Protein superfamily classification using Machine/Deep Learning Approaches:

1. Introduction:

Proteins are large, complex biomolecules that play an essential role in the body. Proteins are made up of thousands of tiny units connected in a long chain called amino acids. Amino acids are simply represented by alphabets. There are 20 different most common and five uncommon types of amino acids that can be combined to form a protein. They are arranged randomly in sequences that build millions of unique proteins. Each protein has a unique 3-dimensional structure and its specific functions, which are governed by a sequence of amino acids. A protein family is a group of proteins that share a common evolutionary origin, reflected by their functions and similarities in sequence or structure. Usually, proteins are arranged in hierarchies' designs where proteins share a common ancestor subclassified into smaller groups. A large group of distantly related proteins is the superfamily, whereas a small group of closely related proteins is called a subfamily. The proteins with similar structural and functional domains are classified as members of a specific family. The traditional method to identify protein families is complicated and time-consuming. So, this project aims to explore many machine learning and deep learning methods that can be used to classify protein families.

2. Data collection and Wrangling:

The dataset is downloaded from Kaggle (https://www.kaggle.com/shahir/protein-data-set), which is initially retrieved from Protein Data Bank (PDB). Two data files are available on the database; the first data file ("no_dups") contains chemical and physical properties such as phValue, residueCount, resolution, density, etc. (total 14 features with 141401 instances) and corresponding classifications. The second file contains amino acid sequences and related classes (a total of five

features with 104812 cases). The two data files are merged based on the “structureId”. Out of all the macromolecules, approximately 87% belong to the protein family (approximately 88000), so the remaining other macromolecule types are excluded from this study (Fig 1).

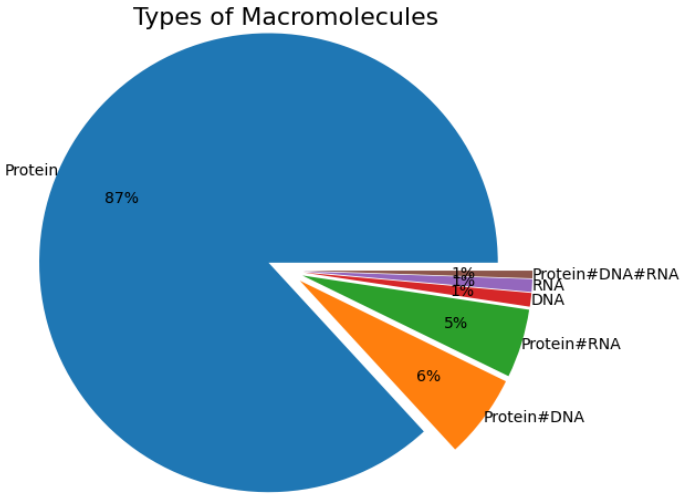


Fig 1: Pie Chart of Macromolecule types

Our goal was to predict the protein classes given their properties and sequence. Therefore, we observed the target variable “classes.” Several entries of classes labeled as different styles such as ‘slashes, comma, parenthesis, etc.’ We identified and removed the redundant labels. After removing the redundant entries, the total number of classes was 2684. The heat map is created to detect the missing data. Only a small proportion of data are missing, so rows with missing data are removed. Also, some irrelevant columns for the classification, such as ‘chainId’ are not considered for the further steps. After cleaning the data, we got the single data file with 87098 samples with features such as ‘experimentalTechnique’, ‘residueCount,’ ‘resolution,’ ‘structureMolecularWeight,’ ‘densityMatthews,’ ‘densityPercentSol’, ‘phValue’, ‘sequence’ and a target column ‘classes’ with 2684 classes.

3. Exploratory Data Analysis and Feature Engineering:

There are 20 most common protein classes in the dataset. The class frequency of each class is plotted in Fig 2.

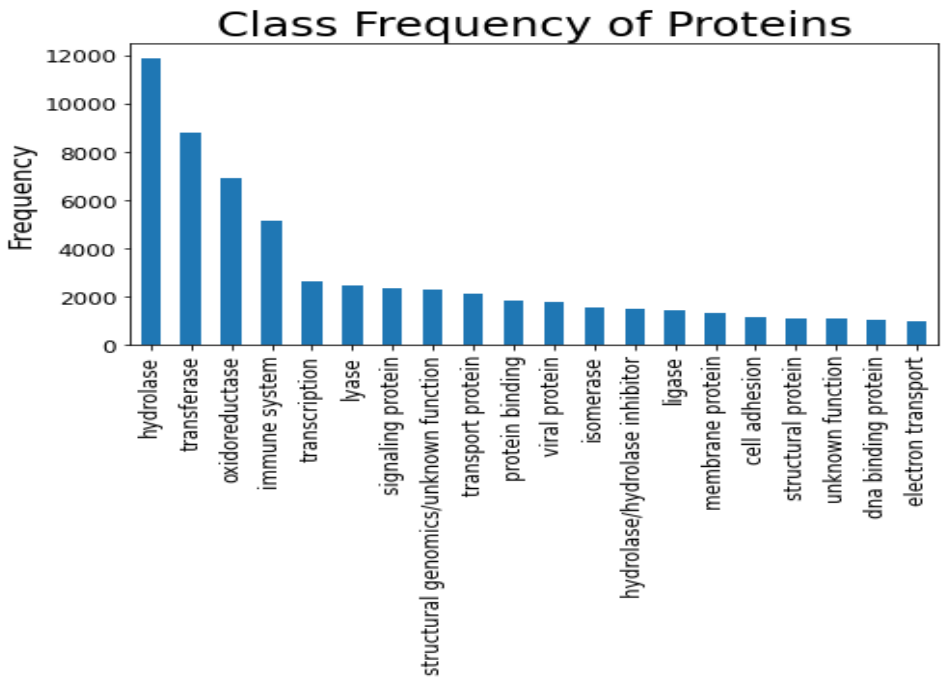


Fig 2: Frequency distribution of 20 most common classes.

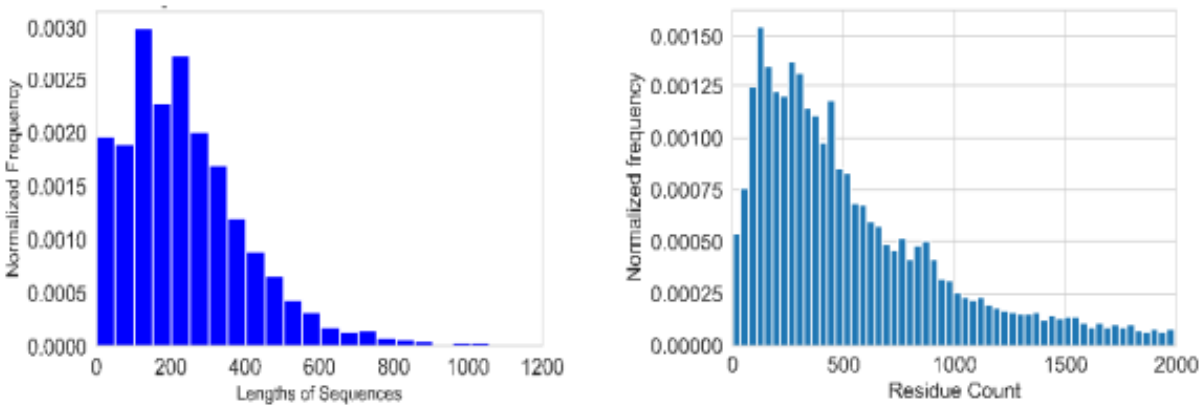


Fig 3: Normalized frequency distribution of length of the sequence (left) and Residue count(right).

Fig 2 shows that the most occurring class is “hydrolase” followed by “transferase” and “oxidoreductase”. Most of the sequence’s lengths are less than 500, with the most occurring in the range of 150 to 200 (Fig 3, left). In this project, the sequence with length than1200 units is excluded, resulting in only 15 classes in the dataset, as shown below.

```
([('hydrolase', 1), ('transferase', 2), ('oxidoreductase', 3), ('immune syste
m', 4), ('transcription', 5), ('lyase', 6), ('signaling protein', 7), ('struct
ural genomics/unknown function', 8), ('transport protein', 9), ('protein bindi
ng', 10), ('viral protein', 11), ('isomerase', 12), ('hydrolase/hydrolase inhi
bitor', 13), ('ligase', 14), ('membrane protein', 15)])
```

These 15 classes of the protein families are considered as a dependent variable (“target”) for machine learning and deep learning models.

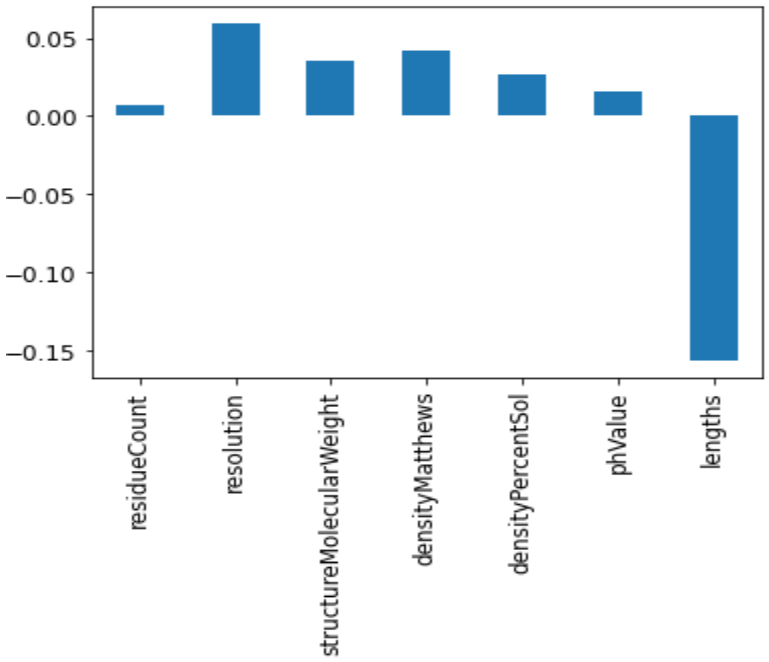


Fig 4: Correlation of class counts with other features

Fig 4 shows that the correlation between class counts with other remaining features. It is seen that all the features except the length are positively correlated with class counts. The length of the sequence is highly negatively correlated with the class counts.

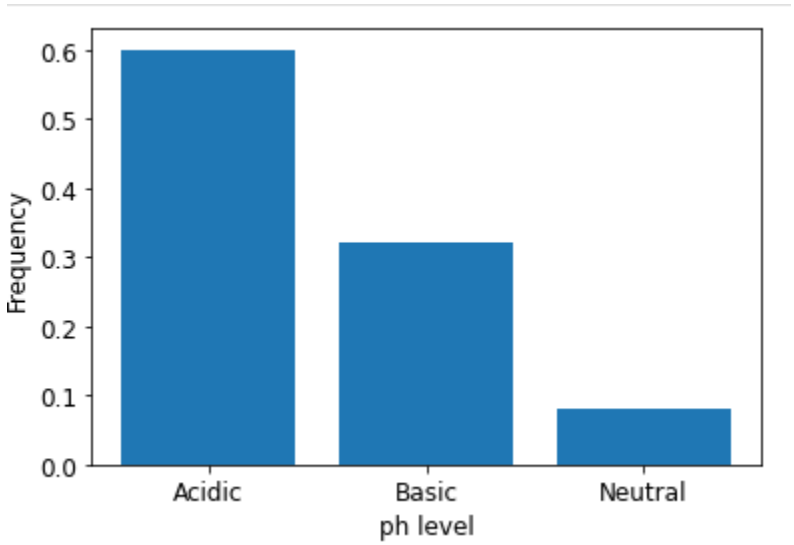


Fig 5: The frequency distribution of ph level types.

To understand the distribution of features, the frequency distributions of ph level types are plotted in fig 5. The ph value is grouped into three groups i.e., acidic (ph value < 7), basic (ph value >7), and neutral (ph value =7). The acidic type has the highest occurrence (~ 60%) followed by the basic type (~32%).

After the EDA, the feature engineering was done. Each categorical class was converted into numeric types by one-hot encoding method. All the features are standardized so the mean of each feature is 0 and the standard deviation of 1.

4. Modeling

Several machine learning and deep learning models are tested. In the first part, six numerical features (phValue, structureMolecularWeight, residueCount, densityPercentSol, resolution, and

densityMatthews) are used to train the machine learning models to predict the classes of the protein family. In contrast, in the second part, Deep learning is trained using sequence of amino acids to predict the classes of the protein family. It should be noted that the outcome of machine learning and deep learning methods do not compare because different features are used to predict the class.

4.1 Machine learning models:

Three machine learning models, decision tree (DT), random forest (RF), and K nearest neighbor (KNN), are trained and tested. 70% of the data are used to train the model, whereas the remaining 30% are used to test the model. Three success matrices (i.e., accuracy, recall, and precision scores) are mainly used to evaluate the performance of the models. Each model was tuned with the best hyperparameter, and the skill scores are presented in table1.

Models	Accuracy	Recall	precision
Decision Tree	0.43	0.43	0.49
Random Forest	**0.52**	**0.52**	**0.57**
KNN	0.28	0.28	0.27

Table 1: Skills scores of machine learning models

Table 1 shows that the RF performs better than DT and KNN. KNN skill scores are worst compared to the others. RF's accuracy, recall, and precision scores are 0.52, 0.52, 0.57, respectively.

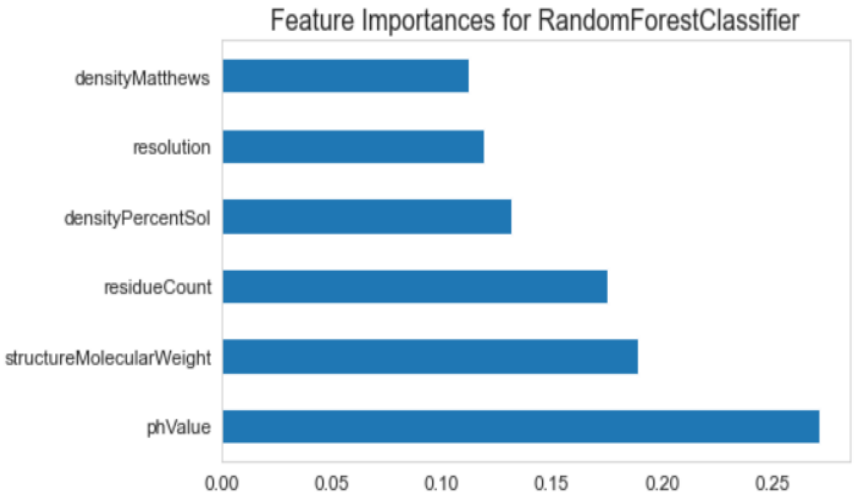


Fig 6: RF classifier feature importance

Fig 6 shows the feature importance of the RF classifier. It is seen that phValue is the most important feature (28%) to classify the data. Although RF performs better than DT and KNN, these scores are not impressive. In other words, ML models have difficulty classifying the protein family classes. Probably more features and datasets are needed to improve the model performance.

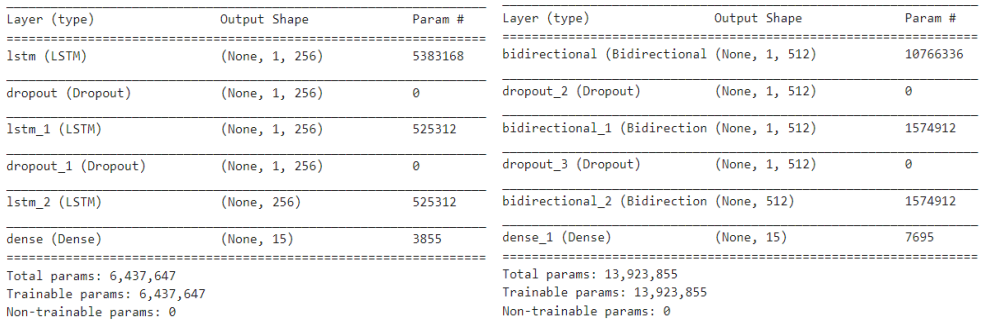
4.2 Deep learning (DL) models:

A recurring Neural Network (RNN) is a sequential model that is used to train the sequential data. In this project, a special kind of RNN, i.e., Long Short-term Memory (LSTM) and Bidirectional LSTM (BiLSTM), is developed to predict the protein family classes. The sequence of the amino acids is used as a feature, whereas protein classes are used as a label. As we know that the sequence of amino acids is a combination of alphabetic letters, it is necessary to convert them into numerical values. We applied a bag of word vectorization techniques (tf-idf) from Scikit-learn library for that purpose. This method has been widely used to vectorize texts (text mining) in Natural Language Processing (NLP). After converting the characters of the sequences to the numerical values, we subjected those vectors as inputs to the DL models for multi-class classification of protein families.

Note that LSTM models are known to train the sequential data. It has the capability to train the long-term dependency data because it has a special kind of cell unit to remember and forget long-term dependencies.

Model architecture:

Both LSTM and BiLSTM were trained using 75% of the data and tested on the remaining 25%. Total 3 LSTM and BiLSTM layers of 256 units are used separately. The details of the model summary are presented in Figure 7.



Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 1, 256)	5383168
dropout (Dropout)	(None, 1, 256)	0
lstm_1 (LSTM)	(None, 1, 256)	525312
dropout_1 (Dropout)	(None, 1, 256)	0
lstm_2 (LSTM)	(None, 256)	525312
dense (Dense)	(None, 15)	3855

Total params: 6,437,647
Trainable params: 6,437,647
Non-trainable params: 0

Layer (type)	Output Shape	Param #
bidirectional (Bidirectional	(None, 1, 512)	10766336
dropout_2 (Dropout)	(None, 1, 512)	0
bidirectional_1 (Bidirection	(None, 1, 512)	1574912
dropout_3 (Dropout)	(None, 1, 512)	0
bidirectional_2 (Bidirection	(None, 512)	1574912
dense_1 (Dense)	(None, 15)	7695

Total params: 13,923,855
Trainable params: 13,923,855
Non-trainable params: 0

Fig 7: Model architecture of LSTM (left) and BiLSTM (right).

Both models have almost the same configuration. These models are trained for 100 epochs with a batch size of 50. The main success matrix is the accuracy score; however, precision, recall, AUC score, and confusion matrix are also generated.

Models	Accuracy	Recall	precision	AUC
LSTM	0.59	0.59	0.61	0.83
BiLSTM	0.59	0.59	0.60	0.84

Table 2: Skills scores of deep learning models

Table 2 shows skill scores of both LSTM and BiLSTM models. Each model performs almost similarly with an accuracy score of 59%. Both models' precision and recall scores are approximately 60%, whereas AUC scores are around 84%. Based on the scores presented in Table 2, the LSTM (either 1 directional or bidirectional) model can classify the protein families, but more effort should be put into building the model. Some potential steps to improve the model might be gathering more data, adding more hidden layers, or increasing model complexity. The confusion matrix for both the training and testing dataset has been generated and presented in Fig 8.

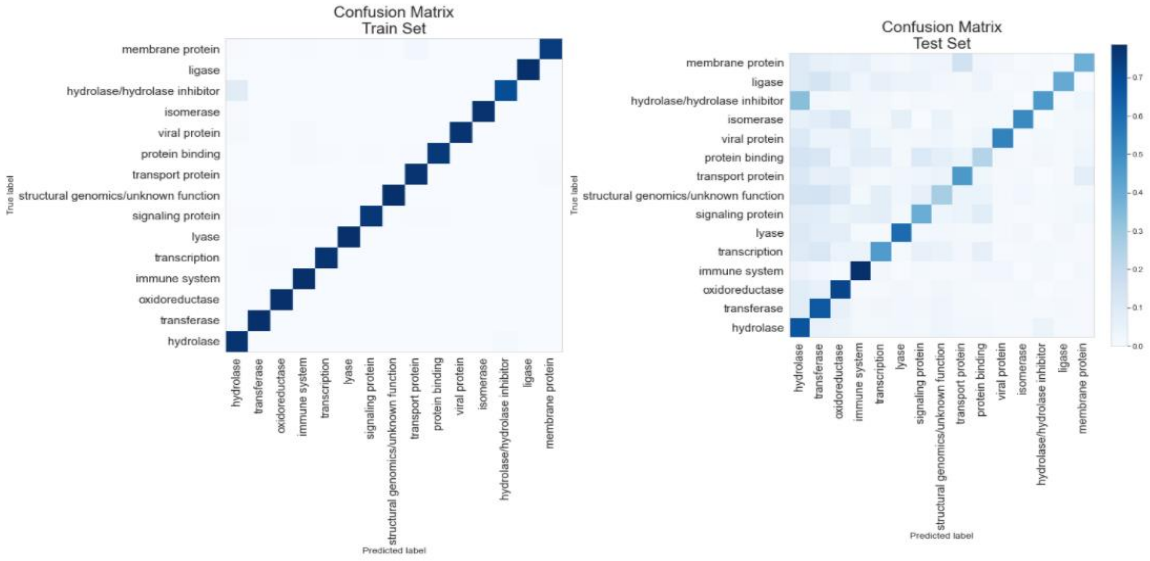


Fig 8: confusion matrix of LSTM classification model for training (left) and test (right) dataset.

5. Conclusion:

This study evaluates the machine learning and deep learning models to classify the class of the protein families based on the physical and chemical properties of macroscopic features and sequence of amino acids. Random forest classifier performs better compared to the decision tree and k nearest neighbor with an accuracy score of 52%. However, it is concluded that the machine learning model has difficulty classifying the protein family classes.

Both LSTM and BiLSTM have almost similar performance to classify the protein family class. Accuracy, recall, and precision scores are approximately 60%, and AUC scores are 83%.

LSTM is better than machine learning models to predict the protein family class. However, they do not compare because the different features are used to train the models. LSTM uses a sequence of amino acids, whereas machine learning models use seven other features to train the model. Although LSTM shows potential to classify protein family classes, the model skill scores should be improved for practical purposes. This can be done by increasing data samples, increasing model complexity, such as adding more hidden layers or units. Also, the model can be tunned using a variety of hyperparameters values.