Journal Homepage: -www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI: 10.21474/IJAR01/20074
DOI URL: http://dx.doi.org/10.21474/IJAR01/20074

ISSN NO. 2320-5407

RESEARCH ARTICLE

PROTEIN-PROTEIN INTERACTION PREDICTION USING A DEEP NEURAL NETWORK WITH BATCH NORMALIZATION AND QUARTILE ALGORITHM

Dr. N'Diffon Charlemagne Kopoin[1], Dr. Alex Armand Josué Akohoulé[2], Dr. Wielfrid Morié[3] and Prof. Olivier Pascal Asseu[4]

1. Assistant Professor, Department of Computer Science, Esatic, Cote D'ivoire.
2. Assistant Professor, Department of Computer Science, Esatic, Cote D'ivoire.
3. Assistant Professor, Department of Computer Science, Una, Cote D'ivoire.
4. Professor, Department of Computer Science, Esatic, Cote d'Ivoire.

Manuscript Info

Manuscript History
Received: 15 October 2024
Final Accepted: 18 November 2024
Published: December 2024

Key words:-
Deep Neural Network, Batch Normalization, Protein-Proteininteraction, quartile

Abstract

Detecting protein-protein interactions (PPIs) is key for disease therapy development. While experimental methods are costly, deep neural network (DNN) models now use available PPI data for prediction, though limited by low-quality sequence-based data. This study introduces FDPPI, a DNN model leveraging a quartile-based algorithm and batch normalization to enhance performance, achieving 98.09% accuracy, 98.34% precision, and 97.72% sensitivity on human PPI data.

Introduction

Protein is the essential component of the living organism and is involved in various life processes such as metabolism, signal transduction, hormone regulation, transcription, and DNA replication. In general, proteins perform their complex functions by interacting with other proteins.The study of protein-protein interactions (PPIs) not only helps to understand the process of life, but also explores the parthenogenesis of diseases and helps target drugs for new diseases such as coronavirus. Some high-throughput proteomic techniques such as proteomic arrays [1], immunoprecipitation, two yeast hybrids [2], have been invented to detect PPIs.These experiments have revealed many unknown interactions; however, they are time-consuming and costly. All these limitations underpin the motivation for developing computer models to predict PPIs on a large scale and efficiently.

To date, many computational approaches have been proposed to predict PPIs from different types of data, including gene fusion, gene ontology and annotation [3],3D structural information, and so on. However, these approaches are not universal, and their accuracy and reliability are highly dependent on the prior information gathered on the proteins. In practice, the 3D structure of many proteins is unknown, gene ontology and annotation are incomplete, and PPIs for many species are rarely available.

With the rapid development of sequencing technology, protein sequence information is collected and stored in large quantities in databases such as the Human Protein Database (HPRD) [4], the Database of Interacting Proteins (DIP) [5] and many others such as the Molecular Interaction Database (IntAct) [6] and the Biomolecular Interaction Network Database (BIND) [7]. Indeed, numerous studies have shown that sequence-based prediction can provide very useful information for basic research and drug design and is therefore of great interest to scientists. For example, the identification of proteases and their types [8], the prediction of protein subcellular localization [9]. Chou [10] has proposed the pseudo-amino acid composition (PseAAC) method to improve the quality of prediction

Corresponding Author:-N'Diffon Charlemagne Kopoin
Address:-Assistant Professor, Department of Computer Science, Esatic, Cote D'ivoire.

of subcellular protein localization and membrane protein types.Its aim is to continue to use a discrete model to represent a protein without completely losing information about its sequence order. The PseAAC method generates more than 20 + *m* components, of which the first 20 are the 20 amino acid composition components and the last m components are the sequence order components. Guo et al [11] achieved 86.55% accuracy on S. Cerevisiae PPIs after applying the autocovariance method to discover information in discontinuous amino acid sequence segments. You et al [12], considering the sequence order and dipeptide information of the primary protein sequence, proposed the support vector machine (SVM)-based method [13] to predict protein interaction.This method achieved 90.06% accuracy, 85.74% sensitivity and 94.37% specificity in the yeast protein dataset. Pan et al [14] proposed a new hierarchical LDA-RF model for directly predicting protein-protein interactions in primary protein sequences, which can extract hidden internal structures buried in noisy amino acid sequences in a small latent semantic space.Experimental results show that this model can effectively predict potential protein interactions. Göktepe and Kodaz [15] used the triad-bigram method [16] to predict interactions on human PPI data from the HPRD. Their model achieved 93.45% accuracy, 89.84% precision, 89.29% sensitivity and 85.71% MCC. Recently, Kopoin et al [16], [17] implemented a bigram feature-based method, and combined it with ANN and SVM classifiers to predict PPI interactions.

Models based on deep learning offer outstanding performance in most fields, such as visual object recognition, speech recognition, drug discovery, cancer prediction and more. They have also proven their power in bioinformatics. For example, deep neural networks have been successfully applied to predict RNA splicing patterns across various tissues. Other examples of the successful application of deep learning techniques include genomic information extraction, as well as protein structure prediction.In the field of PPI, few models have used deep learning. Du et al [18] proposed a neural network (DNN) model with amphiphilic pseudo amino acid composition (APAAC) [19]. The peculiarity of this model is that they took the APAAC-extracted features of two respective proteins as inputs to two distinct deep neural networks (DNA) to predict PPIs. Subsequently, the two learning processes are concatenated to provide a single output network. With this model, performances of 95.7%, 98.6%, 91.10% and 92.5% were observed in precision, accuracy, recall and MCC on S. Cerevisiae PPIs, respectively.Zhang et al [20] have proposed a deep learning ensemble for PPIs. In this model, 3 extraction methods that have produced different features are learned by one network each. After a certain level, the learning information is pooled to produce efficient results. Thus, for precision, they record 92.5% accuracy, 99.3% precision and 95.6% recall. Algorithms based on neural networks perform well with large volumes of data.However, if data quality is poor (e.g. outliers) and the model is not optimized, it will be difficult to achieve the expected performance. In this study, we propose a neural network-based PPI prediction model. To enhance the model's performance, we have developed a data pre-processing algorithm based on Tuckey criteria [21] to evaluate input dispersion. Then, to make learning faster and more efficient, we use batch normalization [22] before each network activation function.

The remainder of this document is organized as follows. In the following sections, we detail the datasets, the feature extraction method used, the proposed solution, the various results obtained and the discussion, and the final section is devoted to the conclusion and future work.

Material

In this work, we propose the implementation of a PPI prediction model using only protein sequences. Thus, we acquired a reference dataset of human PPIs from the work of Pan et al [14]. The positive data in this dataset were collected from the Human Protein Reference Database (HPRD, version 2007), a total of 36630 positive interactions from 9630 different human proteins were formed. For negative PPI pairs, 36480 negative interactions were obtained from 1773 proteins from 6 subcellular locations.

To evaluate the performance of our model, we used two other different PPI datasets. The first PPI dataset is also collected from the HPRD data described by Huang et al [24], which consists of 8161 human protein pairs (3899 interacting pairs and 4262 non-interacting pairs). The second dataset is the IPP dataset described by You et al. [25], this dataset is collected from the core S. Cerevisiae subset in the Interacting Proteins Database (IPD). This dataset consists of 5594 positive and 5594 negative pairs, combined in a total of 11188 proteins.

The various PPI datasets are in FASTA format. The FASTA format facilitates the manipulation and analysis of sequences using word processing tools and scripting languages such as R, Python, Ruby and Perl.

Feature extraction method

In this study, we used the amphiphilic pseudo amino acid composition to extract amino acid sequence features. Based on the pseudo amino acid composition (PseAAC) [10], the amphiphilic pseudo amino acid composition (APAAC) [19] qualifies as a type-2 pseudo amino acid composition. Whereas the PAAC method yields 20 + λ components (λ less than the sequence length), the APAAC method yields 20 + 2λ components. The first 20 components of the descriptor vector are the first type of information, which is simply the frequency of occurrence of each amino acid, called amino acid composition (AAC). The second type of information is the order-sequence correlation of amino acid residues adjacent at a certain distance, represented by the last two components, as shown in Fig. 1. Consider a protein P composed of *L* amino acid residues:

$$R_1R_2R_3 \ldots R_6 \ldots R_L \quad (1)$$

with R_1, the residue at position 1 of the chain, and so on. Thus, APAAC uses hydrophobicity and hydrophilicity values [25] to calculate their correlated functions. First, hydrophobicity and hydrophilicity values are derived from their original value (the original value can be found in [27]).

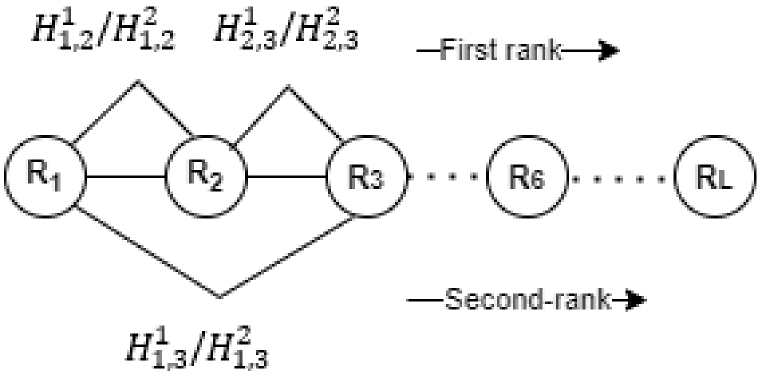

Fig. 1:- APAAC sequence-order correlations.

Assuming that $H_1^0$and H_2^0 are, respectively, the original hydrophobicity and hydrophilicity values of the amino acid *R*i (i = 1, 2..., 20), the derived values are calculated according to equation 2:

$$\begin{cases} H_1^*(R_i) = \dfrac{H_1^0(R_i)-\mu_1}{\sqrt{\sum_{i=1}^{20}\left[H_1^0(R_i)-\mu_1\right]^2 / 20}} \\ H_2^*(R_i) = \dfrac{H_2^0(R_i)-\mu_2}{\sqrt{\sum_{i=1}^{20}\left[H_2^0(R_i)-\mu_2\right]^2 / 20}} \end{cases} \quad (2)$$

where μ_1 and μ_2 are the average hydrophobicity and hydrophilicity values of the 20 amino acids, respectively, andH_1^*, H_2^* the hydrophilicity and hydrophobicity correlation functions defined as in equation 3:

$$H_{i,j}^1 = H_1^*(R_i) * H_1^*(R_j); H_{i,j}^2 = H_2^*(R_i) * H_2^*(R_j) \quad (3)$$

To represent the protein and account for sequence order information in amino acid composition, APAAC descriptors are defined by equation 4:

$$P_m = \begin{cases} \dfrac{f_m}{\sum_{i=1}^{20} f_i + w\sum_{j=1}^{\lambda}\sigma_j}, & (1 \le m \le 20) \\ \dfrac{w\tau_m}{\sum_{i=1}^{20} f_i + w\sum_{j=1}^{\lambda}\sigma_j}, & (20+1 \le m \le 20+\lambda) \end{cases} \quad (4)$$

where *w* represents the weight factor (generally set at 0.5 [10]), fi (*i*= 1,2,...,20) are the normalized frequencies of occurrence of the 20 amino acids in the proteinP, σ_j, the correlation factor of rank j which reflects the order-sequence correlation function between all the j-th most contiguous residues, is defined by equation 5:

$$\sigma_j = \begin{cases} \frac{1}{L-k}\sum_{i=1}^{L-1} H_{i,i+1}^{1}, & if\ j = 2k-1 \\ \frac{1}{L-k}\sum_{i=1}^{L-1} H_{i,i+1}^{2}, & if\ j = 2k \end{cases} \tag{5}$$

where L is the length of the sequence.
In this work, we took λ= 15. This gave a total of 20 + 2 × 15 = 50 APAAC descriptors. The protein pair is obtained by concatenating the features of the two interacting proteins, i.e. 100 APAAC descriptors.

Quartile algorithm

Although neural network algorithms are excellent for learning mass data, if inputs are of poor quality, this can affect model performance. Here, we have developed a quartile algorithm (D-Filter) based on Tuckey's criteria [22] to assess the dispersion of inputs before learning by the neural network (Fig. 2).The D_fiter algorithm is based on the determination of outliers according to the tuckey criterion. This algorithm uses the 1st quartile (Q_1) and 3rd quartile (Q_3) to determine the interquartile range, then deletes each line that contains a few outliers according to the Tuckey criteria. Quartiles are determined from the median, which is the value that divides the series of observations into two groups of equal size. The method involves determining the values of the lower and upper limits of a box (also known as a moustache box) as follows:

$$lb = Q1 - 1.5 \times Q\ ;\ ub = Q3 + 1.5 \times Q \tag{6}$$

with lb, ub and Q, respectively the lower limits, upper limits and interquartile range. The interquartile range is represented by the difference between the 3rd quartile and the 1st quartile (Q_3-Q_1). All values outside the range [lb, ub] are considered outliers. To deal with these outliers, we remove lines containing more than φ outliers. The choice of parameter φ depends on the data, because an outlier is not necessarily bad.

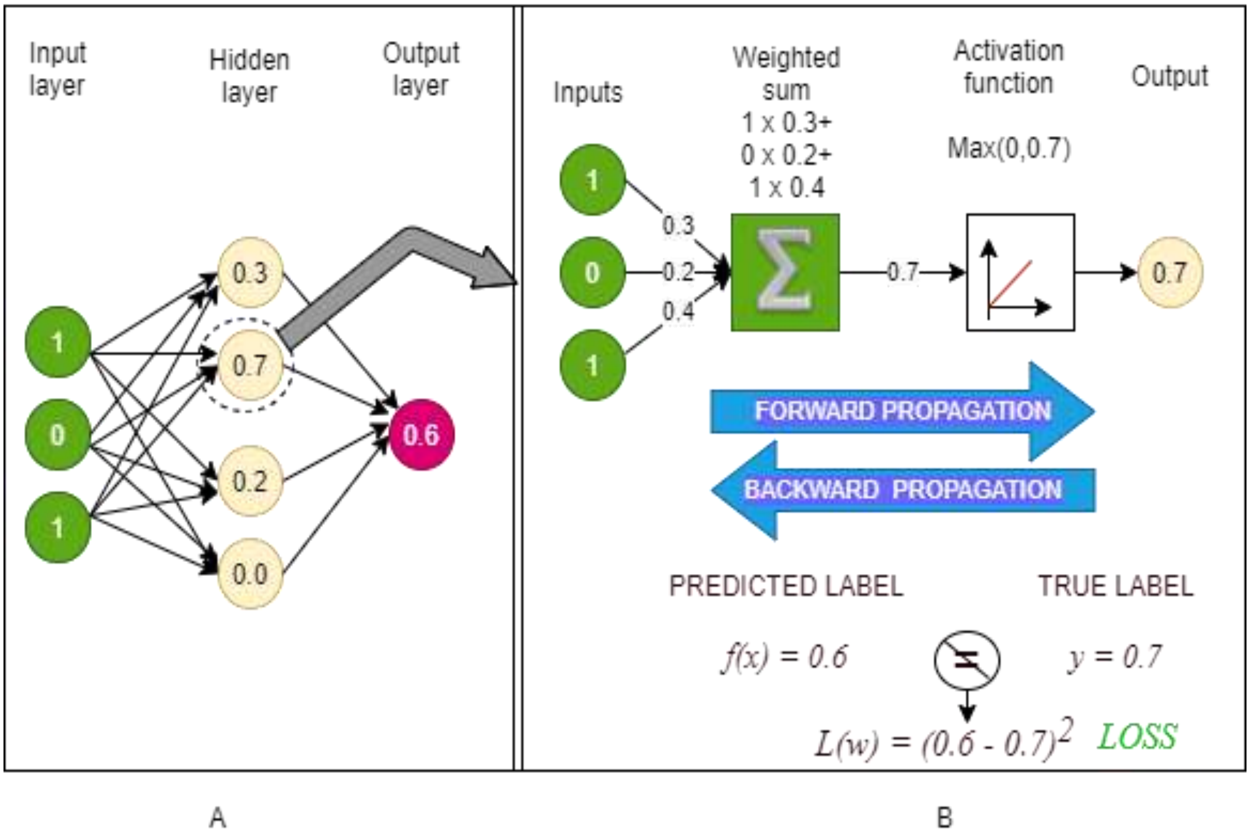

Fig. 2:- Architecture of neural network training procedure. The left-hand side (part A) is the architecture of a hidden layer, and the right-hand side (part B) is the procedure for calculating the output of a hidden layer.

A pseudo code of the D-Filter algorithm is given below:

D-Filteralgorithm

```
Input :    Dataset
Output :   New Dataset
Nb_line=[]  //nb_line is a list
For each linei of the dataset do :
$Q_1$= 1st quartile of i
$Q_3$ = 3rd quartile of i
Threshold= 1.5 x ($Q_3$-$Q_1$)
    Outlier = (value with value <$Q_1$- Threshold | val>$Q_3$ + Threshold)
    Add Outlier to listNb_line
End for
New Dataset = Dataset - Nb_line
```

Neural network training procedure

The initial framework for deep learning was built on artificial neural networks (ANNs) in the 1980s [27],while the real impact of deep learning emerged in 2006. Since then, deep learning has been applied to a wide range of fields, including automatic speech recognition, image recognition, natural language, drug discovery and bioinformatics [17], [28].

In general, a neural network receives data through the input layer, then transforms this data in a non-linear way through several hidden layers and produces the prediction through the output layer (fig. 2-A). Neurons in a hidden or output layer are connected to all neurons in the previous layer. Each neuron calculates a weighted sum of its inputs and applies a non-linear activation function to calculate its f(x) outputs (fig. 2-B).

The depth of a neural network corresponds to the number of hidden layers, and the width to the maximum number of neurons in one of its layers. The most popular activation function is the rectified linear unit function (ReLU), which reduces negative signals to 0 and passes through a positive signal, and is defined by:

$$Relu(x) = \max(0, x)$$

This type of activation function allows faster learning while avoiding the problem of saturation compared with other methods (e.g. sigmoid or tanh [29]). Alternative architectures to these fully connected flow-networks have been developed for specific applications. These include convolutional structure [30], recurrent neural networks for sequential data [31], restricted Boltzmann machines [17], etc.

Proposed model

The APAAC descriptors of the protein pairs will first be filtered using the D-Filter algorithm before being ejected into the neural network for the learning process (fig. 3). Once this step has been completed, we apply batch normalization (BN) before each activation function for faster learning and increased prediction performance.BN offers an elegant way of reparametrizing almost any deep network. Reparameterization dramatically reduces the problem of coordinating updates across multiple layers. It does this by circling the layer output, including normalizing the activations of each input variable in a mini-lot, like the activations of a node in the previous layer. It ensures that gradients are more predictive, enabling a wider range of learning rates and faster network convergence.

Our neural network model has three hidden layers with 256, 128 and 64 neurons, respectively. The ReLU activation function is applied at all levels of the network, except at the output where we applied the sigmoid activation function.

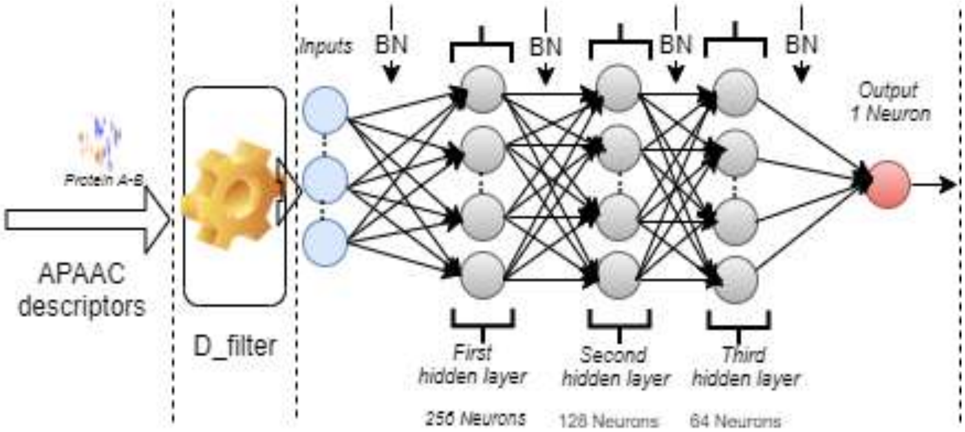

Fig. 3:- Architecture of FDPPI.

In summary, our model is based on the following steps, which are the main contributions: the use of the D-Filter algorithm at the network input to have homogeneous and consistent data for better prediction performance; the input dispersion analysis algorithm to detect outliers that could constitute abnormal or poor data. The use of batch normalization not only speeds up learning, but also boosts performance.

Results and Discussion:-

Experiments were carried out with python software version 3.7 on a basic i7 machine with 16 GB RAM, for the D-Filter parameter we chose φ = 8 (below 8, results are unsatisfactory), and for the learning rate we chose lr = 0.01.

To evaluate our model, we used the following measures: Accuracy (Acc), Precision (Pre), Sensitivity (Sen), Matthews correlation coefficient (Mcc), Area under the ROC curve (AUC) and Recall precision curve (P-R curve). Some of these measures are defined as follows:

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \quad (7)$$

$$Pre = \frac{TP}{TP+FP} \quad (8)$$

$$Sen = \frac{TN}{TN+FN} \quad (9)$$

$$Mcc = \frac{TP \times N - FP \times FN}{\sqrt{(TP+FN)(TP+FP)(TN+FN)(TN+FP)}} \quad (10)$$

Here, TP (true positive) is the number of PPIs expected to be positive, i.e. interact and really interact, FP is the number of PPIs expected to be positive, but not positive, TN is the number of PPIs expected to be negative, i.e. not interact and not really interact evenly, and FN is the number of PPIs expected to be negative but are not negative.MCC is a measure of the quality of binary classification, which is a correlation based on the coefficient between observed and predicted results. It returns the value between - 1 (is considered a false prediction) and +1 (is considered an interesting prediction).The precision recall curve (P-R curve) is a diagnostic tool that helps interpret probabilistic predictions for binary classification predictive modeling problems. In addition, the accuracy-recall curve summarizes the trade-off between true positive rate and positive predictive value for a predictive mo del using different probability thresholds and is recommended in cases of unbalanced data. Til ROC curve and AUC value graphically illustrate the performance of a binary classification system.

First, we used 5-fold cross-validation on human PPI data to evaluate the performance of our model. From Table 1, we can see that FDPPI performs better than 97% on average, with 98.09% in accuracy, 9 8.34%±0.27% in precision, 97. 37%± 0.28% in sensitivity, 96.14%± 0.27% in Mcc and 99.51%± 0.12% in AUC.

Table 1:-Result of five-cross validation with FDPPI on the HPRD dataset.

Fold	*Acc*	*Pré*	*Sen*	*Mcc*	*Auc*
1	98.03%	97.88%	97.72%	96.03%	99.30%
2	98.28%	98.69%	97.17%	96.53%	99.67%
3	98.10%	97.58%	98.19%	96.17%	99.36%
4	98.16%	98.57%	97.29%	96.28%	99.36%

5	97.84%	98.71%	96.43%	95.65%	99.32%
Mean	**98.09%**	**98.34%**	**97.37%**	**96.14%**	**99.51%**
std	**±0.14%**	**±0.31%**	**±0.28%**	**±0.27%**	**±0.12%**
Std correspondent à l'écart type					

In Fig. 4, we show the training loss and validation curves of our model. Before the 60th epoch, both curves descend drastically towards the origin. From the 60th epoch onwards, the two curves merge and continue to descend. This reflets a good model fit.

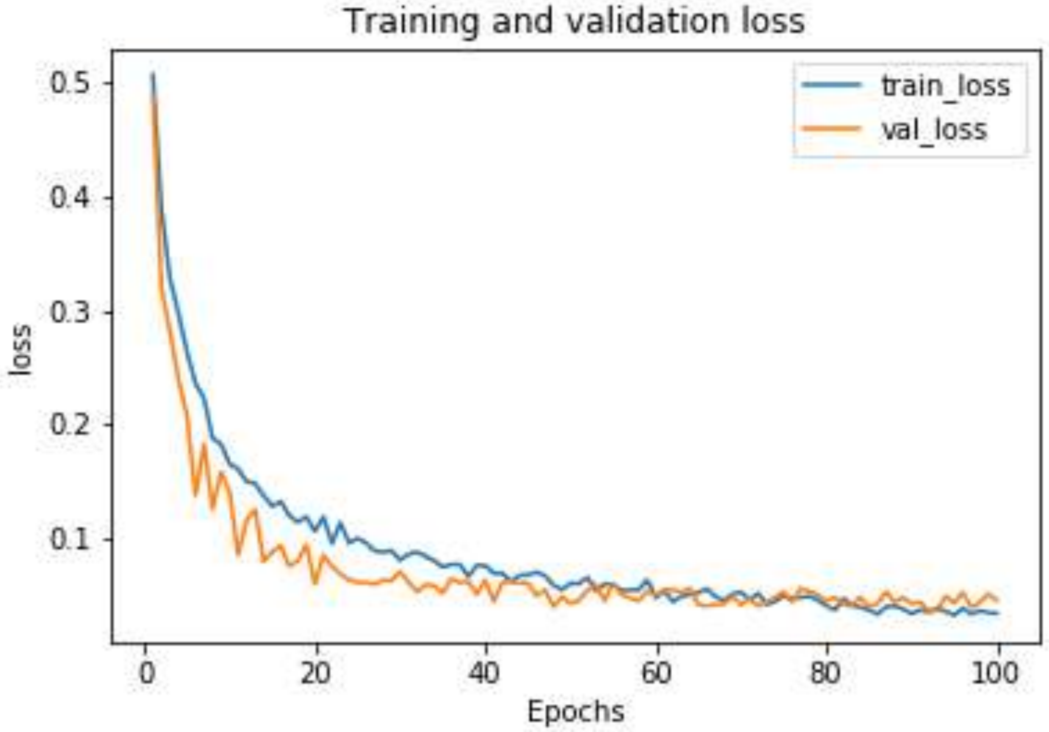

Fig. 4:- Loss and validation curves of FDPPI.

Figure 5 shows the different Precision-Recall (P-R) curves during 5-fold training. We can see that, overall, the model has a good area with over 99% each time.

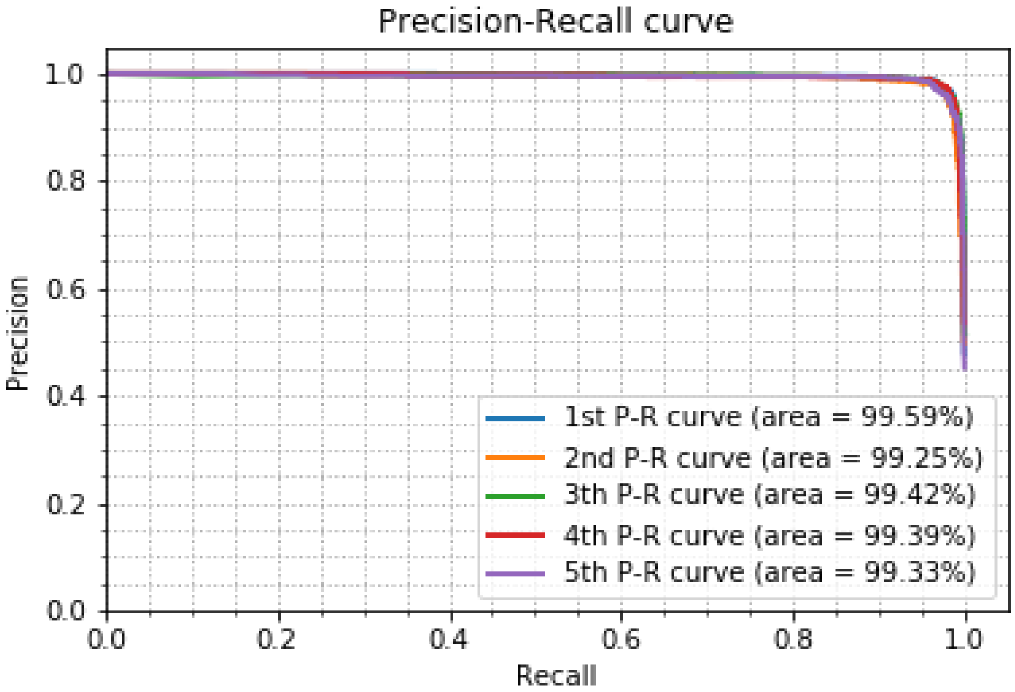

Fig. 5:- Five-cross P-R curves on the HPRD dataset.

To demonstrate the robustness of our FDPPI model, we compared the performance obtained with that of the DeepPPI model by Du et al [18] on HPRD training data. We acquired a refactoring of the DeepPPI model source code from [32] on GitHub. The results in Fig. 6 below show that our FDPPI model learns better than the DeepPPI model on training data, with performance of over 2% on all the measures used, i.e. accuracy, precision, sensitivity and Mcc.

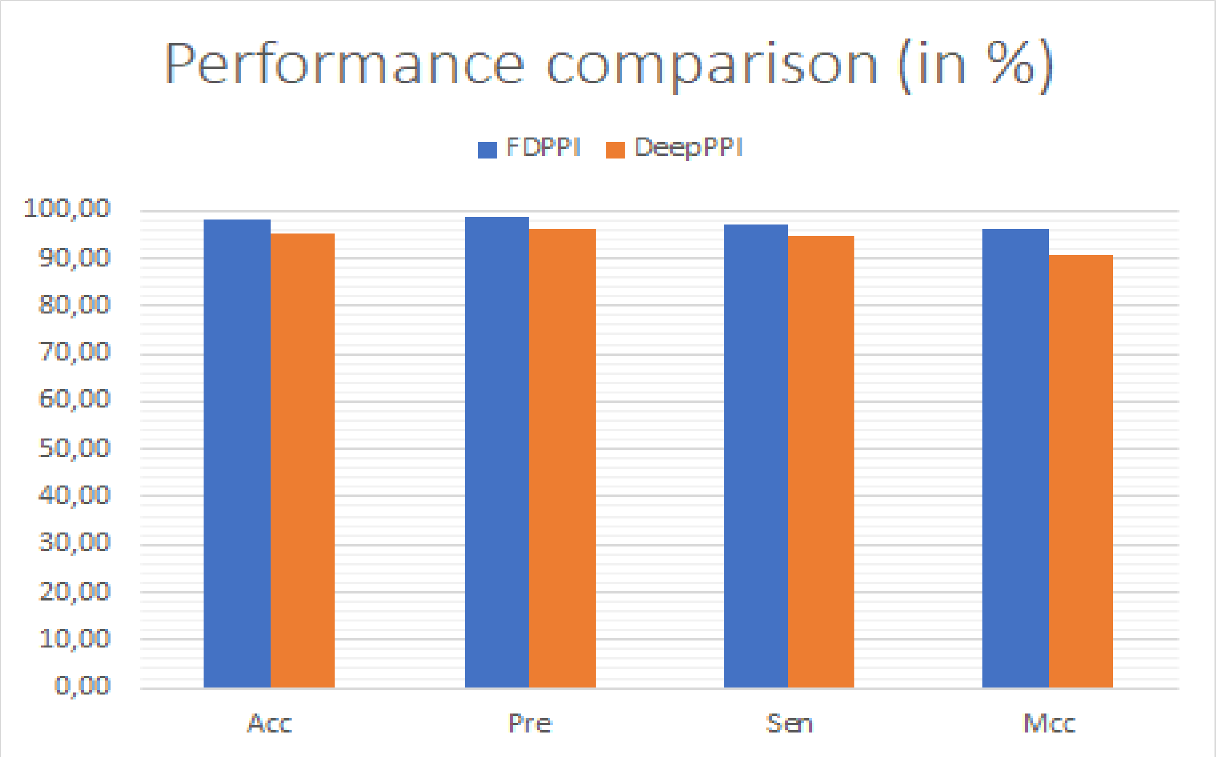

Fig. 6:- Performance comparison between FDPPI and DeepPPI on the HPRD dataset

Table 2:- Learning time comparison between the FDPPI and DeepPPI models.

Model	*Paramètre*	*Time (seconde)*
DeepPPI (Du et al.[18])	Epoch = 100;lr = 0.01	758 s
FDPPI		530 s

lr taux d'apprentissage moyen

We also compared the time taken by our model and the time taken by the DeepPPI model to learn the HPRD data. For a better comparison, we took the same number of epochs (epoch =100) and the same learning rate (learning rate=0.01). Table 2 shows that our model learns faster than the DeepPPI model, by over 200 seconds. This demonstrates better performance in terms of execution time.In Table 3, we show the results of comparing the performance of our model with that of other authors on the human HPRD data described by pan et al. [34]. Except in Mcc where the LDA-RF model outperforms ours by 0.24%, our FDPPI model performs better in all other measures with 0. 14%, 8.65%, 0.56%, and 0. 34% greater in accuracy, precision, sensitivity, and Mcc, respectively.

Table 3:- Comparison with other authors on the HPRD dataset.

Method	*Acc*	*Pre*	*Sen*	*Mcc*
Triade [15]	93.45%	89.84%	89.29%	74.22%
ELM [24]	88.87%	N/A	88.31%	50.22%
LDA-RF [14]	97.95%	N/A	96.96%	95.76%
Kopoin et al. [17]	96.16%	95.97%	96.29%	94.76%
FDPPI	**98.09%**	**98.34%**	**97.72%**	**96.14%**

N/A average not available

In Table 4, we have compared the performance of our FDPPI model with the DeepPPI [18] and DCT+SMR [22] models on H. sapien data in precision, accuracy, sensitivity and Mcc. The performances achieved by our FDPPI model are 97.88% in precision, 97.98% in accuracy, 97.27% in sensitivity and 95.71% in Mcc. Our model achieves the best performance in terms of accuracy and sensitivity. The best precision is obtained by the DCT+SMR model with 99.59%, while the best Mcc is obtained by the DeepPPI model with 96.29%.

Table 4:- Comparison on the Homo Sapien dataset.

Method	*Acc*	*Pre*	*Rec*	*Mcc*
DCT + SMR	96.30%	99.59%	92.63%	92.52%
DeepPPI (D)	88.14%	99.13%	96.95%	96.29%
FDPPI	**97.88%**	**97.98%**	**97.27%**	**95.71%**

In Table 5, we have compared our results with those of Wong et al [33], You et al [24] and Zhou et al [34] on the S. Cerevisiae dataset described by You et al [25]. The performances obtained by our model are 94.27%, 95.01%, 91.68% and86.49% in accuracy, precision, sensitivity and Mcc, respectively. Apart from the accuracy metric, where the best performance is obtained by Wong et al. with 96.45%, our model performs better in all metrics than those of other authors.

Table 5:- Comparison on the S. Cerevisiae dataset.

Model	*Acc*	*Pre*	*Sen*	*Mcc*
Wong et al [33]	93.92%	96.45%	91.10%	88.56%
You et al [24]	87%	87.59%	86.15%	77.36%
Zhou et al [34]	88.56%	89.50%	87.37%	77.15%
FDPPI	**94.27%**	**95.01%**	**91.68%**	**89.49%**

Conclusion:-

Protein-protein interactions play an important role in therapeutic targeting. With new diseases such as covid19 or coronavirus, research to identify protein-protein interactions is of great help in the search for drug solutions. As part of this work, we have proposed a model for predicting protein-protein interactions based on deep learning. We tested our model on HPRD, Homo Sapiens and S. cerevisiae, which are widely used for PPI prediction. The overall results show that the proposed FDPPI model works well on training and test data. Looking ahead, we will test our models on larger data on other mechanisms such as protein discovery, which is essential for organism maintenance.

Références:-

[1] H. Zhu *et al.*, ‘Global Analysis of Protein Activities Using Proteome Chips’, *Science*, vol. 293, no. 5537, pp. 2101–2105, Sep. 2001, doi: 10.1126/science.1062191.

[2] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki, ‘A comprehensive two-hybrid analysis to explore the yeast protein interactome’, *PNAS*, vol. 98, no. 8, pp. 4569–4574, Apr. 2001, doi: 10.1073/pnas.061034498.

[3] C. D. Nguyen, K. J. Gardiner, and K. J. Cios, ‘Protein annotation from protein interaction networks and Gene Ontology’, *Journal of Biomedical Informatics*, vol. 44, no. 5, pp. 824–829, Oct. 2011, doi: 10.1016/j.jbi.2011.04.010.

[4] T. S. Keshava Prasad *et al.*, ‘Human Protein Reference Database--2009 update’, *Nucleic Acids Research*, vol. 37, no. Database, pp. D767–D772, Jan. 2009, doi: 10.1093/nar/gkn892.

[5] I. Xenarios, D. W. Rice, L. Salwinski, M. K. Baron, E. M. Marcotte, and D. Eisenberg, ‘DIP: the Database of Interacting Proteins’, *Nucleic Acids Res*, vol. 28, no. 1, pp. 289–291, Jan. 2000.

[6] B. Aranda *et al.*, ‘The IntAct molecular interaction database in 2010’, *Nucleic Acids Res.*, vol. 38, no. Database issue, pp. D525-531, Jan. 2010, doi: 10.1093/nar/gkp878.

[7] G. D. Bader, D. Betel, and C. W. Hogue, ‘BIND: the biomolecular interaction network database’, *Nucleic acids research*, vol. 31, no. 1, pp. 248–250, 2003.

[8] K.-C. Chou and H.-B. Shen, ‘ProtIdent: A web server for identifying proteases and their types by fusing functional domain and sequential evolution information’, *Biochemical and Biophysical Research Communications*, vol. 376, no. 2, pp. 321–325, Nov. 2008, doi: 10.1016/j.bbrc.2008.08.125.

[9] K.-C. Chou, ‘Prediction of Protein Subcellular Locations by Incorporating Quasi-Sequence-Order Effect’, *Biochemical and Biophysical Research Communications*, vol. 278, no. 2, pp. 477–483, Nov. 2000, doi: 10.1006/bbrc.2000.3815.

[10] K.-C. Chou, ‘Pseudo Amino Acid Composition and its Applications in Bioinformatics, Proteomics and System Biology’, *Current Proteomics*, vol. 6, no. 4, pp. 262–274, Dec. 2009, doi: 10.2174/157016409789973707.

[11] Y. Guo, L. Yu, Z. Wen, and M. Li, ‘Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences’, *Nucleic Acids Res*, vol. 36, no. 9, pp. 3025–3030, May 2008, doi: 10.1093/nar/gkn159.

[12] Z.-H. You, X. Li, and K. C. Chan, ‘An improved sequence-based prediction protocol for protein-protein interactions using amino acids substitution matrix and rotation forest ensemble classifiers’, *Neurocomputing*, vol. 228, pp. 277–282, Mar. 2017, doi: 10.1016/j.neucom.2016.10.042.

[13] S. B. Rakhmetulayeva, K. S. Duisebekova, A. M. Mamyrbekov, D. K. Kozhamzharova, G. N. Astaubayeva, and K. Stamkulova, ‘Application of Classification Algorithm Based on SVM for Determining the Effectiveness of Treatment of Tuberculosis’, *Procedia Computer Science*, vol. 130, pp. 231–238, Jan. 2018, doi: 10.1016/j.procs.2018.04.034.

[14] X.-Y. Pan, Y.-N. Zhang, and H.-B. Shen, ‘Large-Scale Prediction of Human Protein−Protein Interactions from Amino Acid Sequence Based on Latent Topic Features’, *J. Proteome Res.*, vol. 9, no. 10, pp. 4992–5001, Oct. 2010, doi: 10.1021/pr100618t.

[15] Y. E. Göktepe and H. Kodaz, ‘Prediction of Protein-Protein Interactions Using An Effective Sequence Based Combined Method’, *Neurocomputing*, vol. 303, pp. 68–74, Aug. 2018, doi: 10.1016/j.neucom.2018.03.062.

[16] C. N. Kopoin, Nt. Tchimou, B. K. Saha, and M. Babri, ‘A Feature Extraction Method in Large Scale Prediction of Human Protein-Protein Interactions using Physicochemical Properties into Bi-gram’, in *2020 IEEE International Conf on Natural and Engineering Sciences for Sahel’s Sustainable Development - Impact of Big Data Application on Society and Environment (IBASE-BF)*, Feb. 2020, pp. 1–7. doi: 10.1109/IBASE-BF48578.2020.9069594.

[17] C. N. Kopoin, A. K. Atiampo, B. G. N’Guessan, and M. Babri, ‘Prediction of Protein-Protein Interactions from Sequences using a Correlation Matrix of the Physicochemical Properties of Amino Acids’, *International journal of computer science and network security: IJCSNS*, vol. 21, no. 3, pp. 41–47, 2021.

[18] C. Cao *et al.*, ‘Deep Learning and Its Applications in Biomedicine’, *Genomics, Proteomics & Bioinformatics*, vol. 16, no. 1, pp. 17–32, Feb. 2018, doi: 10.1016/j.gpb.2017.07.003.

[19] X. Du, S. Sun, C. Hu, Y. Yao, Y. Yan, and Y. Zhang, ‘DeepPPI: Boosting Prediction of Protein–Protein Interactions with Deep Neural Networks’, *J. Chem. Inf. Model.*, vol. 57, no. 6, pp. 1499–1510, Jun. 2017, doi: 10.1021/acs.jcim.7b00028.

[20] K.-C. Chou, ‘Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes’, *Bioinformatics*, vol. 21, no. 1, pp. 10–19, Jan. 2005, doi: 10.1093/bioinformatics/bth466.

[21] L. Zhang, G. Yu, D. Xia, and J. Wang, ‘Protein-Protein Interactions Prediction based on Ensemble Deep Neural Networks’, *Neurocomputing*, May 2018, doi: 10.1016/j.neucom.2018.02.097.

[22] M. Le Guen, ‘La boîte à moustaches de TUKEY, un outil pour initier à la statistique’, 2001.

[23] N. Bjorck, C. P. Gomes, B. Selman, and K. Q. Weinberger, ‘Understanding batch normalization’, *Advances in Neural Information Processing Systems*, vol. 31, pp. 7694–7705, 2018.

[24] Y.-A. Huang, Z.-H. You, X. Chen, K. Chan, and X. Luo, ‘Sequence-based prediction of protein-protein interactions using weighted sparse representation model combined with global encoding’, *BMC Bioinformatics*, vol. 17, no. 1, p. 184, Dec. 2016, doi: 10.1186/s12859-016-1035-4.

[25] Z.-H. You, Y.-K. Lei, L. Zhu, J. Xia, and B. Wang, ‘Prediction of protein-protein interactions from amino acid sequences with ensemble extreme learning machines and principal component analysis’, *BMC Bioinformatics*, vol. 14, no. S8, p. S10, May 2013, doi: 10.1186/1471-2105-14-S8-S10.

[26] G. D. Rose, A. R. Geselowitz, G. J. Lesser, R. H. Lee, and M. H. Zehfus, ‘Hydrophobicity of amino acid residues in globular proteins’, *Science*, vol. 229, no. 4716, pp. 834–838, Aug. 1985, doi: genetic.

[27] Z.-H. You, J.-Z. Yu, L. Zhu, S. Li, and Z.-K. Wen, ‘A MapReduce based parallel SVM for large-scale predicting protein–protein interactions’, *Neurocomputing*, vol. 145, pp. 37–43, Dec. 2014.

[28] M. W. Gardner and S. R. Dorling, ‘Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences’, *Atmospheric environment*, vol. 32, no. 14–15, pp. 2627–2636, 1998.

[29] U. Michelucci, *Applied Deep Learning: A Case-Based Approach to Understanding Deep Neural Networks*. Berkeley, CA: Apress, 2018. doi: 10.1007/978-1-4842-3790-8.

[30] A. Wanto, A. P. Windarto, D. Hartama, and I. Parlina, ‘Use of Binary Sigmoid Function And Linear Identity In Artificial Neural Networks For Forecasting Population Density’, vol. 1, no. 1, p. 13, 2017.

[31] L. Nanni, S. Ghidoni, and S. Brahnam, ‘Ensemble of convolutional neural networks for bioimage classification’, *Applied Computing and Informatics*, Jun. 2018, doi: 10.1016/j.aci.2018.06.002.

[32] I. Sutskever, J. Martens, and G. E. Hinton, ‘Generating Text with Recurrent Neural Networks’, Nov. 2018, Accessed: Dec. 04, 2018. [Online]. Available: https://openreview.net/forum?id=SkeV89Gfpm

[33] G. d’Ario, *gdario/deep_ppi*. (Feb. 21, 2020). Python. Accessed: Dec. 06, 2020. [Online]. Available: https://github.com/gdario/deep_ppi

[34] L. Wong, Z.-H. You, S. Li, Y.-A. Huang, and G. Liu, ‘Detection of protein-protein interactions from amino acid sequences using a rotation forest model with a novel PR-LPQ descriptor’, in *International Conference on Intelligent Computing*, Springer, 2015, pp. 713–720.

[35] Y. Z. Zhou, Y. Gao, and Y. Y. Zheng, ‘Prediction of Protein-Protein Interactions Using Local Description of Amino Acid Sequence’, in *Advances in Computer Science and Education Applications*, vol. 202, M. Zhou and H. Tan, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 254–262. doi: 10.1007/978-3-642-22456-0_37.