASET S2 ON PROPOSED ALGORITHM WITH SMOTH FILTER

Similarly for the comparison purpose we apply the smoth filter to uniform the dataset trained by our updated algorithm. Further, SWM, KNN and RF are applied for classification and performance measurement and got the 95.59%, 96.543% and 94.39% accuracy for SVM, KNN and RF respectively as shown in the Table 8.. The result of SVM is increased by 1%, KNN increased by 0.7% and RF is decreased by 0.4% as compared to the existing algorithm with smoth filter as shown in the Table 6. **The overalall maximum value is achieved by our updated algorithm with KNN which is 96.54%.**

Table 8. Classification of trained dataset S2 based on proposed algorithm with smoth filter

Method	Accuracy	Sensitivity	Specificity	MCC	F-Measure
SVM	95.59%	0.6758	0.1621	-0.0714	0.6758
KNN	96.5435%	0.6716	0.1935	-0.0744	0.6716
RF	94.3981%	0.6679	0.4042	0.0351	0.6679

PREDICTION PERFORMANCE OF TRAINED DATASET S3 BASED ON EXISTING ALGORITHM

To check the performance of updated algorithm we applied the algorithm on third dataset S3 which has total sequences 4874 divided in two classes: ESM Membrane protein sequnces are 410 and non-ECM proteins sequences are 4464. The prediction results of classifier on dataset S3, is listed in Table 9. In this work, SWM, KNN and RF are used on the trained dataset by Amino Acid composition and translation (AACT) with resut 93.35 %, 91.03% and 93.08% respectively. In the result the SVM has the highest value as compared to KNN and RF.

Table 9.Classification of trained dataset S3 based on existing algorithm

Method	Accuracy	Sensitivity	Specificity	MCC	F-Measure
SVM	93.3525%	0.0202	0.9814	0.0030192	0.0202
KNN	91.0341%	0.05747	0.3546	-0.5363	0.05747
RF	93.0858%	0.01829	0.9703	-0.02110	0.01829

PREDICTION PERFORMANCE OF TRAINED DATASET S3 ON EXISTING ALGORITHM WITH SMOTH FILTER

Since, the dataset has non-unifrm number of sequences in their two classes. Therfore smote filter is used to make the data uniform using Weka function on the dataset trained by AACT. Further, SWM, KNN and RF are applied for classification and performance measurement. The result of SVM is decreased by approximately 1%, KNN increased by 0.8% and RF is decreased by 1%.

Table 10. Classification of trained dataset S3 based on existing algorithm with smoth filter

Method	Accuracy	Sensitivity	Specificity	MCC	F-Measure
SVM	92.6192%	0.10032	0.8435	-0.0479	0.10032
KNN	91.8812%	0.1604	0.09557	-0.4898	0.1604
RF	92.4678%	0.9394	0.9070	0.0008841	0.9394

PREDICTION PERFORMANCE OF TRAINED DATASET S3 BASED ON PROPOSED ALGORITHM

Now we apply the classifiers on trained datset by our proposed algorithm. The classifiers SVM, KNN and RF are applied and obtained 93.59%, 90.17% and 93.20% respectively as performance evaluation as shown in the Table 11. In comparison with Table 9 based on the existing algorithm, we obtained 0.2% greater in SVM, 0.1% in KNN but 0.2% less in RF.. The overalall maximum value is achieved by our updated algorithm with **SVM which is** 93.59%.

Table 11. Classification of trained dataset S3 based on proposed algorithm

Method	Accuracy	Sensitivity	Specificity	MCC	F-Measure
SVM	93.5987%	0.0225	0.9839	0.0108	0.0225
KNN	90.1723%	0.06029	0.3027	-0.5775	0.06029
RF	93.2089%	0.01804	0.09909	0.01727	0.01804

PREDICTION PERFORMANCE OF TRAINED DATASET S2 ON PROPOSED ALGORITHM WITH SMOTH FILTER

Similarly for the comparison purpose we apply the smoth filter to uniform the dataset trained by our updated algorithm. Further, SVM, KNN and RF are applied for classification and performance measurement and obtained 93.811%, 91.16% and 92.43% accuracy for SVM, KNN and RF respectively as shown in the Table 12.. The result of SVM is increased by 0.3%, KNN increased by 0.1% and RF is not changed. **The overalall maximum value is achieved by our updated algorithm with KNN which is 93.811%.**

Table 12. Classification of trained dataset S3 based on proposed algorithm with smoth filter

Method	Accuracy	Sensitivity	Specificity	MCC	F-Measure
SVM	93.8115%	0.1087	0.8593	-0.0245	0.1087
KNN	91.162%	0.01660	0.04282	-0.5288	0.01660
RF	92.43	0.0917	0.93	0.02007	0.0917

CONCLUSION:

In this research, an algorithm of Amino Acid Composition and translation is updated for prediction of mycobacterial membrane proteins and further appied on other datasets of Membrane and non-Membrane protein sequences. The proposed computational method produced improved results on datasets S1 and S2. Wheras, the results on dataset S1 are less as compared to the existing method. Since dataset S2 and S3 are two classes, whereas, dataset S1 has 4 classes, therefore, from the number of classes and from the achieved results; we can conclude that our method has given superior results on dataset with two classes.

This concludes that the updated algorithm is more accurate than the existing algorithm. In this work, Amino Acid composition and translation (AACT) as existing algorithm and the updated version of AACT are used for feature extraction to train the datasets. For classification of the trained data, SVM, KNN and RF are used. For uniforming the datasets smoth filter is used. The predicted results yield the overalall accuracy of 96.54% with KNN which is imum for the dataset S2 by the updated algorithm. The proposed model also achieved an overall accuracy of 93.811%.for prediction of dataset S3 as compared to the existing algorithm after passing through smoth filter.. Therefore, it is anticipated that proposed method is significantly improved the result as compare to the existing method for dataset with two classes and provide information for further studies on membrane proteins.

REFERENCES

[1]. Chou, K. C., & Elrod, D. W. (1999). Prediction of membrane protein types and subcellular locations. *Proteins: Structure, Function, and Bioinformatics*, *34*(1), 137-153.

[2]. Chou, K. C. (2001). Prediction of protein cellular attributes using pseudo-amino acid composition. *Proteins: Structure, Function, and Bioinformatics*, *43*(3), 246-255.

[3]. Wang, M., Yang, J., Liu, G. P., Xu, Z. J., & Chou, K. C. (2004). Weighted-support vector machines for predicting membrane protein types based on pseudo-amino acid composition. *Protein Engineering Design and Selection*, *17*(6), 509-516.

[4]. Chou, K. C., & Cai, Y. D. (2005). Prediction of membrane protein types by incorporating amphipathic effects. *Journal of chemical information and modeling*, *45*(2), 407-413.

[5]. Chou, K. C., & Cai, Y. D. (2005). Using GO-PseAA predictor to identify membrane proteins and their types. *Biochemical and biophysical research communications*, *327*(3), 845-847.

[6]. Augen, J. (2004). *Bioinformatics in the post-genomic era: Genome, transcriptome, proteome, and information-based medicine*. Addison-Wesley Professional.

[7]. Fulekar, M. H. (Ed.). (2009). *Bioinformatics: applications in life and environmental sciences*. Springer Science & Business Media.

[8]. Leavitt, H. J., & Whisler, T. L. (1958). *Management in the 1980's*. November.

[9]. Krogh, A., Larsson, B., Von Heijne, G., & Sonnhammer, E. L. (2001). Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. *Journal of molecular biology*, *305*(3), 567-580.

[10]. Wallin, E., & Heijne, G. V. (1998). Genome-wide analysis of integral membrane proteins from eubacterial, archaean, and eukaryotic organisms. *Protein Science*, *7*(4), 1029-1038.

[11]. Jones, D. T. (1998). Do transmembrane protein superfolds exist?. *FEBS letters*, *423*(3), 281-285.

[12]. Gao, Q. B., Ye, X. F., Jin, Z. C., & He, J. (2010). Improving discrimination of outer membrane proteins by fusing different forms of pseudo amino acid composition. *Analytical biochemistry*, *398*(1), 52-59.

[13]. Russell, R. B., & Eggleston, D. S. (2000). New roles for structure in biology and drug discovery. *Nature Structural & Molecular Biology*, *7*, 928-930.

[14]. Russ, A. P., & Lampel, S. (2005). The druggable genome: an update. *Drug discovery today*, *10*(23), 1607-1610.

[15]. Singer, S. J., & Nicolson, G. L. (1972). The fluid mosaic model of the structure of cell membranes. *Membranes and Viruses in Immunopathology; Day, SB, Good, RA, Eds*, 7-47.

[16]. Wallin, E., & Heijne, G. V. (1998). Genome-wide analysis of integral membrane proteins from eubacterial, archaean, and eukaryotic organisms. *Protein Science*, *7*(4), 1029-1038.

[17]. Bagos, P. G., Liakopoulos, T. D., Spyropoulos, I. C., & Hamodrakas, S. J. (2004). A Hidden Markov Model method, capable of predicting and discriminating β-barrel outer membrane proteins. *BMC bioinformatics*, *5*(1), 1.

[18]. Fairman, J. W., Noinaj, N., & Buchanan, S. K. (2011). The structural biology of β-barrel membrane proteins: a summary of recent reports. *Current opinion in structural biology*, *21*(4), 523-531.

[19]. Afridi, T. H., Khan, A., & Lee, Y. S. (2012). Mito-GSAAC: mitochondria prediction using genetic ensemble classifier and split amino acid composition. *Amino Acids*, *42*(4), 1443-1454.

[20]. Ung, P., & Winkler, D. A. (2011). Tripeptide motifs in biology: targets for peptidomimetic design. *J. Med. Chem*, *54*(5), 1111-1125.

[21]. Kumar, M., Gromiha, M. M., & Raghava, G. P. (2011). SVM based prediction of RNA-binding proteins using binding residues and evolutionary information. *Journal of Molecular Recognition*, *24*(2), 303-313.

[22]. Chou, K. C. (2011). Some remarks on protein attribute prediction and pseudo amino acid composition. *Journal of theoretical biology*, *273*(1), 236-247.

[23]. Ding, H., Deng, E. Z., Yuan, L. F., Liu, L., Lin, H., Chen, W., & Chou, K. C. (2014). iCTX-Type: a sequence-based predictor for identifying the types of conotoxins in targeting ion channels. *BioMed research international*, *2014*.

[24]. Hayat, M., & Iqbal, N. (2014). Discriminating protein structure classes by incorporating pseudo average chemical shift to Chou's general PseAAC and support vector machine. *Computer methods and programs in biomedicine*, *116*(3), 184-192.

[25]. Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.