# Protein Domain Classification: A Compact CNN Framework.

Samiha Afaf Neha
*Department of CSE.*
*Brac University*
Dhaka, Bangladesh
samiha.afaf.neha@g.bracu.ac.bd

***Abstract*—Proteins are vital macro-molecules responsible for the structural and functional mechanisms of living things. In scientific and medical research it is important to properly classify the newly identified or synthesized protein sequence into their respective families in order to obtain functional inferences which can then be used for formulating new therapies, enzymes, small-molecules etc. With the onset of technological advances, newly found amino acid sequences are being obtainable, heightening the need for efficient and reliable family classification. Previously, several machine learning algorithms have been employed for inferring the classes requiring expert level feature extraction. In response to this dilemma, deep learning approaches have been proposed to automatically learn the hidden sequence feature and predict accordingly. This report proposes such a convolutional deep learning model, which attempts to categorize 1-dimentional short chain amino acids, reaching a satisfying speculation rate.**

***Index Terms*—deep learning, protein classification, CNN**

## I. Introduction

In proteomic and genomic research, researchers constantly strive to unravel the inner, hidden patterns of biological molecules in order to understand the structure that results in a particular function. Protein sequence classification is a classic problem in this domain, where the amino acid residues in a protein are analyzed for the task of categorization. Understanding which elements in the chain causes which functionality can lead to synthesis of better drugs, protein-binders etc. As in recent times, the growth of protein data bank has increased along with the instances of unidentified sequences. More than 40% of the protein sequences of NCBI database has been uncategorized as of 2013 [1]. The trajectory of publishing year as shown in Fig 1. of the sequences in our dataset further confirms the situation.

In order to propose predictions, classical machine learning models such as Random Forest, SVM, and Naive Bayes, have been used but have met with limited accuracy [2] due to the major drawback of a lack of explicit features [3]. This led to the incorporation of deep learning methods which yielded better accuracy as the model learned the hidden features automatically as it trained, without requiring any subject knowledge. Comparison between RNN, LSTM and CNN models showed the superiority of the CNN due to its' massive success in the NLP related mining tasks [4].In light of these findings, a compact CNN model is proposed in this paper which attempts to extract information from raw protein sequences using with a competent accuracy using less parameters and smaller sequences.

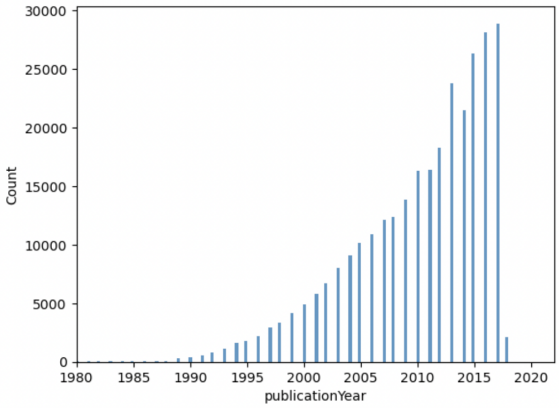


Fig. 1. Sequence publishing years

## II. Related Works

Unlike genomics medicine and the image research field, less work has been done on the sequence analysis problem, due to the hidden feature problem of the residue chain. Related work in the field exercises deep learning approaches, meeting with state-of-the art results where the models learn features by themselves in each epoch of the training process.

The article "Protein classification from scratch using 1D-CNN" deals with the classification task by employing a customized 1D CNN model that runs on sequences of length 50 to 1200 amino acids taken from the UniProt dataset. It used encoding, embedding, CNN, and a fully connected module. Their model has reached an accuracy of 97%, which is significantly greater than other machine learning models. The model incorporates max pooling, dropout, and activation functions such as ReLu and SoftMax for the task. Along with the training, hyperparameters such as batch size, number of epochs, dropout, etc. were tuned while checking accuracy metrics such as F1, recall, and precision scores. One similar sequencing problem was also researched in the paper **"Deep**

**Learning Architectures for DNA Sequence Classification,"** where DNA sequences were up for classification from the 16S data-set. The model has the similar encoding, embedding steps and uses tanh, softmax activation functions along with cross-validation for the training. It compares the LSTM and CNN for a total of 5 bacteria classification tasks, where CNN outperforms the LSTM 4 out of 5 times [5]. It also concludes that multi-tasking approach on a model works better for LSTM but worsens CNN performance.

A review of the deep learning approach for protein classification stipulates that CNN algorithms are capable of extracting distinctive geographical information [6] from the sequence data while reducing data size by amplification and representing local features in the final output. Modern sequence matching tools such as **BLAST** and **FASTA** use heuristic algorithms for sequence matching against already identified protein sequences and give results based on sequence or subsequence similarity scores. Studies have shown that sequence matching is not the most accurate matrix for prediction, as highly identical protein sequences are not always expected to have the same gene ontology. Comparison between RNN, Multimodal architectures and CNN shows that 45% of the time CNN alone is used for such sequence classification tasks.

In protein domain classification from third generation sequencing reads, deep learning techniques, especially CNN, are used, as the convolutional filters are capable of representing motifs that are essential for sequence classification. The paper [7], **"ProDOMA: improving PROtein DOMAin classification for third-generation sequencing reads using deep learning,"** studies this concept and designs a CNN model named **"ProDOMA"** that outperforms the state-of-the-art HMMER and DeepFam models for domain classification. It mainly works well for long, noisy reads without depending on error correction. ProDOMA uses a convolutional layer and a max-over-time pooling layer to automatically extract features from the input. The probability of the sequence against all of the input protein domains was calculated using a classifier with two fully connected layers. The CNN was trained with a modified loss function to make out-of-distribution samples more likely to have a uniform distribution on softmax values, thereby excluding unrelated coding or non-coding DNA sequences. Precision was evaluated based on F1 scores and run time was measured by averaging 5 independent trials with 10,000 random sequences.

## III. Methodology

The methodology of our project incorporates stages of date preprocessing, data encoding and embedding and finally formulation of a deep neural network architecture for the task at hand. Each of these sections are briefly described below.

### A. Dataset and preprocessing

Datasets used for the classification task was obtained from Kaggle, which were curated from TheProteinBank online database. One of the set contained sequences of various biological macro molecules e.g proteins, DNA, RNA while the other had corresponding classification labels. Both the datasets are joined together based on a common identification column, named **structureId**. After merging, entries of molecules other than Proteins are dropped and the index of the data frame is reset to default for ease of data evaluation. The data set contains more than 4000 protein classification labels thus in order to mitigate heavy computational complexity, top 10 most frequent classes are selected for model training while the rest are dropped. As shown in the Fig. 2 the selected classes vary quite a bit in their occurrences, giving us guideline over the evaluation matrices that might work best for the final data.

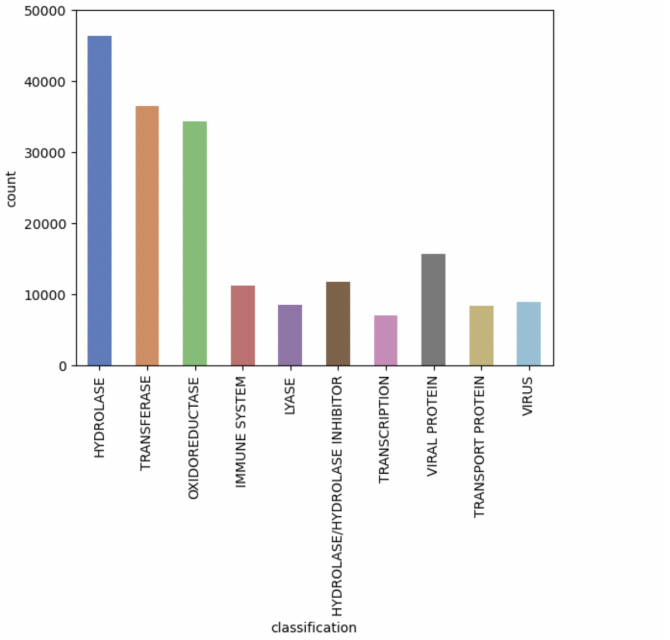


Fig. 2. Frequency map of top 10 selected protein family

### B. Data Encoding and Embedding

After initial data processing, the categorical labels of the dataset are engineered into numerical values using one-hot encoding with the help of LabelBinarizer module of the Scikit-learn software, each entry representing an array of length 10. Each array only has one index set to 1 for the corresponding class name as shown in Fig. 3.

```
array([[1, 0, 0, ..., 0, 0, 0],
       [1, 0, 0, ..., 0, 0, 0],
       [1, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 1, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]])
```

Fig. 3. Classification encoding using one-hot

The target feature i.e the protein sequences of the data set being non-numerical are also transformed into tokens using

the Keras Tokenizer. The sequences are randomly encoded on a character level, each letter presenting a single token. As the length of the sequences are extensive (see Fig. 3), we have chosen the maximum string length to be 250 characters long and the rest of the sequence is padded.This turns down the computational time to a reasonable duration.

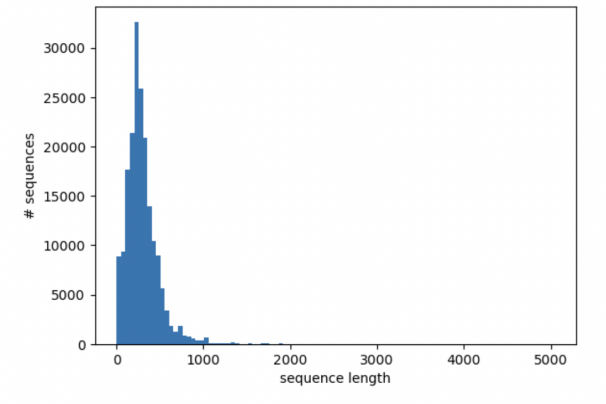


Fig. 4. Histogram representing the sequence length of proteins

### C. Deep learning architecture

The deep learning architecture incorporates three distinct units (see Fig. 5). First comes the embedding unit that takes each tokenized letter from the sequence and converts it into an n-dimensional vector, which is then transmitted into the next CNN module. The CNN module is employed for the 1D convolution along the protein sequences, thus extracting the hidden features along the way. A fully connected module follows the CNN module which finishes the classification part of the architecture.

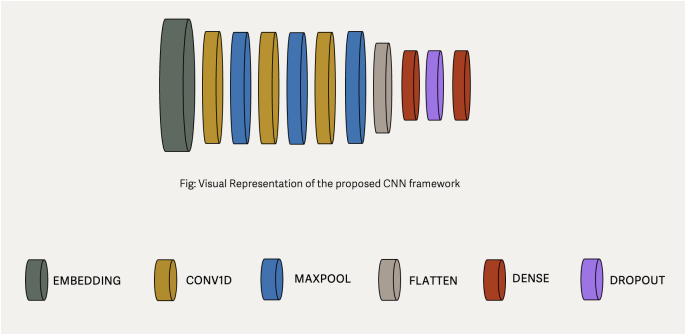


Fig. 5. Proposed Deep Learning model

There are in total 3 convolutional layers each followed by a MaxPool layer. Filter size of 64, 32, 64 with kernel size (5,3,3) was used. The output of the convolution is flattened and fed to 2 consecutive dense layers with activation functions 'relu' and 'softmax' respectively. A small dropout of 10% was used for preventing training overfitting.

## IV. Results

### A. Evaluation matrix

The evaluation metrics used in our method includes the widely used Accuracy, F-1 score and two other infrequently used metrics - Cohan's Kappa (K) and Matthews Correlation Coefficient (MCC), both for multiclass classification. The definitions of the metrics are given in the equations below.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$F1 - score = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

$$MCC = \frac{c \times s - \sum_k^K p_k \times t_k}{\sqrt{(s^2 - \sum_k^K p_k^2)(s^2 - \sum_k^K t_k^2)}} \tag{3}$$

$$K = \frac{c \times s - \sum_k^K p_k \times t_k}{s^2 - \sum_k^K p_k \times t_k} \tag{4}$$

### B. Experimental Setup

The dataset as mentioned earlier was taken from the online TheProteinBank database having more than 3000 classifications and sequence lengths of upto 2500 residues long. For subsidising severe computational complexity, top 10 frequently occuring classes were kept for running in the final model. Before inputting the data into the training phase, the entries were shuffled in order to eliminate any data pattern. Finally, a test split of 10% and a validation split of 10% was taken for optimum training.

Hyperparameters such as CNN filters, kernel sizes, pool sizes, and activation functions were tuned after several test runs of the model. Some of these are shown in Table I. below.

TABLE I
**TUNED HYPERPARAMETERS**

| | |
|---|---|
| **Dropout** | **0.1**, 0.5, N/A |
| **No. of Epochs** | 15, **20**, 30 |
| **Batch size** | 64, **128**, 256 |
| **Amino acid embedding** | 5, **8**, 10 |
| **Optimizer** | Adam |

### C. Evaluation and model comparison

The model proposed proves the notion that a protein class can be predicted with relatively high accuracy only using the provided sequence, without requiring any motif deduction or field expert supervised data encoding. Our model although being a small scale one, does a good job at predicting protein classes with test accuracy of 92.8%, MCC and K score of 94.12% and 91.6% respectively for 10% validation and test split. F-1 and recall were both 93%. The model trains in a decently stable pattern as shown in Fig. 6 and Fig. 7. Keeping all the variable unchanged and taking a 20% validation split gives higher accuracy of 95.4% and MCC score of 91.2% but a lower MCC score of 93.4%, this model is slightly unstable despite the better results.

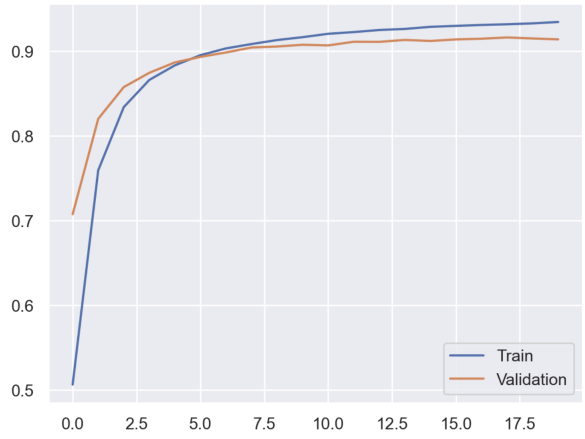


Fig. 6. Training and validation accuracy

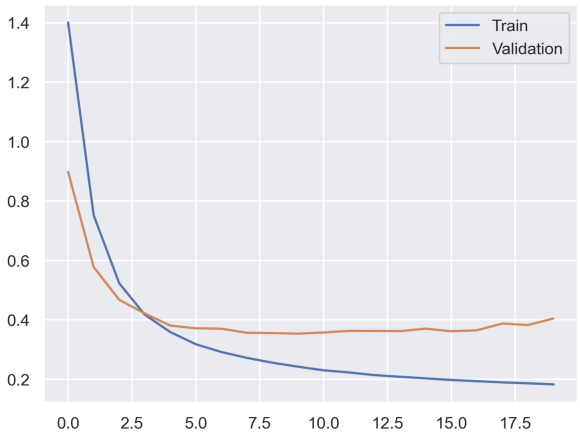


Fig. 7. Training and validation loss

Regardless of the compactness of our model, it still performs slightly well relative to other deep learning models used in this field of research as shown in TABLE II.

TABLE II
PREDICTION PERFORMANCE COMPARISON FOR DIFFERENCT MODELS

| Architecture | Dropuot | Test Accuracy |
|---|---|---|
| biLSTM [8] | 0.7 | 92.2% |
| LSTM [8] | 0.85 | 92.5% |
| SVM [8] | NA | 87.8% |
| Our Model(Stable) | 0.1 | 92.8% |
| Our Model(Unstable) | 0.1 | 95.4% |
| CNN (10 layers) [9] | 0.6 | 97.8% |

On the contrary, the simplicity of our model do pose limitations while trying to achieve up to the minute results as compared to the state-of-the-art CNN model employed for this task, reaching 97.7% test accuracy. As mentioned in the related work section, this model uses a 10 layer stack, and dense inferences nodes, convolution filters of much higher order compared to our model, which only uses half the number of layers.

## V. Conclusion and Future Work

In this study, a concise CNN model has been experimentally trained for the task of protein classification with only 5 layers and sequence length of 250 residues. Although the model performs quite well even with the size limitation, it fails to acheive a state-of-the-art result which is a greater concern. The task can be extended in many ways, one using NLP approaches where proteins are considered as languages [10] while each residue is a word. Attention based classification has not been used much in this frield [11] compared to other techniques and might give satisfying results. Another approach of translating 1-D sequence into 2-D pairplot for classification to analyze the temporal raw sequneces [12] might also be a feasible approach towards better results in classification.

## References

[1] Aggarwal, D., Hasija, Y. (2022). A Review of Deep Learning Techniques for Protein Function Prediction. arXiv preprint arXiv:2211.09705.
[2] Zhang, D., Kabuka, M. R. (2020). Protein family classification from scratch: A cnn based deep learning approach. IEEE/ACM transactions on computational biology and bioinformatics, 18(5), 1996-2007.
[3] Lo Bosco, G., Di Gangi, M. A. (2017). Deep learning architectures for DNA sequence classification. In International Workshop on Fuzzy Logic and Applications (pp. 162-171). Springer, Cham.
[4] Lo Bosco, G., Di Gangi, M. A. (2017). Deep learning architectures for DNA sequence classification. In International Workshop on Fuzzy Logic and Applications (pp. 162-171). Springer, Cham.
[5] Lo Bosco, G., Di Gangi, M. A. (2016, December). Deep learning architectures for DNA sequence classification. In International Workshop on Fuzzy Logic and Applications (pp. 162-171). Springer, Cham.
[6] Aggarwal, D., Hasija, Y. (2022). A Review of Deep Learning Techniques for Protein Function Predic- tion. arXiv preprint arXiv:2211.09705.
[7] Du, N., Shang, J., Sun, Y. (2021). Improving protein domain classification for third-generation sequencing reads using deep learning. BMC genomics, 22(1), 1-13.
[8] Lee, Timothy K., and Tuan Nguyen. "Protein family classification with neural networks." (2016).
[9] Zhang, D., Kabuka, M. R. (2020). Protein family classification from scratch: A cnn based deep learning approach. IEEE/ACM transactions on computational biology and bioinformatics, 18(5), 1996-2007.
[10] Ofer, D., Brandes, N., Linial, M. (2021). The language of proteins: NLP, machine learning protein sequences. Computational and Structural Biotechnology Journal, 19, 1750-1758.
[11] Aggarwal, D., Hasija, Y. (2022). A Review of Deep Learning Techniques for Protein Function Prediction. arXiv preprint arXiv:2211.09705.
[12] Hatami, N., Gavet, Y., Debayle, J. (2018, April). Classification of time-series images using deep convolutional neural networks. In Tenth international conference on machine vision (ICMV 2017) (Vol. 10696, pp. 242-249). SPIE.