Classification of Biological Data using Deep Learning Technique

*Azha Javed * and Muhammad Javed Iqbal*

Department of Computer Science
University of Engineering and Technology, Taxila, Pakistan.
theazhajaved@gmail.com, javed.iqbal@uettaxila.edu.pk

**Corresponding Author*

Abstract

A huge amount of newly sequenced proteins is being discovered on daily basis. The main concern is how to extract the useful characteristics of sequences as the input features for the network. These sequences are increasing exponentially over the decades. However, it is very expensive to characterize functions for biological experiments and also, it is really necessary to find the association between the information of datasets to create and improve medical tools. Recently machine learning algorithms got huge attention and are widely used. These algorithms are based on deep learning architecture and data-driven models. Previous work failed to properly address issues related to the classification of biological sequences i.e. protein including efficient encoding of variable length biological sequence data and implementation of deep learning based neural network models to enhance the performance of classification/ recognition systems. To overcome these issues, we have proposed a deep learning based neural network architecture so that classification performance of the system can be increased. In our work, we have proposed 1D-convolution neural network which classifies the protein sequences to 10 top common classes. The model extracted features from the protein sequences labels and learned through the dataset. We have trained and evaluate our model on protein sequences downloaded from protein data bank (PDB). The model maximizes the accuracy rate up to 96%.

Keywords: Deep learning, 1D Convolution neural network, Protein sequences classification

1. Introduction

Deep learning has presented more benefits in respect of data classification and became the most popular domain of bioinformatics [1]. The development and growth of biological databases that stores the information of data including sequences of DNA, RNA, protein and other macromolecules need advance computational tools [2]. These sequences are increasing exponentially over the decades. However, it is very expensive to characterize functions for biological experiments and also, it is really necessary to find the association between the information of datasets to create and improve medical tools. For classification purpose, several classification techniques were developed [3], [4]. These techniques can be divided in two parts: Sequence alignment and Machine learning algorithms. Sequence alignment method distributes newly discovered proteins on the basis of similarity with existing protein sequences. The challenge with sequence alignment methods is that they are time taking and expensive.

Recently machine learning algorithms [5] got huge attention and are widely used. These algorithms are based on deep learning architecture and data-driven models. An emerging

technique of deep learning is machine learning [6]. The deep learning architecture is more useful as it has multiple non-linear transforming hidden layers which help with understanding the biological principles, solve complex problems and can cover more raw data during the training. Deep learning approach is based on prediction which makes it more capable. In order to drive a prediction model, it plays crucial part in computation of big data science. Deep learning model can deal with huge amount of raw data and limited amount of labeled data. It can extract useful information from complex systems [7]. To train a network, deep learning model uses two-level approach, pre-training and fine tuning. Also large data size and computational power of computers makes deep learning model more reliable. As high performance computing systems are needed in order to train extensive deep learning model. For that purpose, GPU based models [8] can help by reduces the training time from weeks to several hours. Deep neural networks or DNN model needs modified approach for training. There are two methods, supervised and unsupervised learning. In supervised learning method [9], labeled data is used and target outputs are already defined. For unsupervised learning, data is not labeled and training does not have any target output. Supervised learning is used for classification and regression where unsupervised learning is used for feature extraction [10], clustering, association, and generalization.

This work focuses on the development of an efficient deep network model designed for protein sequence classification and comparison with the previous models. A protein sequence consists of 20-letter amino acid alphabet. We have used convolutional neural network (CNN) [11], a deep neural network, for the classification purposes and tried to achieve maximum accuracy.

2. Literature Review

In this research, we focused on the deep sequence encoding method to represents the variable length, less explored biological sequence data for better numeric feature extraction and classification performance of system. A research paper related to protein family classification using deep learning [12] proposed a model that dealt with proteomics classification using deep learning. The proposed model focused on sequences classification of protein vector for the representation of proteomics. Model used data of protein family information from Protein family database known as “PFam”. They used swiss-Prot database which contains nearly 40433 protein sequences. These sequences were passed to protein representation layers. Protein sequences were transformed into numeric vectors. For the sequence representation, *n-gram* and *keras* embedding were used. In n-gram, 3-gram was used with the feature hashing length of 1000. For optimal feature extraction they used different models of deep learning including CNN, RNN, LSTM, and DNN. These features went through fully connected layer where softmax, an activation function was used. At the end, cross entropy was used to minimize loss. To implement all algorithms, tensorflow with keras framework was used.

Jie Hou *et al.* introduced a model named DeepSF [13] for mapping protein sequences to folds. This paper proposed 1D Convolutional neural network which was helpful in recognition of folds and relationship study of sequence and structure. Deep convolutional extracted hidden features of fold from variable length of protein sequence which can be used in numerous applications of protein data analysis such as protein evaluation, grouping, and structure prediction. Model directly assigned folds to a protein sequence were a big accomplishment and surely different from traditional comparison based methods.

Man Li *et al.* suggested a conjugation technique [14] based on convolutional neural network (CNN). An improved n-gram technique had been used for feature extraction with feature weighting strategy. G-protein Coupled Receptors (GPCRs) are the major and most different group of membrane receptors in eukaryotes. Term Frequency -Inverse Document Frequency (TF-IDF) weighting strategy used for the classification of GPCR at different levels: Family level, subfamily level I and II. For protein classification, GPCR is one of the most challenging and valued datasets and the main problem in GPCRs sequence classification is feature selection based on sequence characteristic. The determined accuracy of performance comparisons achieved by the suggested technique at GPCRs family level, subfamily levels I and II was 98.34%, 98.13% and 96.47% respectively. A widespread research could be conducted to evaluate performance of other protein families.

Muhammad Javed Iqbal *et al.* [15] suggested a distance-based feature-encoding technique for precise demonstration of feature extraction in protein sequence classification of amino acid sequences. The work focused on the improvement of feature-encoding method for effective abstraction of less similar sequence features from a variable length protein sequence. Also, this research tried to classify sequences into several super families with considerably better classification accuracy, specificity, sensitivity and less error rate. The dataset from UniProtKB protein database was used for experiments. Different phases of the research includes training and testing of variable length yeast protein sequences, distance-based feature encoding technique, implementation of classification algorithm, and evaluation using different performance measures metrics including precision, specificity, sensitivity and F-Measure. Comparison showed that the maximum average accuracy acquired using the decision tree was 91.2% and it gave better results than the other classifiers in term of classification accuracy, specificity, F-measure and MCC.

3. Methodology

In this section we have discussed detailed architecture of the proposed CNN model but first let's go through the steps taken in order to prepare data of protein sequences for the model. The dataset used in our research was retrieved from protein data bank (PDB) from Research Collaboratory for Structural Bioinformatics (RCSB). Dataset is discussed in detail later. After protein sequence processing, the data is visualized in the form of histogram. *Figure 1* explains the techniques used in this research.

In this research, we described our improved convolutional neural network and other existing methods that were used as baselines for performance comparison. We have proposed an improved version of convolutional neural network for protein family prediction and sequence classification. Model takes raw protein sequences as an input and concludes family of protein as an output. First, the model preprocesses dataset and visualizes it for better understanding. In our experiment, we have selected 10 top protein classes which reduce the dataset to the samples belonging to the selected protein classes. The length of the sequences is from few amino acids to thousands of amino acids. There are many methods for protein sequences encoding out there including distance based encoding [4], [16] that captures statistical characteristics of protein sequences. In our work we transformed the labels in to one hot vector representation using LabelBinarizer from sklearn.preprocessing.

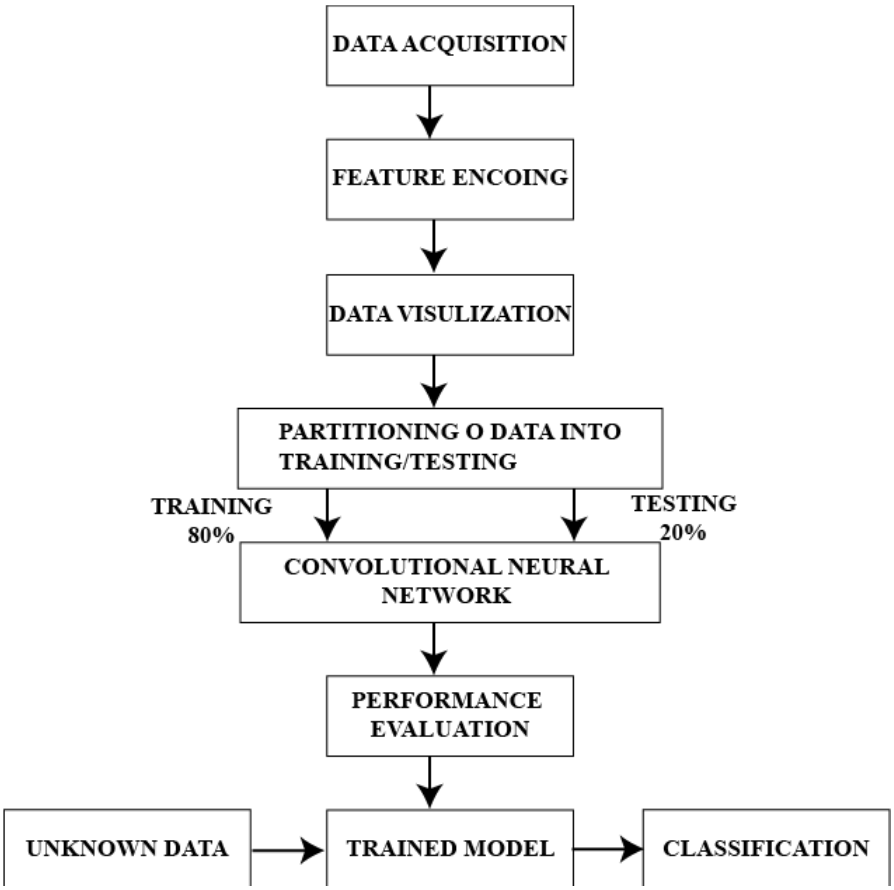

Figure 1: Flow diagram of proposed Protein Sequence Classification

For further processing, we have used keras for protein sequence processing including translation of every character of sequence into an integer. It also ensures that every sequence has same length, in our case it is equals to the maximum length in the protein sequence. For splitting our dataset in to training and testing modules, we have used sklearn from the model selection tool. Our model consists of 10 hidden layers including an input layer, one hidden embedding layer, two subsequent layers of convolution and pooling, a flatten layer, and a fully connected layer, and an output layer that uses softmax as an activation function. We have used dropout technique for preventing over fitting.

3.1 Dataset used in the proposed model

The main dataset used in our research for training and testing was obtained from protein data bank (PDB) from RCSB. The protein data bank (PDB) contains three-dimensional structural data of large biological molecules, such as proteins and nucleic acids [17]. Protein data base facilitated the life science community and helped the human survival by providing data for study about different diseases and come with new drugs.

The data can easily access from internet through the member organizations websites including PDBe [18], PDBj [19], RCSB [20], and BMRB [21]. It is allowed to each member's website that they can take structural data and process that data.

3.2 Training, validation and test datasets

The PDB database along with its holdings list is updated weekly [22] and as of 1 April 2020, the PDB contains 150423 proteins and 3466 nucleic acids. In our research we have used two data files. First data file contains protein metadata including protein classification details, extraction methods, etc. Second data file contains more than 400,000 protein structure sequences.

Initially both data files are merged on the basis of structure ID. But before we get started we have to clean data and fortunately pandas make this very easy. We have dropped rows without labels and without sequence and only one macromolecule type, protein is selected. After cleaning data we get 346321 protein sequences.

In dataset there are 18 labels named structureId, classification, experimentalTechnique, macromoleculeType_x, residueCount_x, resolution, structureMolecularWeight, crystallizationMethod, crystallizationTempK, densityMatthews, densityPercentSol, pdbxDetails, phValue, publicationYear, chainId, sequence, residueCount_y, macromoleculeType_y. *Figure 2* describes dataset structure of protein sequences.

```
<bound method NDFrame.describe of        structureId  ... macromoleculeType_y
4              101M  ...             Protein
7              102L  ...             Protein
8              102M  ...             Protein
11             103L  ...             Protein
12             103M  ...             Protein
...             ...  ...                 ...
471144         9XIA  ...             Protein
471145         9XIM  ...             Protein
471146         9XIM  ...             Protein
471147         9XIM  ...             Protein
471148         9XIM  ...             Protein

[346321 rows x 18 columns]>
```

Figure 2: Structure of Protein sequence dataset

For classification purpose, 10 most top common classes are considered including hydrolase, transferase, oxidoreductase, immune system, transcription, lyase, signaling protein, transport protein, protein binding, and viral protein where the number of instances per class is more than 6500. After sorting top classes, 188353 instances are obtained. Now the dataset is minimized to instances which are one of the top ten most common classes. The length of sequences varies from very few amino acids to several thousand amino acids. Next step uses label encoding and for that purpose we have used LabelBinarizer from sklearn.preprocessing. It is very simple to use and really efficient. In order to transform labels, we have applied a fit_transform function on the input data. Keras prefers inputs to have same lengths but in our case the sequences have different lengths. For that, we have used pad_sequences() function which will pad input sequences. As an output, it provides a transformed hot representation of string labels.

The next step is to encode sequences as integers for further processing, and to be used as input for our model. To process sequences, we are using keras library which gives basic tools to prepare data. Keras is a neural network library [23] and offers consistent high-level APIs. Keras provides the Tokenizer class which we are using for preparing sequences. It encodes every

character of the sequence into a number. Also to ensure that every sequence has equal length we are using pad_sequences() function. The maximum length of sequences is set to be 512.

Before starting to train our model we have to train our data. For this task, we split our dataset into two samples, training and testing. There is a tool named Model Selection in SKlearn library that we are using for splitting dataset. We are using a class in the library named *train_test_split.* By which we can simply split the dataset into the training and the testing datasets in several proportions. In this work we have set *test_size* to 0.2 that means our training data and testing data will split in the ratio of 80:20 respectively, which is a common practice in data science. The dataset is split into independent features dataset donated by X and dependent variables dataset donated by Y. The dataset X is further split into X_train, X_test and dataset Y into Y_train, Y_test.

3.3 Deep Convolutional Neural Network Model

We are purposing an improved deep 1D-Convolutional Neural Network model for the classification of protein sequences. Model consists of multiple layers including embedding, 1D-convolution, subsequent layers of convolution and pooling, fully connected layers, and dense layer. The dataset is used as input for the first layer that is embedding layer. Because of the recent accomplishment in NLP, word embedding is used which employed as *keras* layer. Our approach dedicated on applying NLP theory on protein sequences. The architecture of proposed model is shown below in *Figure 3*.

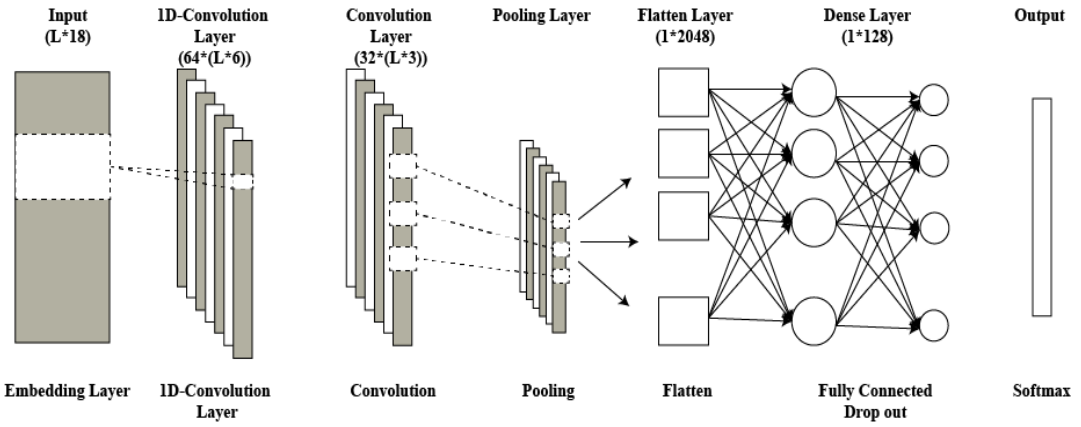

Figure 3: The architecture of improved 1D-convolutional neural network

In our model, we focused on applying word embedding on protein sequences. For each amino acid, there are only 20 different words. In result, we get embedded sequences. The dimension size of embedded sequences is set to 11 and maximum length of the sequence is set as the input length. The output of embedding layer is 2D vector with one embedding. The next layer in proposed model is 1D-convolution layer and directly connected to the embedded output layer. Convolution layer transforms an embedded protein sequence into the vector of features from sequence. To improve performance two layers of convolution layer following by max-pooling layer are used. The first layer has 64 filters with kernel size of 6 and the second layer has 32 filters of size 3. We have used “ReLU” activation function.

In this work we have used 1-max pooling [24] to focus on the existence of conserved regions. The pool size is set to 2. In order to prevent over fitting we adopt dropout after pooling layer. Dropout eliminates some neurons at training time. Flatten and pass activations into fully connected layers where the last layer is a *softmax* activation and size equivalent to the number of classes. The *softmax* layer is used to evaluate the possibility over each family class as shown in *figure 4*. Activation function chooses whether a sequence should be activated or not by computing weighted sum and further adding bias with it.

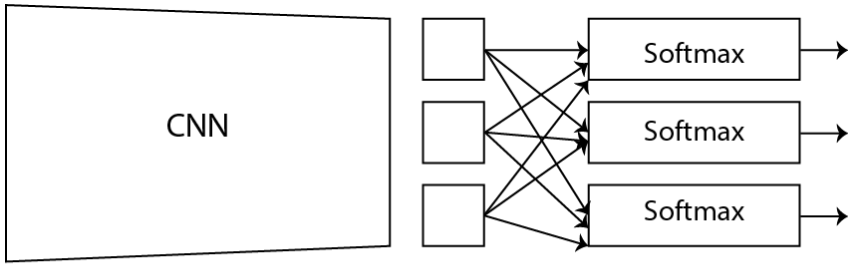

Figure 4: Implementation of Softmax function in CNN model

Later the calculated probabilities are helpful for determining the target class for the given inputs. In order to train our model, we have used cross-entropy as a loss function and adam as an optimizer. At the last layer of our model, we have used crossentropy function as shown in *figure 5*. It calculates loss between the labels and prediction. It is used for multi-class classification. When there are two or more label classes such as in our case there are ten classes provided as one-hot vector. We have used this loss to train proposed CNN to output a probability over the 10 classes for each sequence.

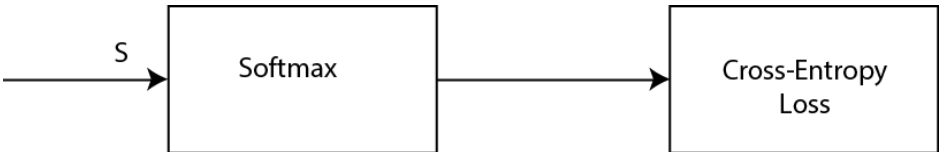

Figure 5: Implementation of Cross-entropy Loss

A general optimization technique for training deep neural network that we implemented in our work as well is Adaptive Moment Estimation (Adam) optimization [25]. It calculates different and adaptive learning rates. The optimization ends after training 20 epochs of the train dataset. We used two 1D-convolution layer having 64 and 32 filters respectively and kernel size 6 and 3 respectively. The hyper-parameters were set as follows: dropout rate, learning rate= 0.8, batch size = 128. We have used two filters with kernel size 6*6 and 3*3. The technique we used to reduce over fitting and improve generalization is dropout. [26] It randomly drops out some alphabets of sequences and when these alphabets are missing, other alphabets are obligatory make assumptions for missing alphabets and that's how they learn the representation of a sequence independently. Dropout technique does take more time. In proposed network the value of dropout is equals to 0.1. In order to control the weight to update in optimization algorithm we used learning rate. After performing multiple experiments, we fixed the value of learning rate equals to

0.8. The number of times the entire training set pass through the network is handled by number of epochs. Considering the performance of proposed technique, we used 25 epochs. Before the updating proposed model, batch size processed specific number of training dataset. In our model, the value of batch size is equals to 128.

4. Results

We trained our model on protein sequences obtained from the protein data bank (PDB) dataset from Research Collaboratory for Structural Bioinformatics (RCSB). After using label encoding method on protein sequence dataset, we extracted features from our data. As an output we got encoded data and used the obtained output of protein sequences as an input to train our model.

We have purposed 1D-convolution neural network architecture with hidden layers including embedding layer, two subsequent layers of convolution and pooling, flatten layer, and dense layer having softmax activation function. At the output layer we implemented an optimization technique called *adam*. Protein sequences were split into train and testing dataset. We applied proposed 1D-Convolutional neural network on testing dataset and achieved accuracy up to 96%. At the end we used confusion matrix to visualize the results.

4.1 Experimental Results

Initially two data files were imported from protein data bank. First data file contains protein metadata including protein classification details, extraction methods, etc. Second data file contains more than 400,000 protein structure sequences. The data files are merged on the basis of structure ID. The rows without labels and without sequence were dropped and only one macromolecule type, protein is selected. After cleaning data, we get 346321 protein sequences in the form of data frame.

In order to summarize protein sequences data frame, we used pandas.DataFrame.describe function and as an output we obtained summary statistics of the provided Data frame. Result index include count, mean, standard division minimum, 25% lower, and 75% upper percentile. Following *table 1* shows the statistics report.

Table 1: summary statistics of protein sequences

Count	346321.000000
Mean	4708.585243
Std	26484.309151
Min	3.000000
25%	398.000000
50%	856.000000
75%	1976.000000
Max	313236.000000

For classification purpose, we selected 18 labels and 10 most top common classes from protein sequence dataset. After sorting top classes, 188353 instances are obtained. By using matplot library we plotted a histogram of protein sequences in terms of classes and sequences frequency for better understanding as shown below in *figure 6*:

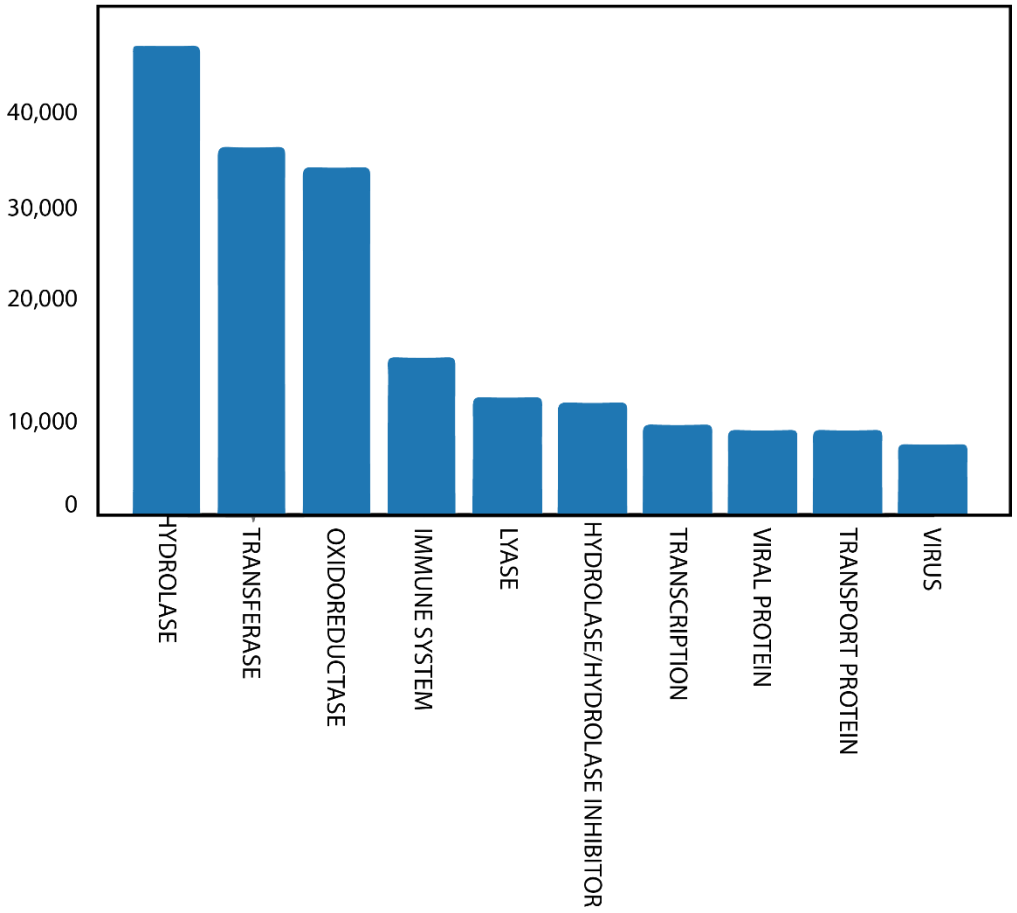

Figure 6: Visualization of data- x-axis represents protein sequences in terms of classes where on y-axis we have sequences frequency

After sorting and preparing protein sequences, the next step was label encoding. For that purpose we used LabelBinarizer from sklearn.preprocessing. In order to transform labels, we have applied a fit_transform function on the input data. Keras prefers inputs to have same lengths but in our case the sequences had different lengths. For that, we used pad_sequences() function and obtained a transformed hot representation of string labels as an output. By using shape () function, we obtained 188353 by 10 matrix.

The proposed 1D-Convolutional Neural Network model consist of multiple layers including embedding, 1D-convolution, subsequent layers of convolution and pooling, fully connected layers, and dense layer.

The summary of trained model is described in *table 2:*

Table 2: Summary of proposed model

Layer	**Output Shape (type)**	**Param #**
embedding_2 (Embedding)	(None, 256, 11)	286
conv1d_4 (Conv1D)	(None, 256, 64)	3584
max_pooling1d_4 (MaxPooling1)	(None, 128, 64)	0
conv1d_5 (Conv1D)	(None, 128, 32)	6176
max_pooling1d_5 (MaxPooling1)	(None, 64, 32)	0
dropout_1 (Dropout)	(None, 64, 32)	0
flatten_2 (Flatten)	(None, 2048)	0
dense_4 (Dense)	(None, 128)	262272
dense_5 (Dense)	(None, 10)	1290
Total params: 273,608 Trainable params: 273,608 Non-trainable params: 0		

To train protein sequences dataset, we split them into two samples, training and testing. The test_size was set to 0.2 which mean training and testing protein sequences were split in the ratio of 80:20 respectively. The dataset was split into independent features dataset donated by X and dependent variables dataset donated by Y. The dataset X is further split into X_train, X_test and dataset Y into Y_train, Y_test. The entire training set to pass through the network was set to 25 times.

To summarize the performance of our classification model, we used confusion matrix and accuracy rate. For that we have used sklearn. The classification report of our model is given below. By implementing the model, we have achieved the accuracy up to 96% which is increased by 3% than the earlier version. The following table 3 shows the train and test accuracy of proposed model.

Table 3: Train and test accuracy of proposed model

Train accuracy	0.9634727439242909
Test accuracy	0.9359719678267102

As mentioned earlier, we chose 10 top most common protein classes. Our model predicts the protein sequence class. We used *sklearn.metrics* for plotting the confusion matrix and the following *figure 7* shows the output matrix:

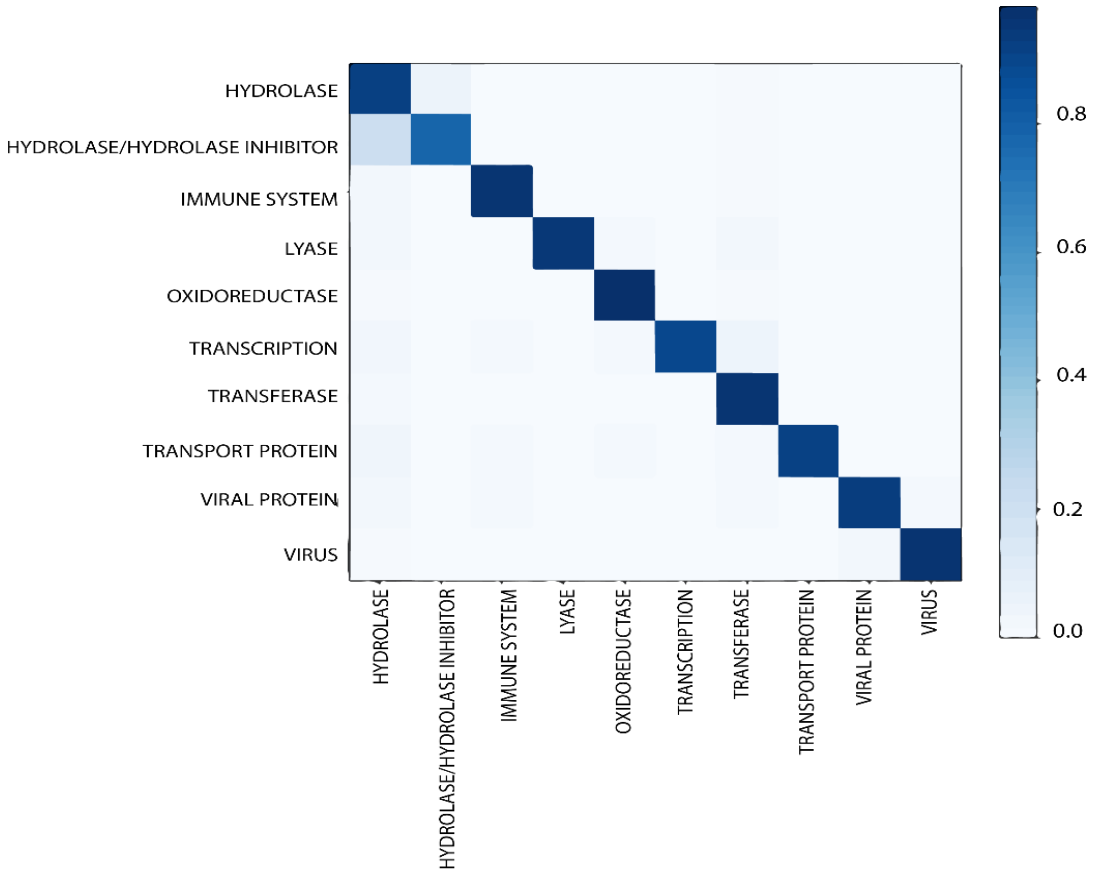

Figure 7: Confusion matrix - x-axis of the matrix represents predicted labels where on y-axis we have true labels

According to the results, we are able to achieve impressive measures. The following *table 4* shows the accuracy rate for each class and also average accuracy of our model:

Table 4: Classification Report of model

	PRECISION	**RECALL**	**F1-SCORE**	**SUPPORT**
HYDROLASE	0.91	0.92	0.91	9309
HYDROLASE/ HYDROLASE INHIBITOR	0.79	0.78	0.78	2249
IMMUNE SYSTEM	0.95	0.96	0.95	3056
LYASE	0.98	0.95	0.96	2351
OXIDOREDUCTASE	0.97	0.98	0.97	6856
TRANSCRIPTION	0.93	0.89	0.91	1791
TRANSFERASE	0.96	0.96	0.96	7329
TRANSPORT PROTEIN	0.97	0.92	0.94	1654
VIRAL PROTEIN	0.92	0.93	0.93	1732
VIRUS	0.97	0.96	0.97	1344
Weighted Average Accuracy	**0.94**	**0.94**	**0.94**	**37671**

According to the classification model, our proposed 1D-convolution neural network has achieved impressive accuracy rate. It has increased 3% from the previous version of the method. The training accuracy is recorded 96% and testing accuracy is recorded 93%. We have compared the results with existing techniques in following *table 5:*

Table 5: Comparison of Results with Existing

Reference	**Proposed Technique**	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Support**
[27]	Na ïve Bayes Model	0.76385	0.76	0.76	0.77	55774
[28]	CNN model	0.60045	0.60	0.60	0.59	7456
[29]	1D- CNN model	0.91006	0.91	0.91	0.91	37671
Proposed Model	Optimized 1D-CNN Model	0.93597	0.94	0.94	0.94	37671

5. Discussion

Previous researches have shown keen interest in structural prediction but the implementation of our improved 1D-Convolutional neural network showed the increased accuracy rate and improved prediction capability of classification. *Kernel of abharg* [27] used dataset contain different types of macromolecules of biological significance but the most of the data records were proteins. The work focused on classification of protein's family based on protein sequences by using machine learning approach. For this purpose they used *CountVectorizer* from *sklearn* with *4-grams.* The simple classification model achieved 76% considering the first 43 classes. As the model only used properties of 4 amino acids, opportunity of developing higher degree amino

acids in theory should be able to generate better accuracy. *Kernel of helme* [29] proposed convolutional neural network technique for protein sequence classification. They achieved much better accuracy than the other previous techniques and became motivation for us. Results suggested implementation of NLP-theory. In their technique the major concern was about over fitting. We successfully implemented dropout technique to overcome this issue.

By comparing our work with previous researches, no one has achieved such better accuracy. The accuracy of our model is 96% which has increased by 3% as compared to previous models. The evaluation of our model shows that it is a fast and accurate protein sequences classification model so far. The classification report of our model shows that we have achieved the accuracy up to 96% which is increased by 3% than the earlier version.

6. Conclusion

We have presented an optimized 1D-Convolutional neural network for the classification of protein sequences. The proposed network is fast and accurate as compared to existing techniques. The early versions of proposed methods couldn't achieve better accuracy and to our knowledge, proposed technique has successfully classified protein sequences with maximum accuracy so far. Our method presented a sequence encoding method to represents the variable length less explored protein sequence data for better numeric feature extraction. One-hot encoding method worked so well in this matter. We used dropout technique to reduce over fitting and improve generalization which certainly took more time but improved the classification performance. Our optimized 1D-Convolutional neural network increased the classification performance of system and achieved accuracy up to 96%.

This work will be helpful in the classification of protein structure sequences and structure prediction which will lead into the analysis of several diseases and help in the drug construction. Improved architecture of CNN model can successfully predict the class of the protein sequences. The accuracy of proposed methodology can be improved which is one of the substantial direction towards future work.

References

[1] Min, S., B. Lee, and S. Yoon, Deep learning in bioinformatics. Briefings in Bioinformatics, 2016. 18(5): p. 851-869.

[2] Jurtz, V.I., et al., An introduction to deep learning on biological sequence data: examples and solutions. Bioinformatics, 2017. 33(22): p. 3685-3690.

[3] Saha, S. and R. Chaki. A Brief Review of Data Mining Application Involving Protein Sequence Classification. 2013. Berlin, Heidelberg: Springer Berlin Heidelberg.

[4] Iqbal, M.J., et al., Computational Technique for an Efficient Classification of Protein Sequences With Distance-Based Sequence Encoding Algorithm. Computational Intelligence, 2017. 33(1): p. 32-55.

[5] Larra ñaga, P., et al., Machine learning in bioinformatics. Briefings in Bioinformatics, 2006. 7(1): p. 86-112.

[6] Rav ì, D., et al., Deep learning for health informatics. IEEE journal of biomedical and health informatics, 2016. 21(1): p. 4-21.

[7] Li, Y., et al., Deep learning in bioinformatics: Introduction, application, and perspective in the big data era. Methods, 2019. 166: p. 4-21.

[8] Mittal, S. and S. Vaishay, A survey of techniques for optimizing deep learning on GPUs. Journal of Systems Architecture, 2019. 99: p. 101635.
[9] Caruana, R. and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. in Proceedings of the 23rd international conference on Machine learning. 2006.
[10] Wang, J.T.-L., et al., New techniques for extracting features from protein sequences. IBM Systems Journal, 2001. 40(2): p. 426-441.
[11] O'Shea, K. and R. Nash, An Introduction to Convolutional Neural Networks. 2015.
[12] K S, N., et al., Protein Family Classification using Deep Learning. 2018.
[13] Hou, J., B. Adhikari, and J. Cheng, DeepSF: deep convolutional neural network for mapping protein sequences to folds. Bioinformatics, 2017. 34(8): p. 1295-1303.
[14] Man, L., L. Cheng, and G. Jingyang. An efficient CNN-based classification on G-protein Coupled Receptors using TF-IDF and N-gram. in 2017 IEEE Symposium on Computers and Communications (ISCC). 2017.
[15] Iqbal, M.J., et al. A distance-based feature-encoding technique for protein sequence classification in bioinformatics. in 2013 IEEE International Conference on Computational Intelligence and Cybernetics (CYBERNETICSCOM). 2013.
[16] Zamani, M. and S.C. Kremer. Amino acid encoding schemes for machine learning methods. in 2011 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW). 2011. IEEE.
[17] ww, P.D.B.c., Protein Data Bank: the single global archive for 3D macromolecular structure data. Nucleic acids research, 2019. 47(D1): p. D520-D528.
[18] Protein data bank in europe Available from: https://www.ebi.ac.uk/pdbe/.
[19] Protein data bank japan Available from: https://pdbj.org/.
[20] RSCB. Available from: http://www.rcsb.org/
[21] Biological Magnetic Resonance Data Bank Available from: http://www.bmrb.wisc.edu/.
[22] PDB Current Holdings Breakdown. Available from: http://www.rcsb.org/pdb/statistics/holdings.do.
[23] Keras. Available from: https://keras.io/.
[24] Boureau, Y.-L., J. Ponce, and Y. LeCun. A theoretical analysis of feature pooling in visual recognition. in Proceedings of the 27th international conference on machine learning (ICML-10). 2010.
[25] Kingma, D.P. and J.A. Ba, A method for stochastic optimization. arXiv 2014. arXiv preprint arXiv:1412.6980, 2019. 434.
[26] Salakhutdinov, N.S.a.G.H.a.A.K.a.I.S.a.R., Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research, 2014. 15: p. 1929-1958.
[27] Bhargava, A. Predicting Protein Classification. 2017; Available from: https://www.kaggle.com/abharg16/predicting-protein-classification/code.
[28] Ofer, D., Protein Sequence family Classification. 2019.
[29] Helme, Protein Sequence Classification. 2018.