EMPIRICAL COMPARISON OF LSTM AND CNN ALGORITHM FOR PROTEIN HOMOLOGY DETECTION

Menaga K
PG Student
Department of Computer Science
Bharathiar University, Coimbatore.
Menagakumarappan1902@gmail.com

Dr.D.Ramyachitra
Assistant Professor
Department of Computer Science
Bharathiar University, Coimbatore.
Jaichitra1@yahoo.co.in

Sajithra N
Ph.D Research Scholar
Department of Computer Science
Bharathiar University, Coimbatore.
sajithramidhun@gmail.com

Anusree V A
PG Student
Department of Computer Science
Bharathiar University, Coimbatore.
anusree1127@gmail.com

Abstract— **In computational biology, protein remote homology detection is the classification of proteins into structural and functional classes given their amino acid sequences, especially, with low sequence identities. Protein homology discovery plays an important part within the field of bioinformatics since homologous proteins consume analogous structures and functions, which is critical for the studies of protein 3D structure and function. Proteins affiliate with each one to play critical places in numerous of those processes that are vital for an organism to live, and that shape its capacities for community with its terrain in cells. Although promising, laboratory experiments generally suffer from the disadvantages of being hamstrung and laborious. The results attained are frequently not robust and doubtful. Protein fold recognition is a crucial problem in structural bioinformatics. Nearly all traditional fold recognition styles use sequence (homology) comparison to laterally prognosticate the pack of a target protein grounded on the pack of a template protein with given structure, which may not explain the connection between sequence and fold. Only a many style had been developed to classify protein sequences into a small number of crowds due to methodological limitations, which aren't generally useful in practice. In this paper, we used classification algorithms namely CNN (Convolution Neural Network) and LSTM(Long Short-Term Memory). By comparing the both algorithm results CNN predicts the best results.**

Keywords — *Protein homology detection, Bioinformatics, CNN, LSTM.*

I. INTRODUCTION

Bioinformatics is an emerging discipline that draws upon the strengths of computer sciences, mathematics, and information technology to determine and analyses genetic information. Bioinformatics leverage synergies between computational and biological sciences. Although the field of bioinformatics originally aimed at extracting information embedded within the 3 billion bases of human DNA, the field has evolved to realize its capabilities for studying information.

content and information flow in biological systems and processes in general.

Bioinformatics involves the manipulation, searching and data mining of DNA sequence data. The development of techniques to store and search DNA sequences have led to widely applied advances in computer science, especially string searching algorithms, machine learning and database theory. In other applications such as text editors, even simple algorithms for this problem usually suffice, but DNA sequences because these algorithms to exhibit near-worst case behaviour due to their small number of distinct characters.

Protein remote homology discovery is one of the most abecedarian and central problems for the studies of protein structures and functions, aiming to descry the distantly evolutionary connections among proteins via computational styles. During the once decades, numerous computational approaches have been proposed to break this important task. These styles have made a substantial donation to protein remote homology discovery.

II. PROTEIN SEQUENCE AND STRUCTURE

One of the main disquisition problems in structural bioinformatics is the prophecy of three dimensional protein structures. Proteins are long sequences formed out of different amino acid remainders that in physiological conditions adopt a unique 3-D structure. Knowledge of the protein structure allows the exploration of natural processes more directly, with advanced resolution and finer detail. Protein Structure Prophecy (PSP) styles induce models of proteins (3D equals of Tittles) predicated on its amino acid composition. PSP remains, after several decades of disquisition, one of the main open problems in biology. Several (constantly complementary) ways and representations for PSP live, predicated on different sources of information and a wide variety of prophecy and model refinement styles. In general these ways bear vast amounts of computational resources.

Nevertheless, presently it has been recognized that not all protein functions are associated to a folded state. Three-dimensional protein structure prophecy has been proposed as a affect to the 3D Protein Structure Prophecy (3D-PSP) problem. These styles can be divided as first principle styles without database information; first principle styles with database information; fold recognition and threading styles; and relative modeling styles and sequence alignment strategies. Deterministic computational ways, optimization ways, data mining and machine knowledge approaches are generally used in the construction of computational results for the PSP problem.

The algorithmic comparison of protein structures is a vital and truly challenging task in bioinformatics. Protein Structure Comparison styles can be used to make consequences on protein function or to distinguish between near-native models and baits in protein structure Vaticination and protein design. Also, what makes this task really challenging is that there is no unique "tableware bullet" measure of similarity that is suitable for all tasks/ datasets. In this area developed both individual algorithms for structural comparison as well as agreement styles integrating multiple structural similarity measures.

Protein sequencing is the practical process of determining the amino acid sequence of all or part of a protein or peptide. This may serve to identify the protein or characterize its posttranslational variations. Generally, partial sequencing of a protein provides sufficient information (one or farther sequence labels) to identify it with reference to databases of protein sequences Derived from the abstract paraphrase of genes. The two major direct styles of protein sequencing are mass spectrometry and Edman declination using a protein sequenator (sequencer). Mass spectrometry styles are now the most considerably used for protein sequencing and identification but Edman declination remains a precious tool for characterizing a protein's N- boundary.

III. METHODOLOGY

There are two algorithms called CNN and LSTM have been used to find the Result of the scop and pfam protein dataset.

a. *PROTEIN HOMOLOGY DETECTION USING CNN*

CNN is a common deep- literacy fashion that can yield slice- edge results for utmost bracket problems. CNN performs well not only on image bracket, but it can also produce good delicacy on textbook data. Substantially, CNN is used to automatically prize the features from the input dataset, in discrepancy to machine literacy models, where the stoner needs to elect the features 2D CNN, and 3D CNN is used for image and videotape data, independently, whereas 1D CNN is used for textbook bracket. Since CNN can work only with numerical data, the DNA sequence is converted into numerical values by applying one hot encoding. CNN armature uses a series of convolutional layers to prize features from the input dataset. Max pooling subcaste after each convolutional subcaste and the confines of uprooted features are reduced. In the convolutional subcaste, the size of the kernel plays a significant part in function birth. The model's hyperparameters are the number of pollutants and kernel size. The softmax function is used as the bracket subcaste, which can perform well for the multiclass problem. It is mainly used to normalize neural networks output to fit between zero and one. It is used to represent the certainty "probability" in the network output. The softmax subcaste consists of units, where the is the number of units. Each unit is completely connected with porous subcaste and computes the probability of each class by means of Equation

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{1}$$

1. INPUT LAYER

1D Convolutional Neural Networks are similar to well-known and more established 2D Convolutional Neural Networks. 1D Convolutional Neural Networks are used mainly used on text and 1D signals.

2. EMBEDDED LAYER

Embedding Layer represents the density of the word vector, unlike what we have done with the Countvectorizer. It is a different way to preprocess the data. This embedding can map semantically similar words. It does not consider the text as a human language but maps the structure of sets of words used in the corpus. They aim to map words into a geometric space which is called an embedding space.

If embedding finds a good relationship between works like for an example

King – man + women = queen

Keras provides a couple of methods for text preprocessing and sequence preprocessing. We can use them to make our data a better fit for the Text CNN model.

3. MAXPOOLING1D

The major benefit of using this kind of pooling operation is that the number of parameters is independent on the length of the document. Only with this change it is possible to obtain more or less the same results in 30s per epoch (probably you require an epoch or two more) on CPU (previously it was >100s. For the fact that the performance on both memory and time one can set maxlen =400 resulting in a boost of performance of ~7 points of accuracy after some epochs (with other small modifications it is possible to easily obtain 90 accuracy, i.e. early stopping after some epochs).

4. FLATTEN LAYER

The last stage of a convolutional neural network (CNN) is a classifier. It is called a dense layer, which is just an artificial neural network (ANN) classifier. And an ANN classifier needs individual features, just like any other classifier. This means it needs a feature vector. Therefore, you need to convert the output of the convolutional part of the CNN into a 1D feature vector, to be used by the ANN part of it. This operation is called flattening. It gets the output of the convolutional layers, flattens all its structure to create a single long feature vector to be used by the dense layer for the final classification.

5. DENSE LAYER

Dense layer is the regular deeply connected neural network layer. It is most common and frequently used layer. Dense layer does the below operation on the input and return the output.

The figure1 is the flow chart of CNN Algorithm, it works in the form of if loop until reaching the valid result.

In the input layer we will insert the sequence and in the embedding the given sequence will be encoded to the numerical form of value 0-19. Conv1D is used to creates a convolution kernel that is convolved with the layer input over a single spatial (or temporal) dimension to produce a tensor of outputs the same process will done in each and every Conv1D. maxpooling1D is used to represent input by taking the maximum value of the window size pool_size. Merge is used to merge the list of inputs. Flatten is used to consider the row by row values of input and insert it into the one long column. Dense is a simple layer of neurons in which each neuron receives input from all the neurons of previous layer.

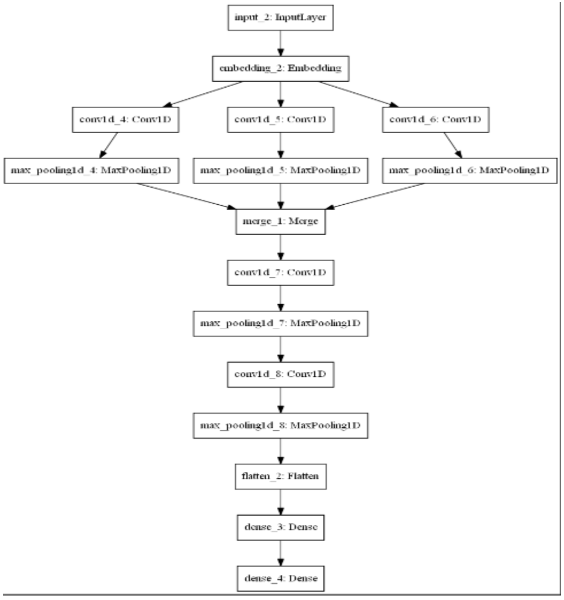
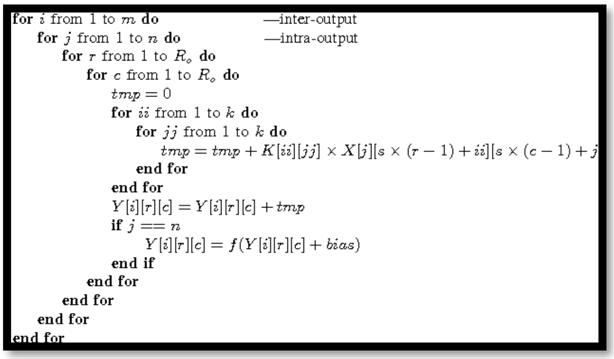


Fig. 1. Work Flow of CNN Algorithm

The figure1 is the flow chart of CNN Algorithm, it works in the form of if loop until reaching the valid result. In the input layer we will insert the sequence and in the embedding the given sequence will be encoded to the numerical form of value 0-19. Conv1D is used to creates a convolution kernel that is convolved with the layer input over a single spatial (or temporal) dimension to produce a tensor of outputs the same process will done in each and every Conv1D. maxpooling1D is used to represent input by taking the maximum value of the window size pool_size. Merge is used to merge the list of inputs. Flatten is used to consider the row by row values of input and insert it into the one long column. Dense is a simple layer of neurons in which each neuron receives input from all the neurons of previous layer.

PSEUDOCODE FOR CNN ALGORITHM



```
for $i$ from 1 to $m$ do                    —inter-output
    for $j$ from 1 to $n$ do                —intra-output
        for $r$ from 1 to $R_o$ do
            for $c$ from 1 to $R_o$ do
                $tmp = 0$
                for $ii$ from 1 to $k$ do
                    for $jj$ from 1 to $k$ do
                        $tmp = tmp + K[ii][jj] \times X[j][s \times (r-1) + ii][s \times (c-1) + j$
                    end for
                end for
                $Y[i][r][c] = Y[i][r][c] + tmp$
                if $j == n$
                    $Y[i][r][c] = f(Y[i][r][c] + bias)$
                end if
            end for
        end for
    end for
end for
```

b. PROTEIN HOMOLOGY DETECTION USING LSTM

Variations of Intermittent neural networks like LSTMs are the first choice when working on NLP grounded problems as they were made to work with temporal or successional data like textbook. RNNs are a type of Neural Network where the affair from former way are fed as input to the current step, therefore remembers some information about the sequence. RNNs are great when it comes to short surrounds, but it has a limitation of remembering longer sequences because of evaporating grade problem. LSTM (Long Short- Term Memory) Networks are bettered performances of RNN, specialized in remembering information for an extended period using a gating medium which makes them picky in what former information to be remembered, what to forget and how important current input is to be added for erecting the current cell state.

This is made up of two reversed unidirectional LSTM. To handle the long mock protein sequences, and better internee the reliance information of subsequences, we tap into all of the intermediate retired values generated by bidirectional LSTM. The retired values generated by the forward LSTM and backward LSTM for the same input subsequence are concatenated into a vector, which is shown in Eq (2).

ht = (hft, hbt) (2)

where h is retired value, f represents the forward LSTM, b represents the backward LSTM, t means the t th time step.

In the bidirectional LSTM caste, the mock protein is reused N- boundary to C- boundary and C- boundary to N- boundary simultaneously. Therefore, hft contains dependences between the target subsequence and its left neighbouring subsequence. hbt contains dependences between the target subsequence and its right neighbouring subsequence. These two reliance connections are concatenated into one vector h t, which can be interpreted as the point of the target subsequence. Therefore, farther comprehensive dependences can be included into the intermediate retired values by using bidirectional LSTM.

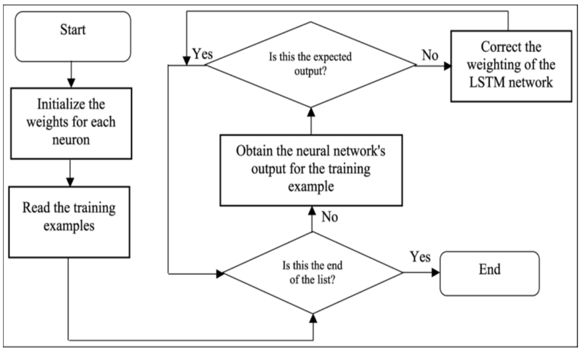
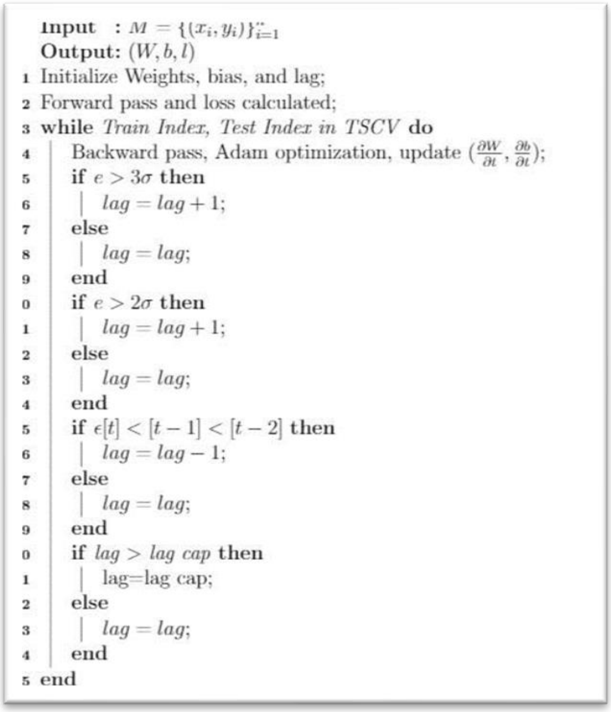

Start
Initialize the weights for each neuron
Read the training examples
Is this the expected output?
Yes
No
Correct the weighting of the LSTM network
Obtain the neural network's output for the training example
No
Is this the end of the list?
Yes
End

Fig 2. Work Flow of LSTM Algorithm

The Figure2 explains about the work process of the LSTM algorithm. First, we want to start the process after that we want to initialize the value to the sequences and we need to compare it with the training sample, if it was the sequence list it will reaches the end. Otherwise, we need to obtain neural network output for training sample if we got the expected output, it will reach the end. If not, we need to correct the weight of the LSTM network.

PSEUDOCODE FOR LSTM ALGORITHM



```
Input  : $M = \{(x_i, y_i)\}_{i=1}^{n}$
Output: $(W, b, l)$
Initialize Weights, bias, and lag;
Forward pass and loss calculated;
while Train Index, Test Index in TSCV do
    Backward pass, Adam optimization, update $(\frac{\partial W}{\partial t}, \frac{\partial b}{\partial t})$;
    if $e > 3\sigma$ then
        $lag = lag + 1$;
    else
        $lag = lag$;
    end
    if $e > 2\sigma$ then
        $lag = lag + 1$;
    else
        $lag = lag$;
    end
    if $\epsilon[t] < [t-1] < [t-2]$ then
        $lag = lag - 1$;
    else
        $lag = lag$;
    end
    if $lag > lag\ cap$ then
        lag=lag cap;
    else
        $lag = lag$;
    end
end
```

c. *DATA OVERVIEWS*

We have been handed with five features, they are as follows

1.Sequence- These are generally the input features to the model. Amino acid sequence for this sphere. There are 20 truly common amino acids (frequency >), and 4 amino acids that are fairly uncommon X, U, B, O, Z.

2.Family Accession- These are generally the labels for the model. Accession number inform PFxxxxx. y (Pfam), where xxxxx is the family accession, and y is the interpretation number.

Some values of y are lower than ten, and so 'y' has two integers.

3.Sequence name- is the form "uniprot_accession_id / start_index - end_index". Aligned_sequence- Contains one sequence from the multiple sequence alignment with the remainder of the members of the family in seed, with gaps retained.

4.Family_id- One word name for family.

IV. RESULT AND DISCUSSION

RESULT OF SPLITTING DATA

Table. 1. Result of splitting data

Classes	Size
Training	55299
Validation	27179
Testing	27025
Unique Classes	1000

Table1 represents about splitting the entire sequence data we have different sizes for each and every stages like Training, Testing, Validation and there are 1000 unique classes.

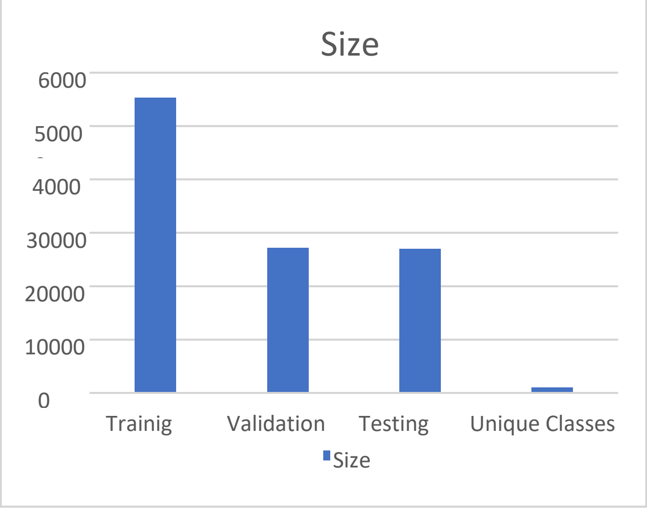
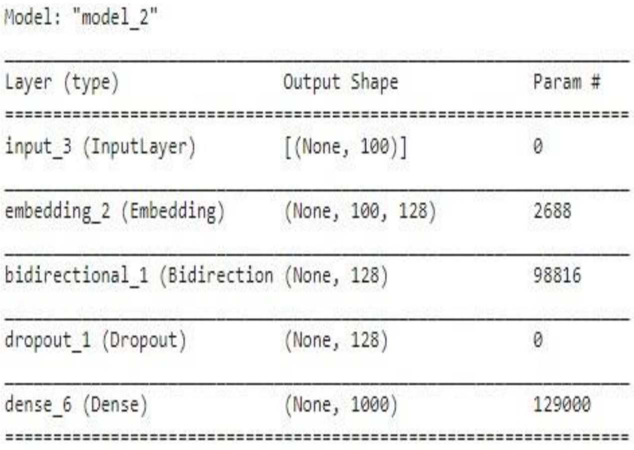


Fig.3. Graph result of splitting data

Figure 3 represents about the size of the three phases of data processing, here size of the training data is high comparing two other two phases.

RESULT OF TRAINING CNN MODEL



```
Model: "model_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_3 (InputLayer)         [(None, 100)]             0
_________________________________________________________________
embedding_2 (Embedding)      (None, 100, 128)          2688
_________________________________________________________________
bidirectional_1 (Bidirection (None, 128)               98816
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_6 (Dense)              (None, 1000)              129000
=================================================================
```

Fig.4. Result of Training CNN Mode1

The Figure 4 represents about the model_2 and there are some values like shape and parameter of each and every layer of the CNN model by using the protein sequence dataset.

FINAL RESULT OF CNN

The table 2 explains about the result of the CNN algorithm by explaining the time taken to complete the iteration value of 10.

Table. 2. Final result of CNN

	Training	Validation	Testing
Time Taken/step	16s73ms/step	8s 73ms/step	8s 73ms/step
Loss	0.0596	0.5927	0.5755
Accuracy	0.982006907	0.886968017	0.8907164
Precision	0.986956596	0.910472274	0.9175877
Recall	0.978354037	0.877183974	0.8814181
F1-Measure	0.982636489	0.89351819	0.8976966

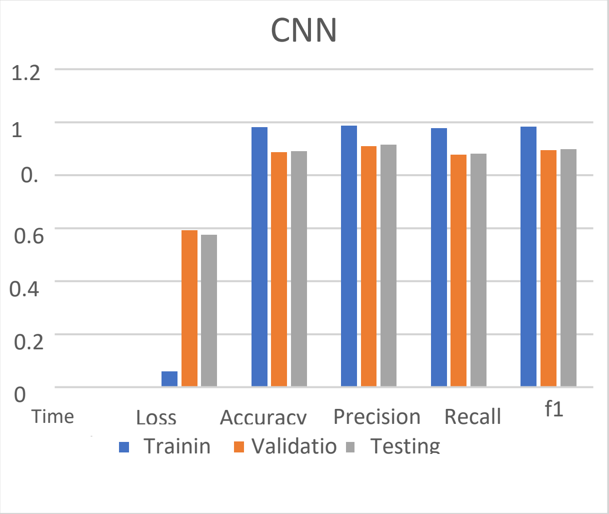


Fig.5. Graphical result of CNN

The Figure5 represents the loss which has less outcome comparing to the other two, and in the accuracy, precision, recall and in f1 result has an same value (0.9%) of training is high comparing to validation and testing and we have measure of training, testing and validation and time taken to run the given amount of iteration.

RESULT OF LSTM

Table.3. Result of LSTM

	Train	Validation	Testing
Time Taken/Step	59s 262ms/step	28s 262ms/step	28s 264ms/step
Accuracy	0.880829692	0.873836756	0.947543919
Precision	0.943532526	0.936967969	0.977345407
Recall	0.840919375	0.829992294	0.917352021
f1 measure	0.889275617	0.88024186	0.88024186
Loss	0.8003	0.8215	0.5335

The above table explains about the result of time taken for running the LSTM. It Gives the value of Accuracy measure, Precision, Recall and f1-measure and the value of the measured loss of the given sequence.

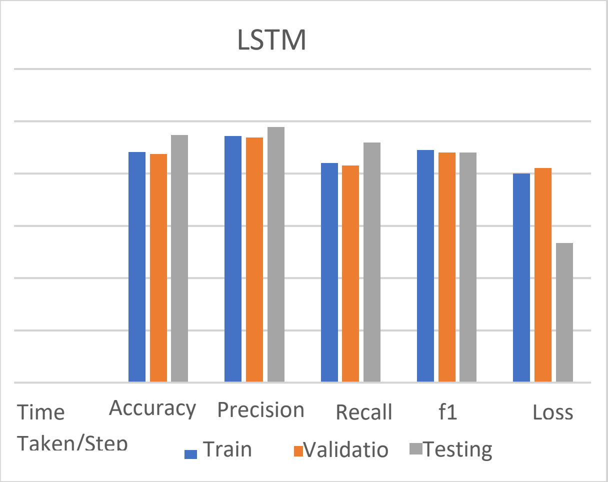


Fig.6. Graphical result of LSTM

The above Figure6 represents the loss which has an low value (0.5%) at the testing time, and in f1 measure the result of the three stages are overall same, and in the recall testing results are high comparing to other two and in precision testing result is high with the value of (0.9%) comparing to the other two time taken to run the given amount of iteration displays on the table shown above.

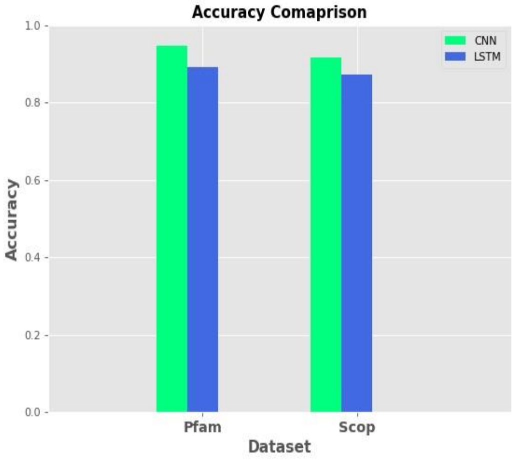


Fig.7. Accuracy Comparison of LSTM And CNN Algorithm

The Figure 7 represents about the accuracy comparison of both CNN and LSTM algorithm by using the help of Pfam and scop dataset. And after comparing both the algorithm CNN secures a high result.

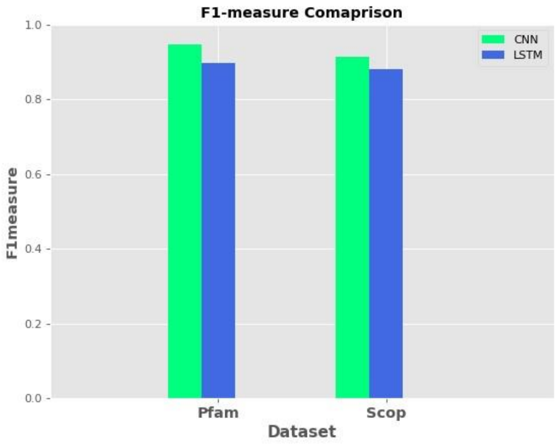


Fig. 8. F1 measure Comparison of LSTM And CNN Algorithm

The Figure 8 represents about the f1 measure comparison of both CNN and LSTM algorithm by using the help of Pfam and scop dataset. And after comparing both the algorithm CNN secures a high result same as accuracy comparison.

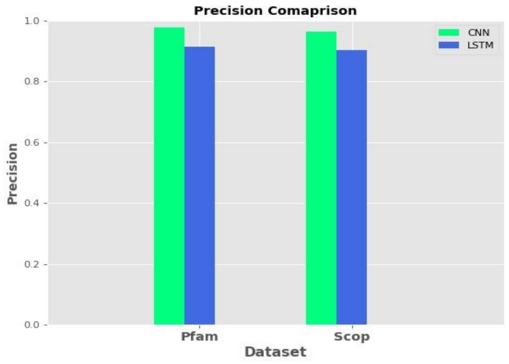


Fig. 9. Precision Comparison of LSTM And CNN Algorithm

The Figure 9 represents about the precision comparison of both CNN and LSTM algorithm by using the help of Pfam and scop dataset. And after comparing both the algorithm CNN secures a high result as done in the above two comparisons.

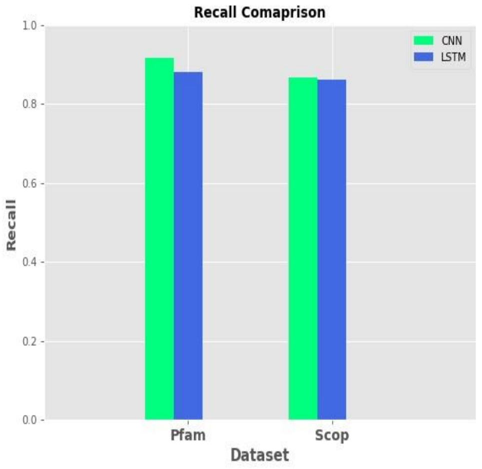


Fig. 10. Recall Comparison of LSTM And CNN Algorithm

The Figure10 represents the recall comparison between CNN and LSTM. By using the pfam and scop dataset, by considering all the above graphs CNN algorithm gives the best and accurate result for our further work so we have considering CNN algorithm as our proposed algorithm.

V. CONCLUSION

In the proposed research, the CNN model was successfully used to detect protein remote homology. The CNN method was achieved the top performance comparing with other existing methods on an SCOP benchmark dataset, a SCOP independent dataset and a protein family dataset. Comparing with handmade protein features used by traditional machine learning methods, the features learnt CNN have more discriminative power. In this case study, we have explored deep literacy models that learn the relationship between unaligned amino acid sequences and their functional reflections. The CNN model has achieved significant results which are more accurate and computationally effective comparing to the other methods to annotate protein sequences. These results suggest deep literacy models will be a core element of unborn protein function vaticination tools.

References

[1]. Singh G.B. (2015) Introduction to Bioinformatics. In: Fundamentals of

[2]. Bioinformatics and Computational Biology. Modeling and Optimization in Science and Technologies, vol 6. Springer, Cham.

[3]. Fogel, G., Corne, D. and Pan, Y. (2008). Computational Intelligence in Bioinformatics. 1st ed. IEE Press Series on Computational Intelligence.

[4]. Raza, K. (2010). Application of Data Mining in Bioinformatics. [online].

[5]. RCSB Protein Data Bank. (2017). Protein Data Bank: Statistics.

[online] Available at: http://www.rcsb.org/pdb/statistics/ [Accessed 21 Mar. 2017].

[6]. Dorna, M., Silva, M, B, e.,.Buriola, L,S.,Lamba,L,C.,(2014) . Threedimensional protein structure prediction: Methods and computational strategies. Computational Biology and Chemistry, ACM Digital Library, Volume 53,Issue PB, Pages 251–276.

[7]. Afendi,F,M.,Ono,N., Nakamura, Y., Nakamura,K., Darusman,L,K.,(2013). Data mining methods for omics and knowledge of crude medicinal plants toward big data biology.

[8]. Hunt, D, F., Yates,J, R., Shabanowitz,J., Winston, S., Hauer, C, R.,(1996). Protein sequencing by tandem mass spectrometry. Proc. Nati. Acad. Sci. USA Volume 83, Issue, 17, Pages 6233–6237.

[9]. JinyongCheng, YihuiLiu, YumingMa, August (2020), Protein secondary structure prediction based on integration of CNN and LSTM model

[10]. Søren Kaae Sønderby, Casper Kaae Sønderby, Henrik Nielsen & Ole

[11]. Winther.(2015),Convolutional LSTM Networks for Subcellular

[12]. Localization of Proteins, 28 July 2015.

[13]. ShuminLi, Junjie Chen & Bin Liu,(2017) Protein remote homology detection based on bidirectional long short- term memory, 10 October 2017.

[14]. Chen-Chen Li, Bin Liu,(2020) MotifCNN-fold: protein fold recognition based on fold-specific features extracted by motif-based convolutional neural networks, Volume 21, Issue 6, November 2020.

[15]. Amarda Shehu, Daniel Barbará, and KevinMolloy,(2016) A Survey of Computational Methods for Protein Function Prediction, October 2016.

[16]. Junjie Chen, Mingyue Guo, Xiaolong Wang, Bin Liu,(2018) A comprehensive review and comparison of different computational methods for protein remote homology detection, Volume 19, Issue 2, March 2018

[17]. Junjie Chen, Ren Long, Xiao-long Wang, Bin Liu & Kuo-Chen Chou,(2016) Detecting remote homology proteins using profile-based pseudo protein sequence and rank aggregation,01 September 2016.

[18]. Inkyung Jung, Dongsup Kim,(2009) Simple protein homology detection method by using indirect signals,2009 Mar 15.

[19]. SutanuBhattacharya, Rahmatullah Roche, Shuvo and Debswapna Bhattacharya., Recent Advances in Protein Homology Detection Propelled by Inter- Residue Interaction Map Threading.

[20]. Jie Hou, Badri Adhikari, Jianlin Cheng,(2018) DeepSF: deep convolutional neural network for mapping protein sequences to folds, Bioinformatics, Volume 34, Issue 8, 15 April 2018.

[21]. Hemalatha Gunasekaran, K. Ramalakshmi,(2021) A. Rex Macedo Arokiaraj, S. Deepa Kanmani, Chandran Venkatesan, and C. Suresh Gnana Dhas, Analysis of DNA Sequence Classification Using CNN and Hybrid Models, Volume 2021, 16 Jul 2021.

[22]. NazarM Zaki,(2012) A comparative analysis of protein homology detection methods, April 2012.

[23]. Lun Hu, Xiaojuan Wang, Yu-An Huang, Pengwei Hu, Zhu-Hong You,(2021) A survey on computational models for predicting protein– protein interactions, 05 March 2021.

[24]. Leo McHugh, Jonathan W Arthur,(2008) Computational Methods for Protein Identification from Mass Spectrometry Data, February 29, 2008.

[25]. Maria Victoria Schneider and Rafael C Jimenez,(2019) Bioinformatics: scalability, capabilities and training in the data driven era, Volume 20, Issue 2, March 2019.

[26]. Jiale Liu & Xinqi Gong,(2019) Attention mechanism enhanced LSTM with residual architecture and its application for protein-protein interaction residue pairs prediction,27 Nov 2019.

[27]. Hailong Hu, Zhong Li, Arne Elofsson, Shangxin Xie,(2019) A Bi-LSTM Based Ensemble Algorithm for Prediction of Protein Secondary Structure by 28 August 2019.

[28]. Mukti Routray & Swati Vipsita,(2021) Protein remote homology detection combining PCA and multiobjective optimization tools, 19 Aug 2021.

[29]. Bin Liu; Shumin Li, (2019) ProtDet-CCH: Protein Remote Homology Detection by Combining Long Short-Term Memory and Ranking Methods, July-Aug.- 2019.