Indonesian Journal of Electrical Engineering and Computer Science
Vol. 22, No. 2, May 2021, pp. 1108~1115
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v22.i2.pp1108-1115

DeepEnz: prediction of enzyme classification by deep learning

Hamza Chehili[1], Salah Eddine Aliouane[2], Abdelhafedh Bendahmane[3], Mohamed Abdelhafid Hamidechi[4]
[1]LIRE Laboratory-Constantine 2, University of Constantine 1, Algeria
[2,3,4]Department of Applied Biology, University of Constantine 1, Algeria

Article Info

Article history:

Received Dec 31, 2020
Revised Mar 23, 2021
Accepted Mar 30, 2021

Keywords:

Classes
Deep learning
Enzyme
NLP
Prediction
Protein

ABSTRACT

Previously, the classification of enzymes was carried out by traditional heuritic methods, however, due to the rapid increase in the number of enzymes being discovered, new methods aimed to classify them are required. Their goal is to increase the speed of processing and to improve the accuracy of predictions. The Purpose of this work is to develop an approach that predicts the enzymes' classification. This approach is based on two axes of artificial intelligence (AI): natural language processing (NLP) and deep learning (DL). The results obtained in the tests show the effectiveness of this approach. The combination of these two tools give a model with a great capacity to extract knowledge from enzyme data to predict and classify them. The proposed model learns through intensive training by exploiting enzyme sequences. This work highlights the contribution of this approach to improve the precision of enzyme classification.

Corresponding Author:

Hamza Chehili
LIRE Laboratory-Constantine 2
University of Constantine 1
Constantine, 25000, Algeria
Email: h.chehili@umc.edu.dz

1. INTRODUCTION

The development of technology has led to the explosion of the quantity of biological data. Like genomics data, enzyme data has known a growing accumulation [1]. Those enzymes belong mainly to six classes: oxidoreductase, transferase, hydrolase, lyase, isomerase and ligase [2]. Bioinformatics is at the heart of this data: it offers the different tools and methods to process and interpret this data [3]. In order to model biological problems, bioinformatics uses mathematics, statistics and computer science [4].

One of the problems addressed by bioinformatics is the classification of proteins to identify the biological functions of them [5], [6] particlulary, the enzymes that know a huge number of discovry [1]. This is an important post-genomic and bioinformatics step [7] that follows the high throughput sequencing [8]. In fact, the primary sequence of the protein obtained after translation will be annotated in order to determine its function. This annotation relies mainly on the fine structure of the protein (2D and 3D structures) and on its physio-enzymatic function.

Biological functions are very important, but their laboratory studies are very expensive [9]. Since the generation of data from newly sequenced proteins is increasing considerably [10], bioinformatics researchers have resorted to automatic prediction of their functions, primarily through computer modeling methods [11]. Traditional approaches rely on alignment algorithms that generally adapt at least linearly to the size of the query and the database [9]. This temporal complexity is unable to keep up with the current size and exponential growth rates of current protein databases, for example, methods based on K-mer [12] and Profile Hidden Markov Model (pHMM) [9].

Hence, artificial intelligence, a field established in the 1950s [13] but experienced a new era in recent years, gives rise to many hopes in various fields [14], thanks to new algorithms and the multiplication of data sets and the tenfold increase in computing power [15]. Artificial intelligence methods, such as machine learning [16] or deep learning [17], have provided new solutions to various problems in biology like prediction of 3D protein structures [18], detection of COVID-19 cases from chest x-ray images [19], [20] or design of new pharmaceutical molecules [21].

Our contribution is to offer a new approach of which artificial intelligence is the pillar. It involves using natural language processing (NLP) [22], [23] with deep learning to determine the function of enzymes. The organization of the remainder of the paper is as follows: The research methodology is described in section 2. Section 3 presents the results and discusses them. The last section concludes this paper and discusses future work.

2. RESEARCH METHODOLOGY

The proposed approach steps are organized in three phases: pretreatment, model learning and prediction (Figure 1). For the pretreatment, the data used was retrieved from the Kaggle website, in the form of two Comma-separated values (csv) files. The files were merged according to the entry to have all the information of an entry on the same record. Then the data was cleaned by deleting the records that were not proteins and then by deleting the entries with the missing protein sequence or classification.

To carry out training, the data must be transformed, that is, the protein sequences and their classes are converted into digital data that can be processed by the convolutional neural network. Then this data is divided into two samples, the first is for training and represents 90% of the data, and the second for the evaluation of the model and represents the remaining 10%. Then, the model is created, trained and evaluated, and the model parameters are modified each time until an adequate result is obtained. Once the optimal model parameters are found, the model is saved. As soon as the model is saved, it can be loaded to make predictions of enzyme classes. The rest of this section describes the steps of each phase.

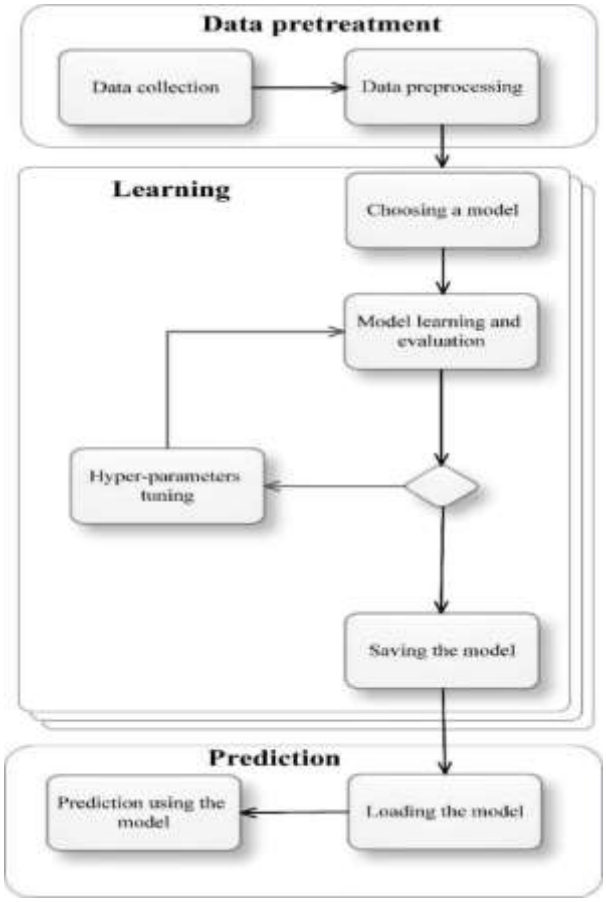

Figure 1. Process of the global approach

2.1. Data pretreatment

a. Data collection

The data used was retrieved from the Kaggle site, which is specialized in data science and deep learning. The source of this data is the protein data bank (PDB) [24]. This data appears in two CSV files and contains many attributes. Those used are: ID, protein sequence and classification.

pdb_data_no_dups.csv contains protein metadata which includes details on protein classification, and extraction methods.

pdb_data_seq.csv contains 346,325 protein structure sequences, as well as other molecules.

b. Data preprocessing

Before learning takes place, the data should be properly preprocessed. It is first loaded, explored and visualized with Pandas. Since the inputs needed for learning should be protein sequences, filtering should be done to remove any other molecules. We select only the necessary attributes for learning in each of the two files and remove all other attributes. Then we join the two datasets in a single DataFrame according to their 'structureId', which represents the entry for each record.

It is necessary to proceed to the deletion of the records with missing values. We end up with 346,321 records containing protein sequences and their classifications, then we exclude:

- Records of proteins not belonging to the six enzymatic classes, we end up with 140,083 records.
- Enzyme's records containing too many unknown AA (X) considering them as a background noise, we end up with 139,637 records.
- Enzyme's records having a size below 30 AA, we end up with 137,314 records.
- Enzyme's records having a size above 1000 AA, we end up with 136,132 records.

Visualization of two graphs: the first graph represents the number of sequences per class, the second represents the number of sequences in relation to their sizes (Figure 2 and Figure 3). The next task of this step consists of the data transformation. It begins with the storage of the enzyme's classes of the DataFrame in a numpy matrix so that they can be digitized and prepared for the deep learning (DL) process, using the LabelBinarizer function of the sklearn library. Then, the enzyme's sequences of the DataFrame are stored in a numpy matrix in order to allow digitizing them and preparing them for the DL process, using the Tokenizer function of the Tensorflow library [25].

At the end of this step, The dataset (136,132 enzymes) should be splitted into two parts, the first represents 90% of the data and will be used to train the model. The second represents the remaining 10% and will be used to test the model and evaluate its accuracy. The train_test_split function of the sklearn library is used to divide the dataset, and the inputs of the two parts are taken in a random way. The number of enzyme sequences that will be used as data for training is 122,518, and the remaining 13,614 will be used for model evaluation.

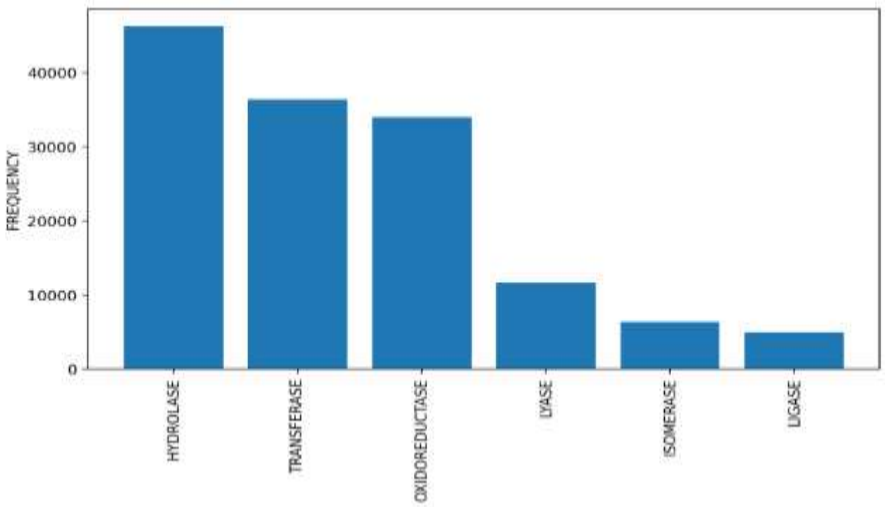

Figure 2. Graphical representation of the number of sequences per enzymatic class

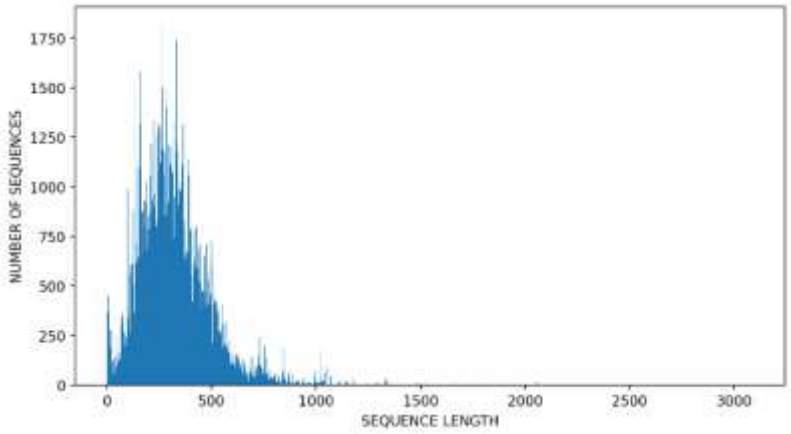

Figure 3. Graphic representation of the number of sequences in relation to their sizes

2.2. Learning

a. Choosing a model

The defined model is a sequential (tf.keras.Sequential), which is a TensorFlow Keras model. It consists of a conv1D layer, dedicated to vector processing and feature extraction, followed by a MaxPolling1D layer which goal is to reduce the size of learned features, consolidating them only to the most essential elements. After the conv1D and MaxPooling1D layers, the learned features are flattened into a long vector and pass through a fully-connected layer before the output layer is used to make the prediction. The fully-connected layer ideally provides a buffer between the learned characteristics and the output in order to interpret the learned characteristics before making a prediction. For regularization, many dropout layers have been used.

The Adam version of the stochastic gradient descent will be used to optimize the network, and the Categorical_crossentropy loss function will be used too since the problem dealt with is a multi-class classification. During its compilation, the model will check if the options chosen are compatible with each other, Table 1 shows the model summary containting all the used layers and the hyper-parameters.

Table 1. The hyper-parameters used to train the model

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 1000, 64)	1664
conv1D	(None, 1000, 256)	262400
dropout (Dropout)	(None, 1000, 256)	0
max_pooling1d (MaxPooling1D)	(None, 125, 256)	0
dropout_1 (Dropout)	(None, 125, 256)	0
flatten (Flatten)	(None, 32000)	0
dropout_2 (Dropout)	(None, 32000)	0
dense (Dense)	(None, 256)	8192256
dropout_3 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32896
dropout_4 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 64)	8256
dropout_5 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 32)	2080
dropout_6 (Dropout)	(None, 32)	0
dense_4 (Dense)	(None, 6)	198
Total params: 8,499,750		
Trainable params: 8,499,750		
Non-trainable params: 0		
None		

b. Model learning and evaluation

The model can now be instantiated and trained using the fit function to initiate training. Once the training is finished, we proceed to the evaluation of the model, i.e. evaluating its accuracy.

c. Hyper-parameters tuning

After evaluating the model, it is time to test the initial parameters of it, i.e. modify these many times, and restart the training and evaluation phases in order to improve the accuracy of the prediction.

d. Saving the model

Once we get the best results by finding the best Hyper-parameters, the model is saved through the function model.save, which is one of the functions that TensorFlow provides.

2.3. Prediction

Once the model is saved, the prediction represents the stage phase where the model becomes functional. First, we start by loanding the model. Then, we use it to predict an unknown class.

a. Loading the model

It is now possible to directly load the model. The Tensorflow load_model function allows fast loading of the model, and shows us the hyper-parameters chosen for the training of the model.

b. Prediction using the model

The model is now ready to be used to predict the class of enzymes. We simply input the enzyme sequence with the unknown class, it gets tokenized the same way our dataset has been tokenized and following the same word index.

3. RESULTS AND DISCUSSION

This section is devoted to the description of the results and a discussion. First, we describe the results by focusing on the precision values and the confusion matrix. After that, we discuss the significance of the results by comparing this work with the related literature.

3.1. Results

To evaluate the efficiency of the model, it is essential to calculate two precision values. The first is the precision of the model training applied to the 90% of the enzymes sample. The second is the precision of the model training applied to the 10% of the enzymes sample left for the evaluation phase to evaluate the enzyme's classification model. It should be noted that the accuracy rates should be as high as possible. After performing the tests, here are the results obtained:

- Train-acc = 0.9933479162245548 ≈ 99 %
- Test-acc = 0.9769355075657411 ≈ 98 %

We used the confusion matrix (which is a matrix that measures the quality of a classification system) to assess the quality of the classifier's output on the enzymes dataset in order to facilitate the reading of the test results obtained. The diagonal line represents the number of points at which the expected classification of enzymes matches the actual classification, while items not belonging to the line are enzymes that are misclassified by the classifier (Figure 4).

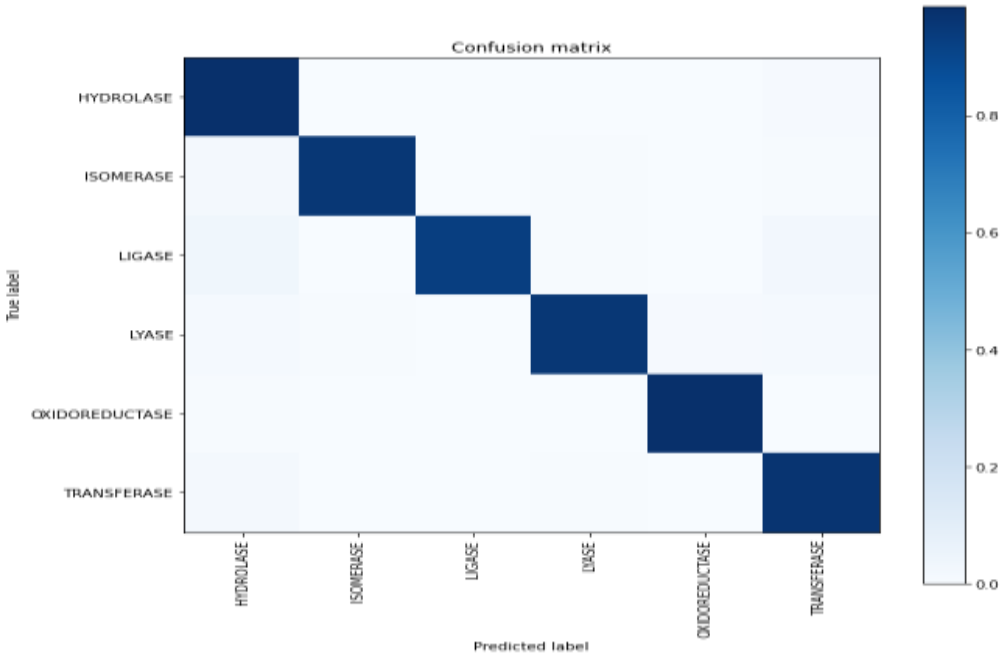

Figure 4. Confusion matrix obtained after the tests

The higher the diagonal values of the confusion matrix get, the better they are, and this proves that the classification was efficient, indicating many correct predictions. At the end of the 100 epochs, we have a precision for the training set 99.33% and 97.69% for the test set. The loss of validation decreases with each epoch (Figure 5 and Figure 6).

The results are represented by the probabilities of the enzymes belonging to one of the six classes. Only values greater than zero are displayed. The greatest probability represents the predicted enzymatic class. For more detailed results of the test, the classification_report, which is a Sklean function, is used to calculate the precision, the recall and the f1-score; it shows also the number of enzymes tested of each class (Table 2).

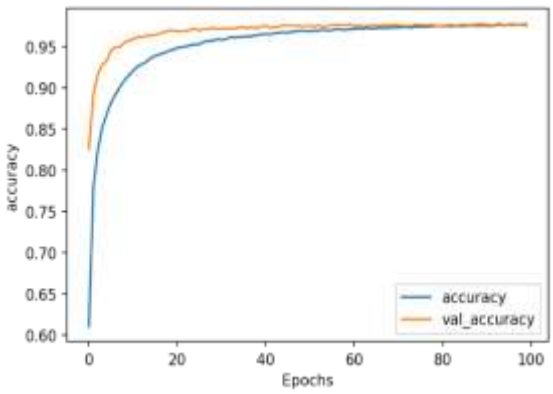

Figure 5. Precision for the first classifier for 100 iterations

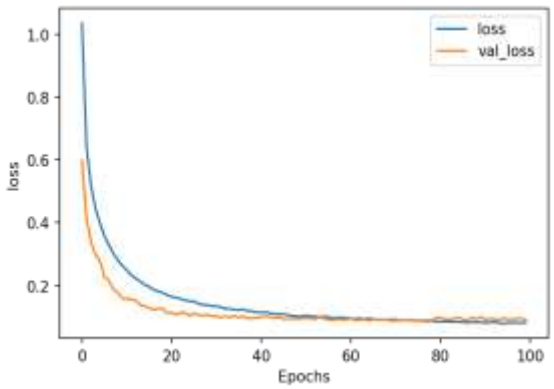

Figure 6. Loss for the first classifier for 100 iterations

Table 2. Classification report showing precision, recall, f1-score and number of enzymes tested of each class

	Precision	Recall	F1-score	Support
HYDROLASE	0.97	0.99	0.98	4474
ISOMERASE	0.99	0.96	0.97	663
LIGASE	0.98	0.93	0.96	471
LYASE	0.96	0.96	0.96	1133
OXIDOREDUCTASE	0.99	0.99	0.99	3373
TRANSFERASE	0.98	0.97	0.97	3500
Accuracy			0.98	13614
Macro avg	0.98	0.97	0.97	13614
Weighted avg	0.98	0.98	0.98	13614

3.2. Discussion

Classical methods are generally deduced from well-motivated methods but still remain heuristic such as the basic local alignment search tool BLAST which searches a database for proteins homologous to a given query protein, via an alignment of multiple sequences, and which subsequently assigns a protein sequence to the function of the most similar protein in its database. Therefore, a more accurate and faster method of predicting proteins classes (enzymes in our case) is needed:

- DL methods, and in particular self-supervised algorithms from natural language processing (NLP), are promising approaches in this direction.
- Alignment-free methods such as our DeepEnz approach, can extract functional information directly from the sequence without the need for multiple alignment.

The work accomplished and the results obtained prove that DL plays an effective role in predictions and classifications, in particular in the classification of enzymes translated in silico. However, on one condition: sequencing and assembling must be done properly because the sequence of a gene is the substrate of functional annotation. This work proposes a concrete solution in order to remedy the problem of functional annotation, which is one of the major problems of bioinformatics.

The most corresponding work that might interrelate with DeepEnz is UDSMprot [26]. This latter exposes the sequences to three levels of models in order to classify the enzymes: Level 0 is devoted to determine whether the protein is an enzyme or not; level 1 is meant to classify enzymes in one of the six classes; the last level (level 2) permits to find out the enzyme's sub-class.

In our approach, while level 0 is considered in pretreatment phase by selecting only enzymes from the PDB dataset, the learning phase focuses on level 1 to determine the class of each enzyme. The result obtained in level 1, that is common in the two approaches, shows that DeepEnz gives a prediction accuracy of 97.69% (≈ 98 %) while UDSMprot obtains 97% (Table 3).

Table 3. Comparison of DeepEnz with UDSMprot

Work	Level of predection	Precision
UDSMprot	Level 1	97 %
DeepEnz	Level 1	98 %

4. CONCLUSION

The emergence of high throughput sequencing techniques has represented a real challenge for the various disciplines related to sequencing. Whether in terms of processing or interpretation, bioinformatics remains the discipline most affected by this advent. One of the challenges is gene annotation, which this work partially supports by predicting the classification of raw protein sequences and particlulary the enzymes. The approach taken by this work is to develop a prediction model based on NLP and DL, to predict raw enzymes classifications. The work accomplished and the results obtained prove that DL plays an effective role in predictions and classifications, especially in the classification of enzymes translated in silico. However, despite the efficiency and accuracy proven by the entire process of prediction, this work remains incomplete, and deserves further research. Thus, the future prospects are: Make use of AI techniques to develop more prediction models in biology domain, Make learning on larger databases; Allow the model to carry out learning on level 0 and level 2 of the enzyme's prediction process; Allow the model to use more data than the enzyme's sequence to make the prediction.

REFERENCES

[1] S. M. Cuesta, S. A. Rahman, N. Furnham, and J. M. Thornton, "The classification and evolution of enzyme function" *Biophysical journal*, vol. 109, no. 6, pp. 1082-1086, 2015, doi: 10.1016/j.bpj.2015.04.020.
[2] K. Tipton, S. Boyce, "History of the enzyme nomenclature system," *Bioinformatics*, vol. 109, no. 6, pp. 34-40, 2000, doi: 10.1093/bioinformatics/16.1.34.
[3] C. Brooksbank, M. T. Bergman, R. Apweiler, E. Birney, and J. Thornton, "The European Bioinformatics Institute's data resources 2014," *Nucleic Acids Res.*, vol. 42, no. D1, pp. D18–D25, Jan. 2014, doi: 10.1093/nar/gkt1206.
[4] See-Kiong Ng and Limsoon Wong, "Accomplishments and challenges in bioinformatics," *IT Prof.*, vol. 6, no. 1, pp. 44–50, Jan. 2004, doi: 10.1109/MITP.2004.1265543.
[5] S. J. Sammut, R. D. Finn, and A. Bateman, "Pfam 10 years on: 10 000 families and still growing," *Brief. Bioinform.*, vol. 9, no. 3, pp. 210–219, May 2008, doi: 10.1093/bib/bbn010.
[6] R. Cao, C. Freitas, L. Chan, M. Sun, H. Jiang, and Z. Chen, "ProLanGO: Protein Function Prediction Using Neural Machine Translation Based on a Recurrent Neural Network," *Molecules*, vol. 22, no. 10, Art. no. 10, Oct. 2017, doi: 10.3390/molecules22101732.
[7] C. Wan and D. T. Jones, "Protein function prediction is improved by creating synthetic feature samples with generative adversarial networks," *Nat. Mach. Intell.*, vol. 2, no. 9, pp. 540–550, Sep. 2020, doi: 10.1038/s42256-020-0222-1.
[8] F. Zhang, H. Song, M. Zeng, Y. Li, L. Kurgan, and M. Li, "DeepFunc: A Deep Learning Framework for Accurate Prediction of Protein Functions from Protein Sequences and Interactions," *PROTEOMICS*, vol. 19, no. 12, 2019, doi: 10.1002/pmic.201900019.
[9] 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, Jul. 2018, doi: 10.1093/bioinformatics/bty275.
[10] M. Kulmanov, M. A. Khan, and R. Hoehndorf, "DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier," *Bioinformatics*, vol. 34, no. 4, pp. 660–668, Feb. 2018, doi: 10.1093/bioinformatics/btx624.
[11] R. D. Sleator and P. Walsh, "An overview of in silico protein function prediction," *Arch. Microbiol.*, vol. 192, no. 3, pp. 151–155, Mar. 2010, doi: 10.1007/s00203-010-0549-9.
[12] A. Zielezinski, S. Vinga, J. Almeida, and W. M. Karlowski, "Alignment-free sequence comparison: benefits, applications, and tools," *Genome Biol.*, vol. 18, no. 1, Oct. 2017, doi: 10.1186/s13059-017-1319-7.
[13] M. Haenlein and A. Kaplan, "A Brief History of Artificial Intelligence: On the Past, Present, and Future of Artificial Intelligence," *Calif. Manage. Rev.*, vol. 61, no. 4, pp. 5–14, Aug. 2019, doi: 10.1177/0008125619864925.
[14] C. Villani *et al.*, "Donner un sens à l'intelligence artificielle: Pour une stratégie nationale et européenne," 2018.
[15] P. Deitel and H. Deitel, "Python for Programmers: with Big Data and Artificial Intelligence Case Studies," Boston, MA: Prentice Hall, 2018.

[16] C. Mishra, and D. L Gupta, "Deep machine learning and neural networks: An overview," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 6, no. 2, pp. 66-73, 2017, doi: 10.11591/ijai.v6.i2.pp66-73.
[17] G. Al-Bdour, R. Al-Qurran, M. Al-Ayyoub and A. Shatnawi, "Benchmarking open source deep learning frameworks," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 5, pp. 5479-5486, 2020, doi: 10.11591/ijece.v10i5.pp5479-5486.
[18] J. Lee, P. L. Freddolino, and Y. Zhang, "Ab Initio Protein Structure Prediction," in *From Protein Structure to Function with Bioinformatics*, D. J. Rigden, Ed. Dordrecht: Springer Netherlands, 2017, pp. 3–35.
[19] L. Wang and A. Wong, "COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images," *ArXiv200309871 Cs Eess*, May 2020, Accessed: Aug. 29, 2020. [Online]. Available: http://arxiv.org/abs/2003.09871.
[20] M. Y. Kamil, " A deep learning framework to detect Covid-19 disease via chest X-ray and CT scan images," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 1, pp. 844-850, 2021, doi: 10.11591/ijece.v11i1.pp844-850.
[21] H. Zhu, "Big Data and Artificial Intelligence Modeling for Drug Discovery," *Annu. Rev. Pharmacol. Toxicol.*, vol. 60, no. 1, pp. 573–589, 2020, doi: 10.1146/annurev-pharmtox-010919-023324.
[22] J. Shruthi and S. Swamy, "A prior case study of natural language processing on different domain," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 5, pp. 4928-4936, 2020, doi: 10.11591/ijece.v10i5.pp4928-4936.
[23] O. J. Ying, M. M. A. Zabidi, N. Ramli, U. U. Sheik, "Sentiment analysis of informal Malay tweets with deep learning," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 9, no. 2, pp. 212-220, 2020, doi: 10.11591/ijai.v9.i2.pp212-220.
[24] S. K. Burley, H. M. Berman, G. J. Kleywegt, J. L. Markley, H. Nakamura, and S. Velenkar, "Protein Data Bank (PDB): the single global macromolecular structure archive," *Protein Crystallography. Humana Press*, pp. 627-641, 2017.
[25] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," *12th {USENIX} symposium on operating systems design and implementation*, pp. 265-283, 2016.
[26] N. Strodthoff, P. Wagner, M. Wenzel, and W. Samek, "UDSMProt: universal deep sequence models for protein classification," *Bioinformatics*, vol. 36, no. 8, pp. 2401–2409, 2020, doi: 10.1093/bioinformatics/btaa003.