Protein Classification using CNN

Arun Kumar
Dept. of Comp. Science & Applications
Chaudhary Ranbir Singh University
Jind, India
aarunbhardwaj@gmail.com

Vishal Verma
Dept. of Comp. Science & Applications
Chaudhary Ranbir Singh University
Jind, India
vishal.verma@crsu.ac.in

***Abstract*—Living organisms have a variety of macromolecules which play crucial role in the biological funtions. These are comprised of amino acids chains. The structure and folding patterns of amino acids decide the functions and features of proteins. The knowledge of the functions of proteins play important role in identifying their biological processes, disorders, therapeutics and medicine and accordingly the measures to prevent the biological disorders can be taken. In the process of achieving the goals, related to proteins and their processes the classification of proteins is vital. This research helps in demonstrating the relationship between protein sequences and its related classification type. The objective is achieved by applying 1D convolutional neural network model on a dataset comprising of more than 3 lakh tuples. The results of above method is compared using learning rate charts. In the current work, researchers have focused on understanding the structural protein sequences dataset accessed from Protein Data Bank database available at https://www.rcsb.org/ website of Research Collaboratory for structural bioinformatics. Also after the detailed analyses of protein dataset, the classification of proteins has been demonstrated using deep learning approaches(convolutional neural network model).**

Keywords—Protein Classification, medicine, Deep Learning

I. Introduction

In recent times, biological data produced by advanced sequencing technologies have motivated the researchers to explore. A proteome is multi-dimensional structure with a complete set of proteins. This has several dedicated operations in maintaining the cellular structure and functionality of proteins. Any small change or error in the genetic sequence of the protein can cause changes in critical sections of proteins structure. This change or alteration impacts the phenotypes as it changes the tertiary structure of protein. A phenotype is a change in normal physiology or behavior. This can also be expressed by the alterations in the structure of protein or regulation of one or more proteins which play vital role in different unique functionalities such as growth and maintenance, causing biochemical reactions, acting as a messenger, providing structures and protection. For example, changes specific to Heart Disease or any cancer.

Research in understanding the structure of proteins, classification and clustering of proteins and changes in proteins which cause to phenotype changes to humans has become popular field of research with increase in biological data provided by advanced sequencing technologies. As per the functions of proteins in different processes classify these in several groups such as transport, enzyme, protection, storage, hormonal, structure and contractile [1]. With advancements in sequencing technologies, number of known proteins is also increasing. As an estimate, 176 million protein sequences are in Uniprot[2] database. The length of a protein may vary from 6 to 35000 amino acids[3].

II. Literature Review

To understand these proteins, classification can be a good measure. This can be done on the basis of similarities in functions and structure. Also another can be relationship with some common disease. Protein databases such as GCPR[4], SCOP[5] classify and store proteins hierarchically as classes, families, and subfamilies. COGs[6], Pfam[7], Uniprot are proteins sequence databases that provide both manually and automatically reviewed proteins using computational methods.

A. Machine Learning Approach

Machine learning Naïve Bias algorithms have been implemented by various researchers in recent times for classification of protein sequences. Feng P-M et. al [8]. have implemented Naïve Bayes Classifier. Researchers used a benchmark dataset of 307 sequences of Phage virus that uses bacteria as hosts. A Naïve Bayes classifier with feature selection was used to classify

Phage Virion proteins containing 99 phage virion protein sequences and 208 Phage non-virion protein sequences from the UniProt database. This method scored 79.6% accuracy over jacknife testing benchmarks.

Yanyuan Pan et al [9]. proposed a method to predict bacteriophage virion proteins using a Multinomial Naïve Bayes classification model based on discrete feature generated from the g-gap feature tree. For this, researchers created a web server (PhagePred) that implements the proposed predictor is available, which can be freely accessed on the Internet. Yuran Jia et al [10] developed a DNA-binding protein identification method called KKDBP. To improve prediction accuracy, we propose a feature extraction method that fuses\multiple PSSM features. The experimental results show a prediction accuracy on the independent test dataset PDB186 of 81.22%, which is the highest of all existing methods.

Yu-Wei Huang et al [11] used “merged moiety-based interpretable features (MMIFs),” which merged four moiety-based compound features as the input features for building random forest (RF) models. By using>200,000 bioactivity test data, researchers classified inhibitors as kinase family inhibitors or non-inhibitors in the machine learning. The results showed that our RF models achieved good accuracy (>0.8) for the 10 kinase families. Binh P. Nguyen et al[12] proposed a method that utilizes a convolutional neural network and Random Forest using the normalized PSSM and the best-selected feature of the ProtBert output M. Saifur Rahmana et al[13] have applied Support Vector Machine (SVM) to classify the sub-Golgi proteins. This method (isGPT), achieves accuracy values of 95.4%, 95.9% and 95.3% for 10-fold cross-validation test, jackknife test and independent test respectively.

B. Deep Learning Approach

Recently many researchers have implemented use of deep learning for protein classification [14],[15]. Deep learning methods have shown better performance when it is compared with the existing methods. DeepSF[16] constructs a convolutional neural network (CNN) with 30 layers to classify protein sequences.

Another model Deepre[17] has a slight change in approach which implements a combinations of a CNN with a recurrent neural network (RNNs). This extracts convolutional and sequential features from protein sequences to predict enzyme classifications. The model was tested using cross-fold validation and the experiments were conducted on two large-scale datasets of the uniProt database.

Szalkai B et al [18] have classified 521,527 protein sequences among 698 families of class labels with accuracy of 99.99% using the Uniprot dataset. ProtCNN[19] (a single deep convolutional neural network), is another work focussing on residual networks architecture in classifying the sequences of Pfam[7] dataset. DeepFam[20] is a deep learning model that is used for classifying proteins to their families.

III. DATASET ANALYSIS AND PREPROCESSING

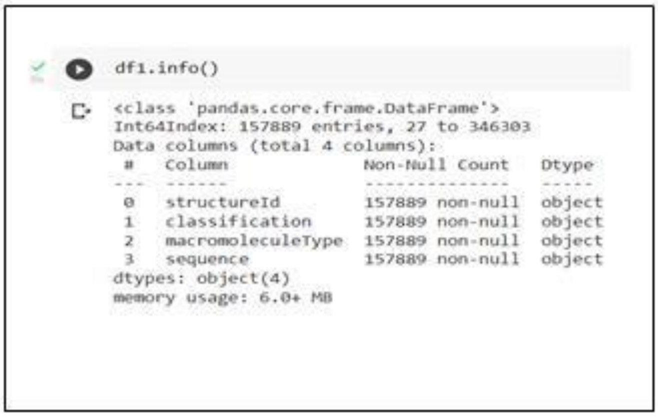
The dataset used here is from RCSB PDB Protein Data Bank[21] . RCSB PDB (RCSB.org) is a data center used to maintain global Protein Data Bank (PDB) archive in the format of 3D structure of large biological molecules such as (proteins, DNA, and RNA) with primary goal of research and education in field of fundamental biology, health, energy, and biotechnology being funded by United States of America.

The dataset contains 2 files with a common field *structuredId*. Using this field, researchers, merged both the files into a single file to visualize all the related fields altogether using inner join functionality of pandas. The fields identified relevant to this research are ***structureId, classification, macromoleculeType*** and ***sequence*** . The main task here was to classify all the proteins available into provided classification column’s values on the basis of sequence column.

To format the dataset as per the needs of the research, the in-depth analysis was done using exploratory data analysis techniques on top of statistical concepts. Few of the EDA outputs are given in the image. There are total of 435372 records of different macromolecule types. Out of which 345321 records are of type protein. These proteins are of different types having their own sequence and constituted using the 20 default amino acid codes. While doing the analysis no null values were found in the dataset. A total of 3017 unique proteins exist in the dataset ranging different value counts. To make computations less time consuming and hardware requirements friendly, only top 6 value counted proteins have been classified in this research. After this operation, only 157889 records were left for the main task. Also to make the dataset compatible for training, classification column’s values have been encoded while values in sequence column were tokenized to implement the classification models. Keras library provides the Tokenizer package for this task.

After preprocessing of data, only 6 major classes of protein remain in the dataset considering the hardware requirements. All sequences have been converted to vector representations using Tokenizer class of Keras library. This way, each word or part of text can be represented in some form of token built on top of some key value architecture being token as key or vector and value as text tokenized. Later a proper binding of these text sequences along with tokens generated are considered as X variable of set of independent variables for model training.

Fig. 1



```
df1.info()
```

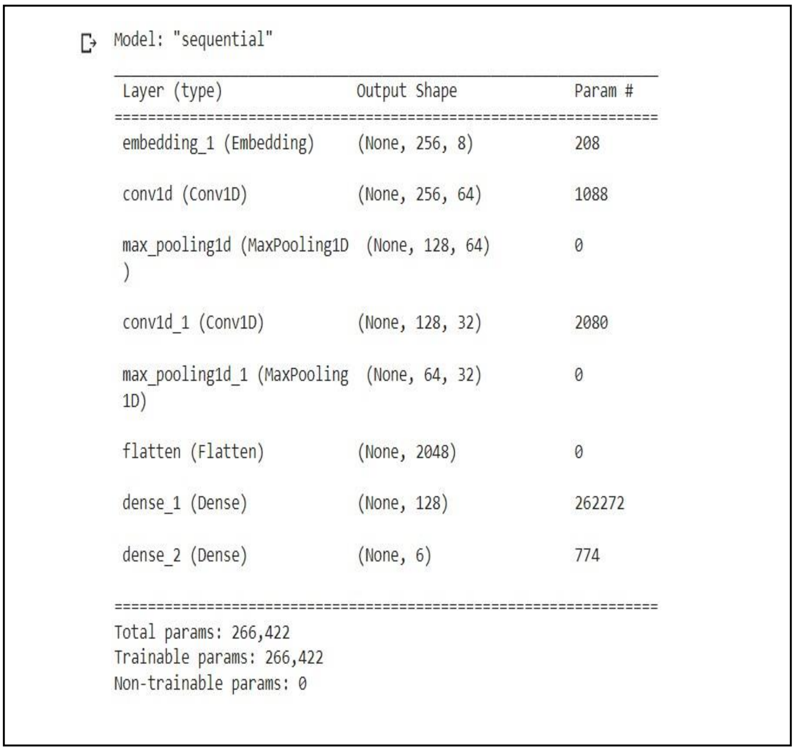
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 157889 entries, 27 to 346303
Data columns (total 4 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   structureId         157889 non-null  object
 1   classification      157889 non-null  object
 2   macromoleculeType   157889 non-null  object
 3   sequence            157889 non-null  object
dtypes: object(4)
memory usage: 6.0+ MB
```

IV. METHODOLOGY

After data preprocessing, To construct 1 dimensional convolutional neural network, architecture first implements an embedding layer that learns the vector representation for each code followed by first convolution of single dimension. In the next layer, Max pooling has been implemented reduce the computational cost and also provides basic translation invariance to the internal representation. After repeating again one 1D convolution and a softmax pooling, next layer is introduced for the purpose of flattening. The current state of vectors is in multi dimensions, so to convert this in a long one dimensional vector, a flatten layer is implemented.

Finally. In the last two steps, 2 dense layers have been implemented. This way, total trainable parameters become 262272 with 0 number of non-trainable parameters.

The output dense layer has 6 output shape as unique classes to the problem are only 6. Below is the model summery.



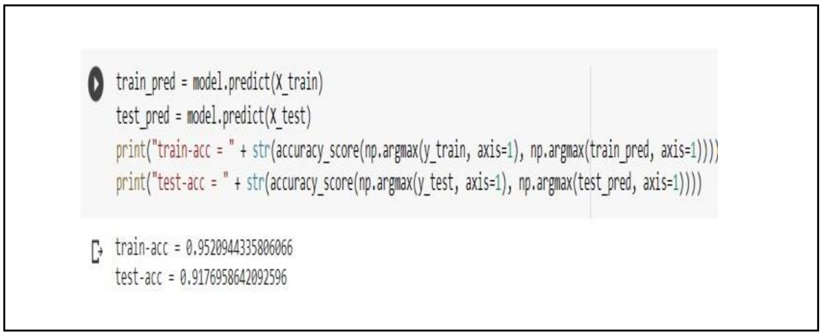
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 256, 8)            208

 conv1d (Conv1D)             (None, 256, 64)           1088

 max_pooling1d (MaxPooling1D  (None, 128, 64)          0
 )

 conv1d_1 (Conv1D)           (None, 128, 32)           2080

 max_pooling1d_1 (MaxPooling  (None, 64, 32)           0
 1D)

 flatten (Flatten)           (None, 2048)              0

 dense_1 (Dense)             (None, 128)               262272

 dense_2 (Dense)             (None, 6)                 774

=================================================================
Total params: 266,422
Trainable params: 266,422
Non-trainable params: 0
```

Fig 2

Before moving ahead with training the model, a train test split of the data has been done on the ratio of 8:2 i.e. 80% of the data in under training phase while 20% of the data under testing phase. The random state hyper-parameter controls the shuffling process of the dataset. This helps in randomness of the learning and validation data. Keras documentation states that number of epochs beyond 11, loss on training set start decreasing and becomes almost zero along-with increment but loss on validation increases as it considers overfitting. To avoid such a situation, this model sets epochs as 10. Also a batch-size of 128 has been set.

RESULTS

In the current study, researchers have tried to classify the protein sequences on the basis of their respective protein type. After going through the past work and some analysis of the statistical theorems, the proposed work implements a convolutional neural network model. Model learning accuracy on train data is 95% while on test data it performs well with an accuracy of 91%. Output of the is attached in image.



```
train_pred = model.predict(X_train)
test_pred = model.predict(X_test)
print("train-acc = " + str(accuracy_score(np.argmax(y_train, axis=1), np.argmax(train_pred, axis=1))))
print("test-acc = " + str(accuracy_score(np.argmax(y_test, axis=1), np.argmax(test_pred, axis=1))))
```

```
train-acc = 0.9520944335806066
test-acc = 0.9176958642092596
```

Fig 3

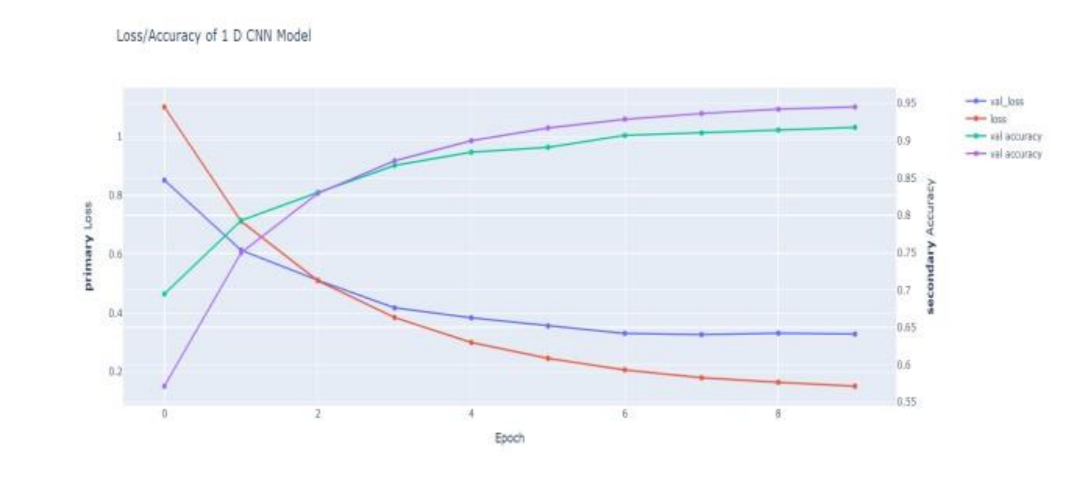


Fig 4

The image here represents how the model grew while learning along-with accuracy score of training and test data.

References

[1] J. R. Bradford, J. A. Siepen, and D. R. Westhead, “Fundamentals of protein structure and function,” *Encycl. Genet. Genomics, Proteomics Bioinforma.*, 2005, doi: 10.1002/047001153x.g307214.

[2] A. Bateman *et al.*, “UniProt: The universal protein knowledgebase,” *Nucleic Acids Res.*, vol. 45, no. D1, pp. D158–D169, 2017, doi: 10.1093/nar/gkw1099.

[3] M. Levitt, “Nature of the protein universe,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 106, no. 27, pp. 11079–11084, 2009, doi: 10.1073/pnas.0905029106.

[4] A. J. Kooistra *et al.*, “GPCRdb in 2021: Integrating GPCR sequence, structure and function,” *Nucleic Acids Res.*, vol. 49, no. D1, pp. D335–D343, 2021, doi: 10.1093/nar/gkaa1080.

[5] A. Andreeva, E. Kulesha, J. Gough, and A. G. Murzin, “The SCOP database in 2020: Expanded classification of representative family and superfamily domains of known protein structures,” *Nucleic Acids Res.*, vol. 48, no. D1, pp. D376–D382, 2020, doi: 10.1093/nar/gkz1064.

[6] M. Y. Galperin, Y. I. Wolf, K. S. Makarova, R. V. Alvarez, D. Landsman, and E. V. Koonin, “COG database update: Focus on microbial diversity, model organisms, and widespread pathogens,” *Nucleic Acids Res.*, vol. 49, no. D1, pp. D274–D281, 2021, doi: 10.1093/nar/gkaa1018.

[7] J. Mistry *et al.*, “Pfam: The protein families database in 2021,” *Nucleic Acids Res.*, vol. 49, no. D1, pp. D412–D419, 2021, doi: 10.1093/nar/gkaa913.

[8] P. M. Feng, H. Ding, W. Chen, and H. Lin, “Naïve bayes classifier with feature selection to identify phage virion proteins,” *Comput. Math. Methods Med.*, vol. 2013, 2013, doi: 10.1155/2013/530696.

[9] Y. Pan, H. Gao, H. Lin, Z. Liu, L. Tang, and S. Li, “Identification of bacteriophage virion proteins using multinomial Naïve bayes with g-gap feature tree,” *International Journal of Molecular Sciences*, vol. 19, no. 6. 2018, doi: 10.3390/ijms19061779.

[10] Y. Jia, S. Huang, and T. Zhang, “KK-DBP: A Multi-Feature Fusion Method for DNA-Binding Protein Identification Based on Random Forest,” *Front. Genet.*, vol. 12, no. November, pp. 1–9, 2021, doi: 10.3389/fgene.2021.811158.

[11] Y. W. Huang *et al.*, “Discovery of moiety preference by Shapley value in protein kinase family using random forest models,” *BMC Bioinformatics*, vol. 23, pp. 1–13, 2022, doi: 10.1186/s12859-022-04663-5.

[12] B. P. Nguyen, Q. H. Nguyen, G. N. Doan-Ngoc, T. H. Nguyen-Vo, and S. Rahardja, “IProDNA-CapsNet: Identifying protein-DNA binding residues using capsule neural networks,” *BMC Bioinformatics*, vol. 20, no. Suppl 23, pp. 1–12, 2019, doi: 10.1186/s12859-019-3295-2.

[13] M. S. Rahman, M. K. Rahman, M. Kaykobad, and M. S. Rahman, "isGPT: An optimized model to identify sub-Golgi protein types using SVM and Random Forest based feature selection," *Artif. Intell. Med.*, vol. 84, pp. 90–100, 2018, doi: 10.1016/j.artmed.2017.11.003.

[14] S. Min, B. Lee, and S. Yoon, "Deep learning in bioinformatics," *Brief. Bioinform.*, vol. 18, no. 5, pp. 851–869, 2017, doi: 10.1093/bib/bbw068.

[15] Estefan, "Advanced Techniques in Biology & Medicine," *Adv Tech Biol Med*, vol. 3, no. 3, pp. 10–11, 2015, doi: 10.4172/2379-1764.

[16] J. Hou, B. Adhikari, and J. Cheng, "DeepSF: Deep convolutional neural network for mapping protein sequences to folds," *Bioinformatics*, vol. 34, no. 8, pp. 1295–1303, 2018, doi: 10.1093/bioinformatics/btx780.

[17] Y. Li *et al.*, "DEEPre: Sequence-based enzyme EC number prediction by deep learning," *Bioinformatics*, vol. 34, no. 5, pp. 760–769, 2018, doi: 10.1093/bioinformatics/btx680.

[18] B. Szalkai and V. Grolmusz, "Near perfect protein multi-label classification with deep neural networks," *Methods*, vol. 132, pp. 50–56, 2018, doi: 10.1016/j.ymeth.2017.06.034.

[19] M. L. Bileschi *et al.*, "Using deep learning to annotate the protein universe," *Nat. Biotechnol.*, vol. 40, no. 6, pp. 932–937, 2022, doi: 10.1038/s41587-021-01179-w.

[20] S. Seo, M. Oh, Y. Park, and S. Kim, "DeepFam: Deep learning based alignment-free method for protein family modeling and prediction," *Bioinformatics*, vol. 34, no. 13, pp. i254–i262, 2018, doi: 10.1093/bioinformatics/bty275.

[21] S. K. Burley *et al.*, "RCSB Protein Data Bank: Biological macromolecular structures enabling research and education in fundamental biology, biomedicine, biotechnology and energy," *Nucleic Acids Res.*, vol. 47, no. D1, pp. D464– D474, 2019, doi: 10.1093/nar/gky1004.