Received: August 11, 2021. Revised: September 30, 2021. 642

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

MDPFP-FCNN: Multidomain Protein Function Prediction Using Fuzzy Convolutional Neural Network

Raghad M. Eid[1]* Eman k. Elsayed[2,3] Fatma T. Ghanam[2]

[1]*Palestine Technical University-Kadoori, Tulkarm, Palestine*
[2]*Faculty of science, Al-Azhar University, Cairo, Egypt*
[3]*Canadian International College CIC, School of Computer science*
* Corresponding author's Email: raghad.abushaar@ptuk.edu.ps

Abstract: More than seventy million protein sequences exist now, with just around 1% of their functions known. Multi-domain protein prediction is one of the problems in the Bioinformatics field, which was conducted to find the function of proteins. As a result, the researchers attempt to develop algorithms that predict protein functions based on their sequences. We might think of the tremendous rise in sequencing and structural genomics as providing us with a lot of data to expose the complicated sequence, structure, and functional correlations that exist in proteins. However, because of the critical functions that these macromolecules perform in biological mechanisms, acquiring a thorough understanding of their activity has recently emerged as a major challenge. Furthermore, multi-domain protein function prediction approaches are more efficient than methods that are fully processed based on protein sequencing. In this research, we used a model for predicting multi-domain protein function using a Fuzzy Convolutional Neural Network. In this research, we used a novel hybrid CNN and Fuzzy for sequence labelling, specifically to predict the function of the protein. Our hybrid model managed to outperform recent previous studies in terms of accuracy (95.02%) and performance on the UniProtKB dataset, and they showed significant improvements, in the execution time spent in the prediction process. It's crucial to keep your attention on this strategy. Further, study will be needed.

Keywords: FCNN, Protein function, Multi-domain proteins, Function prediction, UniProtKB.

1. Introduction

Proteins are macromolecules that have a complicated structure made up of 20 distinct types of amino acids and are essential for cellular survival. Protein structure is important in cell biology, pharmacology, molecular biology, and medical science because protein function is strongly connected to protein structure [1]. However, due to the limitations of experimental techniques such as X-ray crystallography and nuclear magnetic resonance, which are both time-consuming and costly, determining protein structure remains a major challenge [2]. The lower cost of sequencing has resulted in a huge quantity of information being accumulated in sequence databanks. This information is completely deposited in universal archives, e.g. UniProt, the public protein knowledgebase (UniProtKB) [3], NCBI Gene Bank [4], and the EMBL Nucleotide Archive for gene sequences [5].

The UniProtKB database contains a vast number of protein sequences as well as complete descriptions [6]. As much annotation information as is reasonable is added in addition, the fundamental information needed for every UniProtKB entry (chiefly, amino acid arrangement, protein designation or description, taxonomical information, and citation info). To understand this data, the archived sequences should be carefully described with regard to their function and evolution characteristics or qualities. The biological complexity of organisms makes identifying the roles of genes and gene products a difficult task.

However, there are a variety of efforts out there that aim to regularize the explanation of biological

sequence efficient characteristics by requiring the use of restricted expressions. Gene Ontology is the most exhaustive and conventional method of protein investigation (GO) [7]. GO uses a directed acyclic graph structure (DAG) [8] that certainly defines the functions from generic to specific in three classifications: molecular function, biological process, and cellular component [9].

Although the finding of these proteins' special attributes is considered a critical phase forward in exploratory recognition, biological investigation of these proteins is both time-consuming and costly. This led to the development of several computational approaches based on similarity in order to predict unknown characteristics of these proteins depending upon similarity when referring to proteins that have been characterized by experimentation. Sequence alignment might be the uppermost method used (BLAST) [10, 11]. Protein sequence alignment is the technique of finding evolutionary or structurally related locations in a series of amino acid sequences. This approach is useful when there is a relationship in terms of the sequence's recognizable evolutionary history. This is something that has to be observed and addressed [12].

Proteins are built up of biological units known as domains [13], which can fold and develop independently [14], and influence the protein's function. Multiple domains are found in more than 80% of eukaryotic and 67% of prokaryotic proteins [15].

Each domain of a multi-domain protein performs a different function, which is typically linked. It's crucial to study the correlation among protein domains in order to figure out how domain groupings encode complicated functionalities. Therefore, another strategy is to identify evolutionarily conserved sequence segments (e.g., motifs and domains) and correlate these arranged segments with particular tasks [16-18].

Many biological database systems are focused on the finding of purposeful domains and the classification of related protein sequences into categories. The database systems generate functional annotations for domains and families, with some of the most often used domains in sequence/family databases are: Pfam [12, 19], PROSITE [20], HAMAP [21], SUPERFAMILY [22] and InterPro [23, 24]. All of the previous database systems are integrated into Inter-Pro, which also provides a thorough categorization of those proteins.

What may be added to this is that a protein's function is a sole character arising from the involvement of all building blocks, rather than the sum of functions of separate domains [25]. As a result, the concept of domain architectures/arrangements (DA) was represented, which is defined as a protein's organizational characteristics in relation to the domains it contains. Domain material, linear command of domains in a protein arrangement, and recurrence of domains in proteins are examples of these characteristics. It's also useful to note that in DA-based techniques, the characteristics mentioned previously in this context are used to determine key commonalities between tried proteins [26].

In this paper, we proposed a model for predicting multi-domain protein function using a Fuzzy Convolutional Neural Network (FCNN). FCNN is used to extract and enrich the features, bringing the data into higher layers and retrieving important features. We also used the FCNN to capture long-range interdependencies between each residue in sequences. However, there is a disadvantage when using FCNN for sequence labelling tasks. The convolution and the pooling process of the FCNN will cause the sequence length to be reduced. Therefore, in this study, we implemented the Multilayer Shift-and-Stitch technique [27] so that the length of the sequences does not decrease after the convolution and pooling stages.

Afterwards, we exploit the capability of FCNN to classify data with high dimensionality, as it is safe to increase the number of features that will be fed into the FCNN because the regularization parameter of FCNN will decide which of these features are impactful and which are not. The model managed to capture the relationship between each sequence's residue and increase the data features, while the FCNN showed better performance when replacing the dense layer in classifying high dimensional data.

The model processes the large feature map data generated by FCNN and predicts secondary protein structure labels for each position in the sequences. This research is an extended version of the authors' thesis [26, 28, 29].

In this paper, we introduced specific contributions as follows: (1) A new architecture for predicting multi-domain protein function using a Fuzzy Convolutional Neural Network. (2) The model (MDPFP-FCNN,) managed to outperform FunFHMMer [28], InterPro2GO [29] and UniProt-DAAC [26] in terms of accuracy and performance. (3) Our model was able to identify new functions of the protein and define its domain, which is a novel contribution according to critical assessment of functional annotation (CAFA) [30-33].

We organized this paper as follows: Chapter 1 explains the problem of recognizing the functions of proteins in multi-domains. Chapter 2 shows the numbers of related studies. Chapter 3 explains the architecture proposed in this paper. Chapter 4 explains the experimental environment. Chapter 5 explains the Evaluation measures, Chapter 6 explains the results of this research and discusses the findings. Chapter 7 shows the conclusion and possibilities for future work.

2. Related works

In FunFams, Rentzsch and Orengo (2013), employed domain families to predict the function of the whole protein. This approach combines sequence clustering with supervised cluster assessment, based on the available (GO) annotation data in the following phase. It organizes domain sequences into families based on the GO annotations of their parent proteins [34], We differ in our proposal in the methodology of the entire work, but we used some of the processes mentioned in the previous special study to address the annotation.

Teng et al. (2014), introduced the SeekFun approach, which tries to predict protein function using a weighted mapping of domains and GO terms. They use resident domains of proteins and protein-level GOA as cues to annotate proteins instead of utilizing the amino acid sequence directly [35]. Also, here we differ in our proposal in the methodology of the whole work because the previous study relied only on protein-level GOA as a cue to annotate proteins.

Wang Et Al. (2018) incorporate a domain architecture inference method based on Bayesian statistics that assesses the likelihood of having a GO term and predicts protein functions in the form of Gene Ontology (GO) terms [36]. Here they used statistics as the mainstay of their prediction.

In 2015, Piovesan et al. (2015) in INGA created a web server in INGA (29) to predict protein function using a mixture of three approaches: sequence similarity, domain architecture searches, and integrated protein-protein interaction network (PPI) data to provide general predictions for GO keywords using functional enrichment [37], Here they used more than one method in the prediction process, and the results were good, but the special factors in time and performance were bad, and this is what distinguishes us from them. We have studied and developed the performance and accuracy factors.

Das et al. (2015) used FunFHMMer web server, which provided Gene Ontology (GO) annotations for query protein sequences based on the functional classification of the domain-based CATH-Gene3D resource, and the server also provided valuable information for the prediction of functional sites [28], We disagree with this study on the entire methodology because it is based on the functional classification of the domain-based CATH-Gene3D resource.

Doğan, et al. (2016) used a novel approach (UniProt-DAAC) in the field of automatic functional annotation of protein sequences with the alignment and classification of domain architectures (DAs). (1) where they used DAs as the basis of a similarity measure between proteins to propagate GO annotation; (2) the employment of multi-label classification where each class represents a unique GO term, thus enabling the optimization of the parameters for each term independently and (3) they used InterPro as the domain resource in order to increase the coverage of domain annotation on the proteins [26]. We used the same special steps for protein sequencing and differed in the methodology.

In SDN2GO, Cai et al., 2020 presented a deep-learning-based integrated classification model that predicts protein functions. Convolutional neural networks are used to train and extract features from sequences, domains, and PPI networks, and then a weight classifier is used to combine these characteristics and provide exact GO expression predictions [38]. We disagreed with the algorithms used and the idea of basic research.

FunSite is a machine learning model that uses characteristics obtained from protein sequence and structure, as well as evolutionary data from CATH functional families, to identify catalytic, ligand-binding, and protein–protein interaction functional sites (FunFams) [39], We used some of the evaluation methods used for this study.

Finally, our work aimed to use a model of multi-domain protein function prediction using a fuzzy convolutional neural network. The MDPFP-FCNN prediction approach can provide functional annotations for over 19 million UniProtKB and Ensembl domain sequences. It is critical to focus on this technique, which is designed to complement rather than replace traditional sequence-based strategies.

We differ from all the studies mentioned above and also the ones mentioned in the introduction chapter in the methodology and the modernity of the dataset. We agree on some necessary processes in the processing of basic sequences, we expect great progress in the results through our model, especially after documenting the functions during the previous years.

3. Methods

Fig. 1 illustrates the general framework design for multidomain protein function prediction using Fuzzy Convolutional Neural Network, with the suggested mechanism of action.

3.1 Data representation

The steps of schematic data representation will be explained below.

1- Input Protein Sequence (Unknown): We enter the protein insertion sequence (unknown) without any prior knowledge of its length or preconditions.

2- Domains Identification by Creation of Domain Architectures (DAs): The Multi-Domain-Benchmark findings for UniProtKB proteins are used to build domain architectures (DAs). The Multi-Domain-Benchmark integrates protein attribute information from 25 distinct consortium member directories. This data is made up of sequence signatures that are responsible for the protein's specific characteristics. Manual curation is used to combine signatures from various member databases into unique entries in Multi-Domain-Benchmark, where we have a particular clarification in Section 3.1 Dataset.

3- Weighted domain architectures (DAs) matching: 50% of the mismatch value for gap openings and 50% of the gap opening value for gap extensions.

4- Segmentation of domains: The rule of Multi-Domain-Benchmark is used to segregate domains.

3.2 Feature extraction

5- Assignment of Domains to Gene Ontology (Go) Terms: Furthermore, this GO term prediction reference data mainly includes DAs designed for protein elements by UniProtKB-Swiss-Prot (v2020/12) as well as the related GO description (by trial validation codes) from the Uni-Prot-GOA record system [26]. Evidence codes that are generally categorized "experimental" in the GO system (codes: EXP, IDA, IPI, IMP, IGI, and IEP) are considered to be the greatest value and dependability. Annotations are enlarged to encompass all parents of items following the mining of the UniProt-GOA dataset, allowing root (top level) terms to be removed from all GO classifications [40].

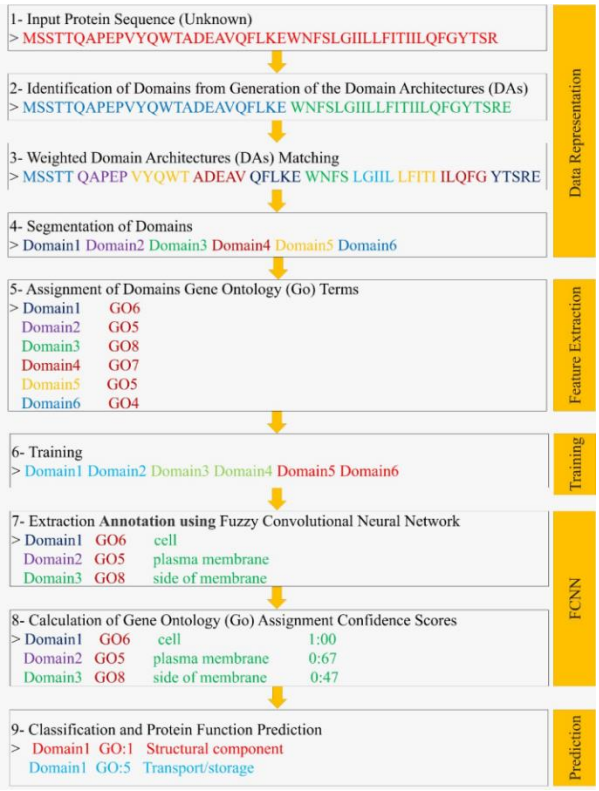

Figure. 1 General framework architecture

3.3 Training and fuzzy convolutional neural network:

As described in Section 3.2 FCNN Architecture, some processes are represented below; training, Fuzzy Convolutional Neural Network, and test proteins.

1- Extraction of Annotation using Fuzzy Convolutional Neural Network: This phase is critical because the protein's annotation is extracted since it comprises several items, the most significant of which are the processes and functions that will be predicted later.

2- Calculation of Gene Ontology (Go) Assignment Confidence Scores: Nodes are arranged according to Confidence Scores.

3- Classification and Protein Function Prediction: Following classifying each protein by domain, our model predicts protein activities depending on the relevant processes.

3.4 Dataset

As shown in the image below, we used Multi-domain-Benchmark (MDB), a database suite comprising 412 curated multi-domain queries and 227,512 target sequences, representing at least 5108 species and 1123 phylogenetically diverse protein categories, as well as their relevance interpretation and domain placement [1].

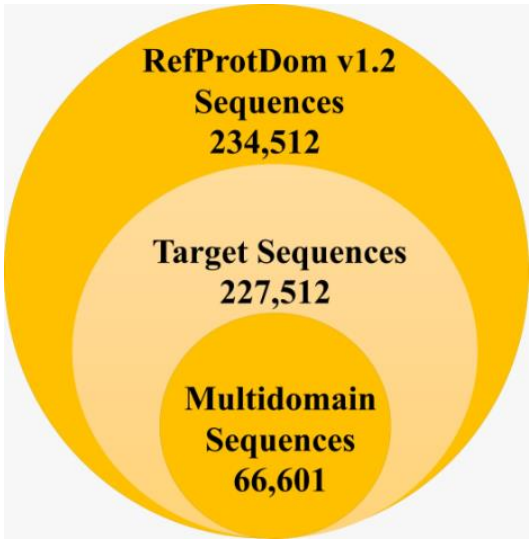

Figure. 2 Domain architecture (DA)

Most significantly, Multi-Domain-Benchmark is regarded as a comprehensive evaluation suite for genetic databases of search algorithms. It's from RefProtDom v1.2 [41], which is based on Pfam [42]. Furthermore, Multi-Domain-Benchmark is designed to offer a comprehensive evaluation of genomic databases while searching for query sequences that comprise several domains. Please keep in mind that benchmarks have target databases related to multi-domains [41], and the Multi-domain-Benchmark is considered as the only benchmark known to the authors containing at least 100's of query sequences with multi-domains.

Furthermore, we employ protein sequences from UniProtKB/Swiss-Prot (Reviewed) (v2020/12) along with their empirically validated GO annotations to show that the study results have potential use for protein function estimation[40]. We also employ the UniProtKB/TrEMBL(Unreviewed) (v2020/12) approach to offer functional predictions for protein entries in the database [43-45].

In the GO system, the evidence codes EXP, IDA, IPI, IMP, IGI, and IEP are regarded as having the highest quality and reliability. Following the mining of the UniProt-GOA dataset, comments are enlarged to cover all roots of characters found, leaving all GO classifications without root (top level) terms [40].

3.5 FCNN architecture

A number of aspects and points, such as the matrix value, the input, the fuzzication layer, and so on, may help us grasp this science. Extensive practice broadens your comprehension. Perfect practice makes perfect.

It is worth noting that input has been embedded by imbedding the layer to the matrix's real value. The fuzzication layer then transforms the input matrix into the fuzzy domain. This is an essential consideration to make. As a result, the fuzzy representation has been twisted in the fuzzy convolutional layers, which function as filters to extract high-level features from the data. Following the passage of the fuzzy convolutional phase, the recovered feature set is transformed into what is known as a crisp value by the defuzzication layer. Finally, the layer with a complete connection serves as an output classification for FCNN.

3.6 FCNN embedding level

FCNN assigns a score to each label. This is a critical point; in order for you to be able to use it, the model accepts input as a sequence and then passes it through the layers of the model. Furthermore, at each layer, features from higher levels have been collected and transmitted to the subsequent layer. The model then extracts characteristics from the word's vector level to the domain's level, which is an important step[2].

Please bear in mind that, owing to appropriate arithmetic using FCNN, characters in the domain should be shown as quantitative data as well. The embedding of the sequence level was the first stage in mapping each character in the sequence to a d-dimensional vector, thus each sequence would be converted into a matrix of size md, where m is the length (number of characters in the sequence) and d is the dimension of the embedded vector. The lengths of the sequences in the dataset are considered to vary. To make things easier, we pad unique characters at the end of sequences for the same length in all sequences. That is a critical element to consider.

Relating to each sequence that does consist of M characters, the $(w_1, w_2, \dots, w_m, \dots, w_M)$, the w_M characters in sequence is going to be pretty fairly transformed into a vector $u_m = u_1, u_2, \dots, u_D$. We have static size V dictionary, then, we use implanting matrix $D \in R^{(d \times |V|)}$. We get mapping of w_M to the vector u by using the Eq. (1).

$$u_m = Dv^m \tag{1}$$

Where v^m is vector dimension $|V|$ that has a value of 1 in the index w and 0 in remaining positions. w is regarded to be the index of character w_m in dictionary V. First and foremost, matrix D has been randomly initiated, after that, its values have been considered to be trained during training of model. This also should be noted as well.

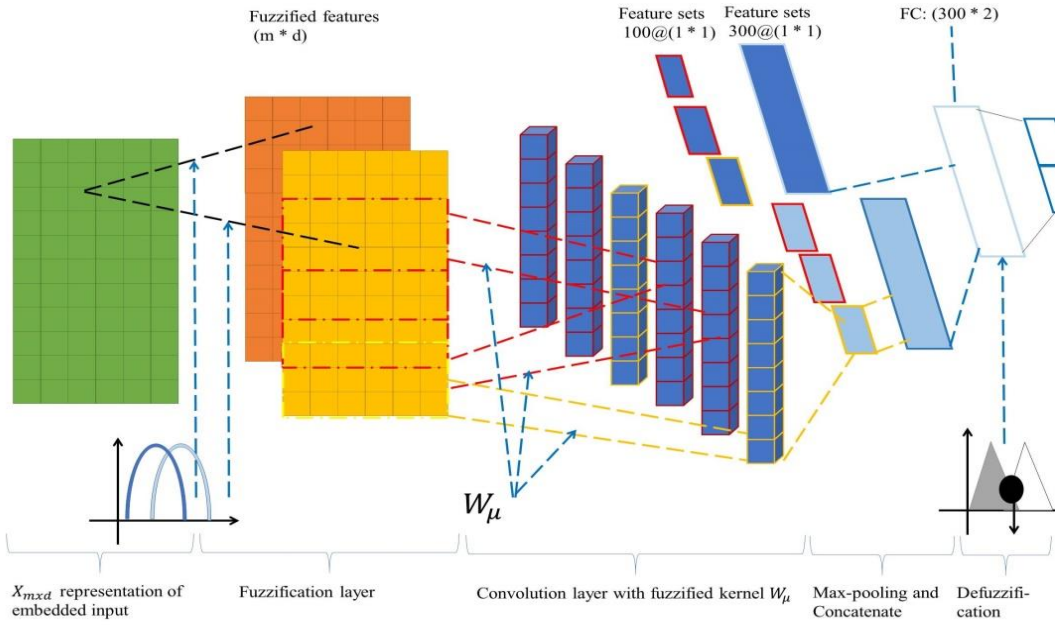

Figure. 3 FCNN architecture

3.6.1. FCNN classifier level

Each element in the input matrix X has been allocated a range of linguistic depending on membership functions following the embedding of each input into matrix X. The fuzzification function returns a grade that represents the membership of an input node in a fuzzy collection. The fuzzy groups $\tilde{X}$ in Eq. (2) have been given by Eq. (3), which is calculated by using the maxproduct operation. What is more, they also have the possibilities of the input as well as the output data relating to pre-defined reference of the fuzzy numbers $\widetilde{MF}_{ij}$ in the universe of discourse. That is clearly noted as you deal with that.

$$\check{X} = fuzzification\left(x_{ij}\middle|cx_{ij}\right) \tag{2}$$

Where i, j is regarded to be guides of element x in the input matrix X, as well as the center of the input the function of the fuzzy membership cx.

$$x_{ij} = possibility\left(x_{ij}\middle|\widetilde{MF}_{ij}\right)$$

$$= max_{x\in X}\left(\widetilde{MF}_{ij}\delta\left(x - x_{ij}\right)\right) \tag{3}$$

Where $\delta\left(x - x_{ij}\right)$ is considered to be the Kronecker delta function.

Please note that each layer of the fuzzy convolution tends to have 3 processing phases, specifically the phase of the fuzzy convolution, nonlinearity phases, as well as pooling phase as well. The phase of the fuzzy convolutional has been a process of applying the filters of the fuzzy convolution to the initial 2D data, as appears in Eq. (4) in which fuzzy convolutional filters W_μ have been calculated as Eq. (6), with W is original convolution filter as well. This is also something very important to be referred to.

$$x_{ij} = \sum_{a=0}^{m-1}\sum_{b=0}^{d-1} W_\mu x_{(i+a)(j+b)} \tag{4}$$

$$W_\mu = fuzzification(W) \tag{5}$$

Furthermore, Eq. (6) is completely considered to be a non-linear transformation of a fuzzy convolution phase output. After the feature extraction phase, the last step is considered to be another operation named or known as pooling (for example, Max Pooling), which is a summary statistic of adjacent outcomes. This phase tends to bring about support to representation, which is defined as a matter that is invariant to input translation, as well as the size of the input for next layer of the fuzzy convolution that may lowered in some way. Please keep that in mind.

$$y_{ij} = \sigma\left(x_{ij}\right) \tag{6}$$

where σ (.) is regarded as convolution layer initiation function.

Furthermore, the layer that has a full connection to the FCNN that is already acting as a classifier with the input characteristics that are crisp value

z_i gained as well from deffuzzification procedure, with the method of gravity center in Eq. (7), where C_y is medium of the function of defuzzification membership. $\hat{y}_i$ is output of classifier as well as W_{fc} is weight matrix of the layer of the full connection.

$$z_i = defuzz(x_i) = \frac{\sum C_y x_i}{\sum x_i} \quad (7)$$

$$\hat{y}_i = W_{fc} z_i \quad (8)$$

3.6.2. FCNN training

Cross entropy is a loss function that is entirely utilized to measure yield error, as shown in Eq. (9), where , $\hat{y}$ is the target, y is the output of the classifier, and N is the number of samples. This is really useful in understanding.

$$E = -\frac{1}{N}\sum_{n=1}^{N}[y_n \log(\hat{y}_n) + (1 - y_n) \log(1 - \hat{y}_n)] \quad (9)$$

Furthermore, the model parameters have been trained by having conventional back-propagation of the learning algorithm with the loss function of the cross entropy as well.

The weight update appears in Eq. (10).

$$W_{fc}(k+1) = W_{fc}(k) - \alpha_{fc}\frac{\partial E}{\partial W_{fc}} \quad (10)$$

Moreover, the centers $C_y(k)$ of the functions of the defuzzification membership have been restructured in Eq. (11), where a_{cy} represents the knowledge rate of updating center, y_{k+1} and $\hat{y}_{k+1}$ are, the output target as well as actual output of the model as well. Please take that into account. That is extremely important.

$$C_y(k+1) = C_y(k) + a_{cy}\nabla c_y \quad (11)$$

Additionally, center value C_w as well as variance σ of the function of the fuzzification category of convolution phases weight have been certainly estimated by using Eqs. (12) to (15) with the learning rate α_{C_w}

$$C_w(k+1) = C_w(k) + \alpha_{C_w}\nabla_{W_\mu} \quad (12)$$

And

$$\sigma_{C_w}(k+1) = \sigma_{C_w}(k) + \sigma_{C_w}\nabla_{W_\mu} \quad (13)$$

Where,

$$\delta_k = (W_{\mu k})^T \delta_k^{(3)} f'(x_k) \quad (14)$$

$$\nabla_{W_{\mu k}} = \sum y_{ij} \times rot180(\delta_k) \quad (15)$$

Eqs. (16) and (17) may be used to update the mean and variance of the fuzzification association function of the layer, where α_{Cx} is the learning rate of the fuzzification layer as well. Concentrate on this issue as well, as it is critical that it be discovered.

$$C_x(k+1) = C_x(k) + \alpha_{Cx}\nabla_{C_x} \quad (16)$$

$$\sigma_{C_x}(k+1) = \sigma_{C_x}(k) + \sigma_{C_x}\nabla_{C_x} \quad (17)$$

3.6.3. Algorithm

The experimental test of FCNN with back propagation in the method may be regarded fairly simple. With regard to the testing data with feature set X and target y, we can simply apply the mini-batch training procedure to achieve the finest FCNN and CNN parameters ever [2].

Please keep in mind that the hyperparameters such as learning rates, dropout rates, batch size, and training epoch have all been empirically chosen. That is a critical issue. Please pay attention to it since knowing a lot about such hyperparameters is really useful.

4. Experimental environment

We tested our idea in a real-world setting, using a cloud server with the following specifications: -

- CPU: (64 core)
- RAM: 512 GB
- GPUs: NVIDIA Pascal X (8GB×4)

5. Evaluation measures

We employ two metrics to assess the effectiveness of our model: F max (protein centric maximum) and AUC (area under the curve) (area under the precision-recall curve). These two measures are employed in the CAFA challenge [3-6]. We calculate the F_{max} measure using the standard formulas provided by CAFA:

$$F_{max} = max_t\left\{\frac{2.AvgPr(t).AvgRc(t)}{AvgPr(t)+AvgRc(t)}\right\} \quad (18)$$

Where $Pr(t)$, $Rc(t)$ are the precision and the recall of the threshold t ∈ [0, 1]. $AvgPr(t)$ is average precision over the proteins where at least

one GO term is predicted. They are calculated using the following formulas:

$$AvgPr(t) = \frac{1}{m(t)} . \sum_{i=1}^{m(t)} pr_i(t) \quad (19)$$

$$AvgRc(t) = \frac{1}{n} . \sum_{i=1}^{n} rc_i(t) \quad (20)$$

Such that m(t): the total number of proteins, n: the number of proteins in the target data test set.

$pr_i(t)$, $rc_i(t)$ are the precision and recall of some protein i using threshold t, they are calculated as follows:

$$pr_i(t) = \frac{\sum_f I(f \in P_i(t) \wedge f \in T_i)}{\sum_f I(f \in P_i(t))} \quad (21)$$

$$rc_i(t) = \frac{\sum_f I(f \in P_i(t) \wedge f \in T_i)}{\sum_f I(f \in T_i)} \quad (22)$$

For the second measure, we calculate precision, recall, accuracy, and Matthew's correlation coefficient (MCC) for model of the Go using the following formulas:

$$(TPR)\ Precision = \frac{TP}{TP+FP} \quad (23)$$

$$(TPR)\ \text{HYPERLINK https://en.wikipedia.org/wiki/Precision_and_recall\ l RecalloPrecision and recallRecall} = \frac{TP}{TP+FN} \quad (24)$$

$$(FPR)\ false\ positive\ rate = \frac{FP}{FP+TN} \quad (25)$$

$$(ACC) Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (26)$$

Such that TP: the number of true positives, FN: the number of false negatives, FP: the number of false positives, and TN: the number of true negatives.

Matthew's correlation coefficient (MCC) is calculated as follows:

$$MCC = \frac{TP.TN - FP.FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (27)$$

6. Results and discussions

Proteins are macromolecules with a complicated structure that comprise amino acid building blocks and play an important function in the survival of our cells [7]. Knowing the precise structure of a protein is critical for understanding its activities and role in biological processes. Protein activities are, in fact, closely connected to their structure [8]. The discovery of X-ray crystallography revealed the three-dimensional structure of protein [9]. Furthermore, because the number of protein sequences is continually rising, the great majority of proteins can only be annotated computationally. In this study, a Fuzzy Convolutional Neural Network to create a model for predicting multi-domain protein function is used. Protein function estimate utilizing a Convolutional Neural Network (CNN) based on gene ontology on a nearly explored data set with small size proteins. They are looking for connections between protein properties and activity [10]. We attempted to maximize the model's potencies in this study by fine-tuning, increasing the size of the features, and replacing the thick layers.

For GO annotation of the full database, the method is applied to about 220 million protein items in UniProtKB/TrEMBL. These findings show that the method is both beneficial and successful, allowing for the discovery of efficient multi-domain protein interactions. For all UniProtKB records, DAs are now created and saved in the UniProt domain architecture database with each release. Furthermore, DAs have undoubtedly been created for all UniProtKB entries at each release and preserved in the UniProt Domain Architecture Database.

Table No. 1 shows statistics for the DA generation process in relation to the UniProtKB/Swiss-Prot and UniProtKB/TrEMBL databases (v2020 12). As indicated in Table No. 1, DAs have been utilized to cover 96% of UniProtKB/Swiss-Prot entries and 76% of UniProtKB/TrEMBL entries. According to Table No. 1, the number of unique DAs produced in UniProtKB/Swiss-Prot accounts for 86 percent of the total number of entries with domain discoveries. The rate for the UniProtKB/TrEMBL is just 15%, which is most likely owing to a higher level of redundant matter in the UniProtKB/TrEMBL compared to the UniProtKB/Swiss-Prot. This is regarded as one of the most critical issues that we should be aware of.

MDPFP-FCNN, FunFHMMer [11], InterPro2GO [12], and UniProt-DAAC [13] are all compared and contrasted in Table 2. In terms of performance and accuracy, as well as the ability to anticipate greater rates, our MDPFP-FCNN approach beat the other research. In comparison to the other research, the findings in Table 2 demonstrate that our model has a strong capacity to create a significant number of unique entries, unique GO words, and GO terms predicted by each system. Furthermore, our model identified the number of

Table 1. DAs generation statistics on the UniProtKB Dataset

Dataset UniProtKB	Swiss-Prot (2020-12)	TrEMBL (v2020-12)
No. of input protein entries:	565254	219174961
No. of entries with InterPro domain hits: (Family and domain databases)	546292	166656803
No. of unique DAs generated:	16478	7942841

Table 2. Comparison of statistics and performance between MDPFP-FCNN, FunFHMMer, InterPro2GO and DAAC

	MDPFP-FCNN	InterPro2GO [12]	FunFHMMer [11]	UniProt-DAAC [13]
Total no. of mappings	18631	6382	22462	25626
No. of unique entries	9481	2927	7951	8248
No. of unique GO terms	791	1411	658	778
No. of GO terms predicted by each system	698	1188	489	555
	(No. of shared terms: 218)			
No. of mapped GO term relations with the other system	547 in relation 143 independent	760 in relation 651 independent	667 in relation 159 independent	625 in relation 153 independent

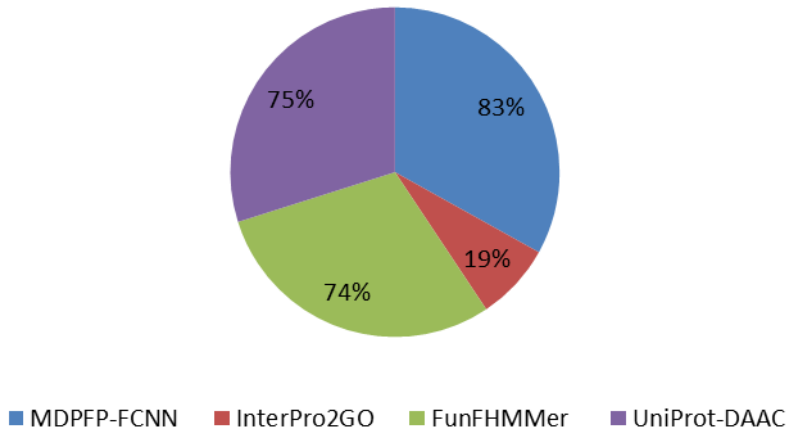

Figure. 3 Rate of Specificity of the mapped GO terms compared to other system

mapped GO word relationships with the other system and shown superior understanding of GO terms when compared to the other research.

The findings also revealed that the study's model has an 83% specificity of the mapped GO keywords when compared to another system (see Fig. 4).

We also compared the Accuracy, F-score, Recall, Precision, and FPR (fall-out) ratios. As demonstrated in Table 3, our MDPFP-FCNN model outperformed multi-domain protein functions in prediction accuracy. The study model's accuracy and performance were substantially higher than 95.02%.

Fig. 6 shows a few examples of results for character count, domain count, execution time, and accuracy. Prediction execution time increases as the number of characters and domains in the provided sequence grows, according to the findings we obtained.

The performance based on the standard execution time and the CPU usage rate are examined in the study. The findings revealed that increasing the sequence's domain count resulted in an increase in the prediction's execution time, as shown in Fig. 6.

It was concluded from comparisons with prior studies that when the number of proteins increases

in the next years, there would be a new problem in finding and forecasting functions, due to the insane rise in the connections between proteins and functions.

Therefore, the relationship between the size of the sequence and the time it takes for our model to predict jobs and the percentage of accuracy in identifying them are measured. Accordingly, a significant relationship with the increase in the size of the sequence and also the increase in the number of ontology terms over time, the results change.

We evaluated and compared our work against the CAFA challenge of Standardization of Evaluation Standards.

Table 3. Accuracy and performance comparison between MDPFP-FCNN, FunFHMMer, InterPro2GO, and UniProt-DAAC

	Accuracy (%)	F-score	Recall	Precision	FPR (fall-out)
MDPFP-FCNN	95.02%	0.921	0.887	0.962	4.04×10^{-4}
InterPro2GO [12]	75.32%	0.675	0.615	0.909	1.98×10^{-5}
FunFHMMer [11]	85.14%	0.861	0.854	0.901	4.19×10^{-5}
UniProt-DAAC [13]	90.45%	0.874	0.843	0.919	4.57×10^{-4}

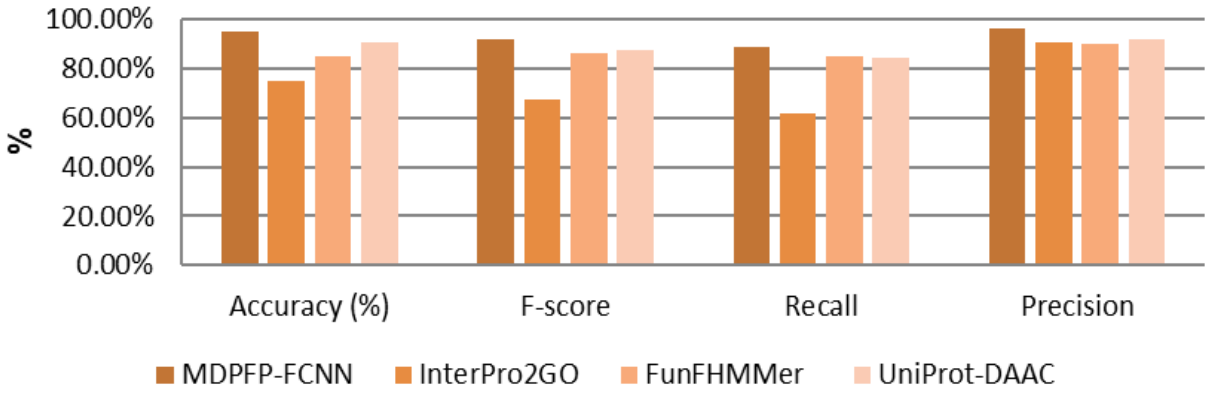

Figure. 4 Accuracy and performance comparison between MDPFP-FCNN, FunFHMMer, InterPro2GO and UniProt-DAAC

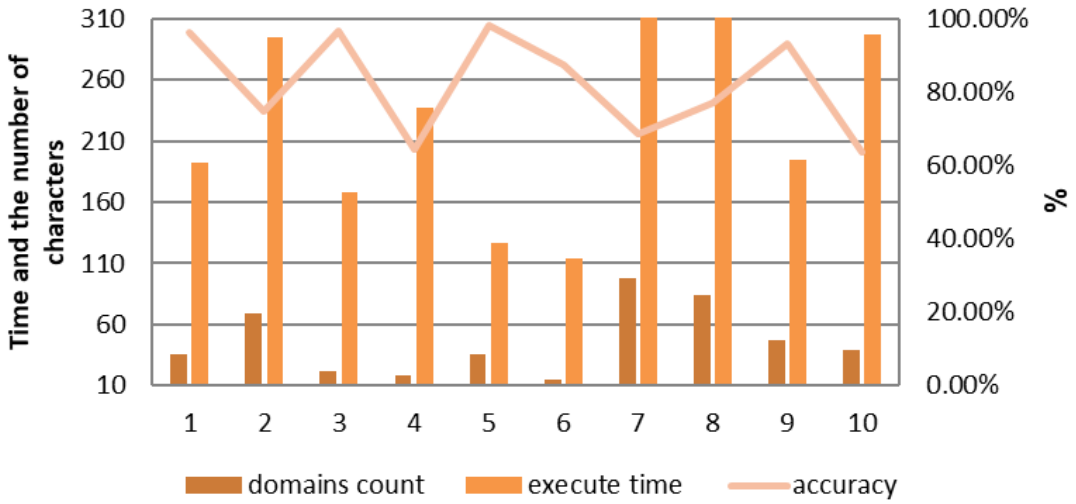

Figure. 5 Sample illustrate domains count and execution time (ms) and, Accuracy of some investigated sequences

7. Conclusion

MDPFP-FCNN is a highly unique technique in the field of automated function identification of protein sequences using alignment and DA classification, which we present in this study. This strategy differs from traditional approaches in the following three ways: To begin, Domain Architectures (DAs) were used as the foundation for a very similar point of measure amongst proteins in order to propagate GO annotation as well; Second, using a multi-label categorization where each class tends to reflect a specific GO term, allowing for parameter optimization of each term separately as well as overall. Finally, employing the Multi-domain Benchmark (MDB) as a domain resource to expand protein domain annotation coverage (more points have been clarified in supplemental information). We used a model for predicting multi-domain protein function using a Fuzzy Convolutional Neural Network. In this research, we have a novel hybrid CNN and Fuzzy for sequence labelling, specifically to predict the function of the protein. MDPFP-FCNN has enabled the connection of Domain Architectures (DAs) with operational terms (each represented by a distinct category) as well as the rapid annotation of non-annotated proteins with similar architectures. Using multi-label categorization allows the protein (as well as its DA) to belong to several classes and functional annotations. That is, in fact, a significant point. Setting up distinct classifiers for each GO term also allows for the selection of different parameters for each class. As a result, we may optimize the Domain Architecture (DA) of similar points criteria for each class. We should integrate the MDPFP-FCNN technique into UniProt's automatic annotation production pipeline so that we can get enrichment for UniProtKB/automatic TrEMBL's annotation function as well. We intend to use the technique to predict additional kinds of annotations, including recommended protein names, subcellular locations, keywords, comments, and features. Finally, our hybrid model managed to outperform recent previous studies in terms of accuracy (95.02%) and performance on the UniProtKB dataset, and it showed significant improvements in the execution time spent in the prediction process. It's crucial to keep your attention on this strategy. Further study will be needed.

Conflicts of Interest

"The authors declare no conflict of interest."

Author Contributions

Conceptualization, Eman K. Elsayed gave the idea of the paper, Raghad M. Eid; software and designed the experiments, Eman K. Elsayed, and Raghad M. Eid; formal analysis, investigation, resources, data preparation, Raghad M. Eid; writing original draft preparation, Fatma T. Ghanam; writing review and editing, Fatma T. Ghanam and Raghad M. Eid; supervised the study, analyzed the results, and verified the findings of the study.

References

[1] I. Dubchak, I. Muchnik, S. R. Holbrook, and S. H. Kim, "Prediction of protein folding class using global description of amino acid sequence", *Proceedings of the National Academy of Sciences*, Vol. 92, No. 19, pp. 8700-8704, 1995.

[2] J. Cheng and P. Baldi, "A machine learning information retrieval approach to protein fold recognition", *Bioinformatics*, Vol. 22, No. 12, pp. 1456-1463, 2006.

[3] A. C. Papageorgiou, N. Poudel, and J. Mattsson, "Protein Structure Analysis and Validation with X-Ray Crystallography", *Protein Downstream Processing: Springer*, pp. 377-404, 2021

[4] T. U. Consortium, "UniProt: the universal protein knowledgebase in 2021", *Nucleic Acids Research*, Vol. 49, No. D1, pp. D480-D489, 2020.

[5] K. Clark, I. K. Mizrachi, D. J. Lipman, J. Ostell, and E. W. Sayers, "GenBank", *Nucleic Acids Research*, Vol. 44, No. D1, pp. D67-D72, 2016.

[6] R. Leinonen, R. Akhtar, E. Birney, L. Bower, A. C. Tárraga, Y. Cheng, I. Cleland, N. Faruque, N. Goodgame, R. Gibson, G. Hoad, M. Jang, N. Pakseresht, S. Plaister, R. Radhakrishnan, K. Reddy, S. Sobhany, P. Ten Hoopen, R. Vaughan, V. Zalunin, and G. Cochrane, "The European Nucleotide Archive", *Nucleic Acids Research*, Vol. 39, No. Database issue, pp. D28-D31, 2011.

[7] The UniProt Consortium, "UniProt: the universal protein knowledgebase", *Nucleic Acids Research*, Vol. 45, No. D1, pp. D158-D169, 2016.

[8] "The Gene Ontology resource: enriching a GOld mine", *Nucleic Acids Research*, Vol. 49, No. D1, pp. D325-D334, 2021.

[9] J. Zhu, Q. Zhao, E. Katsevich, and C. Sabatti, "Exploratory gene ontology analysis with interactive visualization", *Scientific reports*, Vol. 9, No. 1, pp. 1-9, 2019.

[10] A. Tomczak, J. M. Mortensen, R. Winnenburg, C. Liu, D. T. Alessi, V. Swamy, F. Vallania, S. Lofgren, W. Haynes, and N. H. Shah, "Interpretation of biological experiments changes with evolution of the Gene Ontology and its annotations", *Scientific Reports*, Vol. 8, No. 1, pp. 1-10, 2018.

[11] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool", *Journal of Molecular Biology*, Vol. 215, No. 3, pp. 403-410, 1990.

[12] I. Lobo, "Basic local alignment search tool", *Nat Educ*, Vol. 1, No. 1, 2008.

[13] W. R. Pearson and D. J. Lipman, "Improved tools for biological sequence comparison", *Proceedings of the National Academy of Sciences*, Vol. 85, No. 8, pp. 2444-2448, 1988.

[14] R. D. Finn, A. Bateman, J. Clements, P. Coggill, R. Y. Eberhardt, S. R. Eddy, A. Heger, K. Hetherington, L. Holm, J. Mistry, E. L. L. Sonnhammer, J. Tate, and M. Punta, "Pfam: the protein families database", *Nucleic Acids Research*, Vol. 42, No. Database issue, pp. D222-D230, 2014.

[15] C. Vogel, M. Bashton, N. D. Kerrison, C. Chothia, and S. A. Teichmann, "Structure, function and evolution of multidomain proteins", *Current Opinion in Structural Biology*, Vol. 14, No. 2, pp. 208-216, 2004.

[16] Y. Wang, H. Zhang, H. Zhong, and Z. Xue, "Protein domain identification methods and online resources", *Computational and Structural Biotechnology Journal*, Vol. 19, pp. 1145-1153, 2021.

[17] X. Zhou, J. Hu, C. Zhang, G. Zhang, and Y. Zhang, "Assembling multidomain protein structures through analogous global structural alignments", *Proceedings of the National Academy of Sciences*, Vol. 116, No. 32, pp. 15930-15938, 2019.

[18] T. L. Bailey, M. Boden, F. A. Buske, M. Frith, C. E. Grant, L. Clementi, J. Ren, W. W. Li, and W. S. Noble, "MEME SUITE: tools for motif discovery and searching", *Nucleic Acids Res*, Vol. 37, No. Web Server issue, pp. W202-8, 2009.

[19] T. Doğan and B. Karaçalı, "Automatic identification of highly conserved family regions and relationships in genome wide datasets including remote protein sequences", *PloS One*, Vol. 8, No. 9, p. e75458, 2013.

[20] P. Tompa, N. E. Davey, T. J. Gibson, and M. M. Babu, "A million peptide motifs for the molecular biologist", *Molecular Cell*, Vol. 55, No. 2, pp. 161-169, 2014.

[21] J. Mistry, S. Chuguransky, L. Williams, M. Qureshi, G. A. Salazar, E. L. Sonnhammer, S. C. Tosatto, L. Paladin, S. Raj, and L. J. Richardson, "Pfam: The protein families database in 2021", *Nucleic Acids Research*, Vol. 49, No. D1, pp. D412-D419, 2021.

[22] C. J. A. Sigrist, E. D. Castro, L. Cerutti, B. A. Cuche, N. Hulo, A. Bridge, L. Bougueleret, and I. Xenarios, "New and continuing developments at PROSITE", *Nucleic Acids Research*, Vol. 41, No. D1, pp. D344-D347, 2012.

[23] I. Pedruzzi, C. Rivoire, A. H. Auchincloss, E. Coudert, G. Keller, E. D. Castro, D. Baratin, B. A. Cuche, L. Bougueleret, S. Poux, N. Redaschi, I. Xenarios, and A. Bridge, "HAMAP in 2015: updates to the protein family classification and annotation system", *Nucleic Acids Research*, Vol. 43, No. Database issue, pp. D1064-D1070, 2015.

[24] D. Wilson, R. Pethica, Y. Zhou, C. Talbot, C. Vogel, M. Madera, C. Chothia, and J. Gough, "SUPERFAMILY - Sophisticated comparative genomics, data mining, visualization and phylogeny", *Nucleic Acids Research*, Vol. 37, pp. D380-6, 2008.

[25] A. Mitchell, H. Y. Chang, L. Daugherty, M. Fraser, S. Hunter, R. Lopez, C. McAnulla, C. McMenamin, G. Nuka, S. Pesseat, A. S. Vegas, M. Scheremetjew, C. Rato, S. Y. Yong, A. Bateman, M. Punta, T. Attwood, C. Sigrist, N. Redaschi, and R. Finn, "The InterPro protein families database: The classification resource after 15 years", *Nucleic Acids Research*, Vol. 43, 2014.

[26] A. Mitchell, H. Y. Chang, L. Daugherty, M. Fraser, S. Hunter, R. Lopez, C. McAnulla, C. McMenamin, G. Nuka, and S. Pesseat, "The InterPro protein families database: the classification resource after 15 years", *Nucleic Acids Research*, Vol. 43, No. D1, pp. D213-D221, 2015.

[27] M. Blum, H. Y. Chang, S. Chuguransky, T. Grego, S. Kandasaamy, A. Mitchell, G. Nuka, T. P. Lafosse, M. Qureshi, and S. Raj, "The InterPro protein families and domains database: 20 years on", *Nucleic Acids*

Research, Vol. 49, No. D1, pp. D344-D354, 2021.

[28] M. Bashton and C. Chothia, "The generation of new protein functions by the combination of domains", *Structure*, Vol. 15, No. 1, pp. 85-99, 2007.

[29] T. Doğan, A. MacDougall, R. Saidi, D. Poggioli, A. Bateman, C. O'Donovan, and M. J. Martin, "UniProt-DAAC: domain architecture alignment and classification, a new method for automatic functional annotation in UniProtKB", *Bioinformatics*, Vol. 32, No. 15, pp. 2264-2271, 2016.

[30] Z. Lin, J. Lanchantin, and Y. Qi, "MUST-CNN: a multilayer shift-and-stitch deep convolutional architecture for sequence-based protein structure prediction", In: *Proc. of the AAAI Conference on Artificial Intelligenc*e, Vol. 30, No. 1. 2016

[31] S. Das, I. Sillitoe, D. Lee, J. G. Lees, N. L. Dawson, J. Ward, and C. A. Orengo, "CATH FunFHMMer web server: protein functional annotations using functional family assignments", *Nucleic Acids Research*, Vol. 43, No. W1, pp. W148-W153, 2015.

[32] R. Rentzsch and C. Orengo, "Protein function prediction using domain families", *BMC Bioinformatics*, Vol. 14, p. S5, 2013.

[33] Z. Teng, M. Guo, Q. Dai, C. Wang, J. Li, and X. Liu, "Computational Prediction of Protein Function Based on Weighted Mapping of Domains and GO Terms", *BioMed Research International*, Vol. 2014, p. 641469, 2014.

[34] Z. Wang, C. Zhao, Y. Wang, Z. Sun, and N. Wang, "PANDA: Protein function prediction using domain architecture and affinity propagation", *Scientific Reports*, Vol. 8, No. 1, p. 3484, 2018.

[35] D. Piovesan, M. Giollo, E. Leonardi, C. Ferrari, and S. C. E. Tosatto, "INGA: protein function prediction combining interaction networks, domain assignments and sequence similarity", *Nucleic Acids Research*, Vol. 43, No. W1, pp. W134-W140, 2015.

[36] Y. Cai, J. Wang, and L. Deng, "SDN2GO: An Integrated Deep Learning Model for Protein Function Prediction", *Frontiers in Bioengineering and Biotechnology, Methods*, Vol. 8, No. 391, 2020.

[37] S. Das, H. M. Scholes, N. Sen, and C. Orengo, "CATH functional families predict functional sites in proteins", *Bioinformatics*, Vol. 37, No. 8, pp. 1099-1106, 2020.

[38] J. Tao, K. A. Brayton, and S. L. Broschat, "Automated Confirmation of Protein Annotation Using NLP and the UniProtKB Database", *Applied Sciences*, Vol. 11, No. 1, p. 24, 2021.

[39] H. D. Carroll, J. L. Spouge, and M. Gonzalez, "MultiDomainBenchmark: a multi-domain query and subject database suite", *BMC Bioinformatics*, Vol. 20, No. 1, pp. 1-9, 2019.

[40] E. Boutet, D. Lieberherr, M. Tognolli, M. Schneider, and A. Bairoch, "Uniprotkb/swiss-prot", *Plant Bioinformatics: Springer*, pp. 89-112. 2007.

[41] E. Boutet, D. Lieberherr, M. Tognolli, M. Schneider, P. Bansal, A. J. Bridge, S. Poux, L. Bougueleret, and I. Xenarios, "UniProtKB/Swiss-Prot, the manually annotated section of the UniProt KnowledgeBase: how to use the entry view", *Plant Bioinformatics: Springer*, pp. 23-54. 2016

[42] Sugiyarto, J. Eliyanto, N. Irsalinda, and M. Fitrianawati, "Fuzzy sentiment analysis using convolutional neural network", *AIP Conference Proceedings*, Vol. 2329, No. 1: AIP Publishing LLC, p. 050002, 2021.

[43] J. Peng, J. Li, and X. Shang, "A learning-based method for drug-target interaction prediction based on feature representation learning and deep neural network", *BMC Bioinformatics*, Vol. 21, No. 13, pp. 1-13, 2020.

[44] W. Gao, S. P. Mahajan, J. Sulam, and J. J. Gray, "Deep learning in protein structural modeling and design", *Patterns*, p. 100142, 2020.

[45] N. Zhou, Y. Jiang, T. R. Bergquist, A. J. Lee, B. Z. Kacsoh, A. W. Crocker, K. A. Lewis, G. Georghiou, H. N. Nguyen, and M. N. Hamid, "The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens", *Genome Biology*, Vol. 20, No. 1, pp. 1-23, 2019.

[46] Y. Jiang, T. R. Oron, W. T. Clark, A. R. Bankapur, D. D'Andrea, R. Lepore, C. S. Funk, I. Kahanda, K. M. Verspoor, and A. B. Hur, "An expanded evaluation of protein function prediction methods shows an improvement in accuracy", *Genome Biology*, Vol. 17, No. 1, pp. 1-19, 2016.

[47] I. Friedberg and P. Radivojac, "Community-wide evaluation of computational function prediction", *The Gene Ontology Handbook: Humana Press*, pp. 133-146, 2017.

[48] P. Radivojac, W. T. Clark, T. R. Oron, A. M. Schnoes, T. Wittkop, A. Sokolov, K. Graim,

C. Funk, K. Verspoor, and A. B. Hur, "A large-scale evaluation of computational protein function prediction", *Nature Methods*, Vol. 10, No. 3, pp. 221-227, 2013.

[49] J. Zhu, Q. Zhao, E. Katsevich, and C. Sabatti, "Exploratory Gene Ontology Analysis with Interactive Visualization", *Scientific Reports*, Vol. 9, No. 1, p. 7793, 2019.

[50] A. Tomczak, J. M. Mortensen, R. Winnenburg, C. Liu, D. T. Alessi, V. Swamy, F. Vallania, S. Lofgren, W. Haynes, N. H. Shah, M. A. Musen, and P. Khatri, "Interpretation of biological experiments changes with evolution of the Gene Ontology and its annotations", *Scientific Reports*, Vol. 8, No. 1, p. 5115, 2018.