Computing and Informatics, Vol. 42, 2023, 667–689, doi: 10.31577/cai_2023_3_667

BERTDOM: PROTEIN DOMAIN BOUNDARY PREDICTION USING BERT

Ahmad Haseeb, Maryam Bashir, Aamir Wali

FAST School of Computing
National University of Computer and Emerging Sciences
Lahore, Pakistan
e-mail: l182081@lhr.nu.edu.pk, {maryam.bashir, aamir.wali}@nu.edu.pk

Abstract. The domains of a protein provide an insight on the functions that the protein can perform. Delineation of proteins using high-throughput experimental methods is difficult and a time-consuming task. Template-free and sequence-based computational methods that mainly rely on machine learning techniques can be used. However, some of the drawbacks of computational methods are low accuracy and their limitation in predicting different types of multi-domain proteins. Biological language modeling and deep learning techniques can be useful in such situations. In this study, we propose BERTDom for segmenting protein sequences. BERTDOM uses BERT for feature representation and stacked bi-directional long short term memory for classification. We pre-train BERT from scratch on a corpus of protein sequences obtained from UniProt knowledge base with reference clusters. For comparison, we also used two other deep learning architectures: LSTM and feed-forward neural networks. We also experimented with protein-to-vector (Pro2Vec) feature representation that uses word2vec to encode protein bio-words. For testing, three other bench-marked datasets were used. The experimental results on benchmarks datasets show that BERTDom produces the best F-score as compared to other template-based and template-free protein domain boundary prediction methods. Employing deep learning architectures can significantly improve domain boundary prediction. Furthermore, BERT used extensively in NLP for feature representation, has shown promising results when used for encoding bio-words. The code is available at https://github.com/maryam988/BERTDom-Code.

Keywords: Protein, protein domain boundary, BERT, biLSTM

1 INTRODUCTION

Protein domain boundary are the residues on a protein sequence where a domain starts and ends. A protein sequence or chain can consist of single domains or multiple domains where each domain is comprised of its own folded and independent sub-structures [1]. Protein domains are structural or functional units of a protein. Domains are recurring sequences that give very important information for the prediction of protein structure, function, and evolution. Numerous modular proteins families can have domains of different degrees of quantity and order [2]. Protein domains are building blocks of protein and so they can be arranged in different combinations to form proteins with more complex functions. Therefore, accurate identification of domains in protein is key to understanding the evolutionary mechanisms and protein function [3].

There are two ways of identifying domains in proteins: the first one is to predict boundaries of the domain from proteins having known three-dimensional (3D) structures, and the second one is the protein domain identification of those having unknown 3D structures. Domain boundary prediction is the first crucial step in protein classification and predicting protein 3D structures, which is a high-complexity problem [1]. Precise and accurate prediction of domain boundaries is the basis of various kinds of protein research because these researches start with the segmentation of a protein into its domains, which are its functional units [4]. The domain boundary prediction can optimize search methods for templates used in comparative modeling as the classification of templates is based on protein domains. Also, accurate prediction for homologous domains plays a central role in reliable MSA (multiple sequence alignment) [5].

Currently, the most accurate and reliable depiction of the protein domain is by experimental methods. Experimental methods for identifying protein domains require huge amount of proteins, effort and time. High-throughput technologies generate a large amount of data, so it is not possible to manually detect protein domain. This is why computational protein domain prediction methods are preferred. Computational methods use protein sequences to predict and identify protein domains. The delineation of protein domains using only protein sequences is still difficult. The computational domain boundary prediction methods mainly consist of template-based methods and ab-initio. Template-based methods use patterns or templates of existing similar protein sequences with known domain information for the prediction of proteins with unknown boundary information. Ab-initio methods use machine learning and statistical algorithms for prediction. These methods are more popular than template-based methods because they can be applied to any protein sequence. Some examples of these methods are DomPro [4], PPRODO [2], DROP [6], and DeepDom [3]. They are mostly used because they can predict any protein. However, the major drawback of ab-initio methods is low accuracy and precision as compared to template-based methods [3].

1.1 Motivation

Characterization of proteins using high-throughput experimental methods is difficult. Most of the template-free computational methods proposed for protein boundary domain prediction rely on machine learning techniques. To the best of our knowledge, not much work has been done to predict the domain boundary using deep learning methods except [3].

1.2 Objectives

The primary objective of this study is to predict the protein boundary domain using deep learning techniques. Furthermore, this study also aims to explore if deep learning techniques used in conjunction with biological language modeling and NLP techniques like bi-directional encoder representations from transformers (BERT) [7] can improve prediction.

Protein domain boundary prediction pipeline usually has the following steps. Protein sequences are segmented. For this purpose there are various techniques such as wordPiece, sentence-piece, or TAPE tokenizers like IUPAC and UniRef. Then the tokens or bio-words are encoded. BERT is a popular method for language representations. It provides a contextual representation of every bio-word in a sequence and can therefore be used for encoding. Other encoding schemes include word2vec and pro2vec. Finally, these representations are used to train classifiers. Thus, every step can be performed using a number of techniques. Another objective of this paper is to experiment with different combinations of segmentation-encoding-classification techniques and identify which combination works best for protein domain boundary classification. For this purpose, various deep learning architectures and methods like BERT, long short-term memory (LSTM) and fully convolutional neural networks (FCNN) are used which are extensively applied in other NLP and bio-informatics tasks. The prediction models are trained on protein sequences alone and does not rely on features engineering like sequence profile, solvent accessibility (SA), secondary structure (SS), etc.

1.3 Contributions

Following are the main contributions of this study.

- A protein domain boundary prediction model called BERTDom is proposed using deep learning techniques, BERTDom outperforms other template-based and computational techniques on benchmark datasets.
- Pre-trained BERT from scratch for protein bio-word embeddings for the first time.
- Protein vector representations created using pro2vec are used as features for protein domain boundary prediction.

- A multi-facet comparison is done involving two feature representations, three segmentation techniques and three deep learning models.

The rest of the paper is organized as follows. Section 2 presents necessary background required for understanding the problem of protein domain boundary prediction. Section 3 presents literature review on relevant work related to this study. Section 4 presents methodology used for protein domain boundary prediction in this study and Section 5 presents experimental details. Section 6 presents results and discussion and Section 7 concludes the study.

2 EXTENDED BACKGROUND

In this section, all the necessary concepts concerned with protein domain boundary are presented.

2.1 Protein and Its Domains

Protein performs a wide range of functions within living organisms, including transporting molecules from one location to another, responding to stimuli, providing structure to cells and organisms, DNA replication, and catalyzing metabolic reactions [8]. Protein is composed of amino acids and typically, 20 types of amino acids are found in proteins. Depending on the protein sequence, i.e., the position of amino acids in the protein chain, proteins fold into the specific 3D structure that allows them to do their functions and interact with other molecules and proteins. Proteins that have a common ancestor or diverged from the same ancestral gene are called homologous and have similar sequences [8].

Protein domain is a constant part of a protein sequence and makes a compact 3D structure that can fold independently. The length of a domain can be anywhere from 50 to 250 residues [9]. Due to molecular evolution, protein domains can be used as building blocks and they can be combined in different ways to form proteins with distinct structures and functions [10]. Each domain contributes to the overall functions of the protein. For instance, enzyme phospholipase D1 protein is a multi-domain protein since it has 3 different types of domains each performing a different sub-function to achieve an overall function of breaking down phosphatidylinositol.

Most domains comprise one continuous segment; some domains may consist of several discontinuous polypeptide segments [3]. The prediction and identification of discontinuous domains is still a very challenging problem.

2.2 Methods for Predicting Protein Domain Boundary

Methods for predicting protein domain boundaries are of two types: experimental and computational. These are discussed next.

2.3 Experimental Methods

Experimental methods are procedures performed on actual proteins. These methods use the particular biophysical or biochemical attributes of protein complexes. They can be done in a controlled lab environment (in-vitro) or inside a living organism (in-vivo). To speed up the process, high-throughput large-scale experimental methods have been designed to identify domains in a protein on a proteomic-wide scale. High-throughput experimental methods used for identifying domains are NMR (Nuclear Magnetic Resonance) analysis [9], and X-ray crystallography [11]. These methods are expensive in terms of labor, money, and time. These methods also need large quantities of proteins. Their results have high false negatives and false positives because the experiment's quality is affected by many factors [12]. Due to these limitations, computational methods are needed in the domain boundary prediction. Hence, it is of great practical importance to design accurate, reliable, and efficient computational methods to predict domains in less time, with high efficiency and at low cost.

2.4 Computational Methods

Computational methods for the prediction of protein domain boundary can be classified as template-based methods or template-free/ab-initio methods. The template-based methods search for similar protein sequences whose domain information is known and them map this information to the protein with unknown domain.

Some template-based approaches use sequence alignment in which the query and target protein sequences are aligned to predict the domain [13]. While other methods predict by aligning the secondary structure (SS) of a protein against the known domain boundary information of proteins given in class, architecture, topology, and homology (CATH) database [4].

Ab-initio or template-free methods are based only on the primary 1D protein sequence instead of any specific target protein [14]. These methods are more commonly used as compared to template-based methods since ab-initio methods can predict the domain boundary of any protein.

Ab-initio based machine learning (ML) methods directly or indirectly use the amino acid sequence as features to predict whether an amino acid is situated at a domain boundary. Ab-initio techniques are assisted by the accessibility of protein domain information databases. Ab-initio methods usually use the same input features like sequence profiles (SP), predicted solvent accessibility (SA) and predicted secondary structure (SS). For example, [15] also used amino acid composition and solvent accessibility to predict secondary structure.

The prediction accuracy of the ab-initio methods is usually lower than the template methods because of the lack of complete domain boundary information in protein sequence [3]. Most ab-initio methods are effective and successful in predicting domain boundaries when the target protein sequence has obvious resemblances to other sequences in domain classification databases or if the new

domains' length does not significantly differ from the average length of known protein sequences. In this paper, the focus is on identifying boundaries for proteins with two domains. The accuracy for one-domain proteins using computational methods is only 75–85 %, and it is significantly less for multi-domain proteins [5].

3 LITERATURE REVIEW

In this section, various computational methods for predicting protein domain boundaries are discussed. As mentioned earlier, computational methods can be ab-initio or template-based. A few hybrid techniques are also presented.

3.1 Template-Based Methods

Although the focus of this paper is the ab-initio methods, some template-based methods TBMs are briefly discussed.

Bondugula et al. [16] proposed FIEFDom, a homology-based approach for protein domain boundary prediction for multi-domain protein using features such as sequence profile and protein sequence using an FMO (fuzzy mean operator). The FMO assigns a likelihood score for each amino acid of the target sequence as corresponding to a domain boundary or not by using the NR (non-redundant) sequence database along with an RPS (reference protein set) database comprising already identified domain boundaries. This method vigorously identifies adjoining boundary sites. Authors claim the average prediction accuracy for single-domain and multi-domain proteins is 97 % and 58 % respectively. The proposed model has the ability to use new structure/sequence information after each RPS update without re-parameterization. When tested on other datasets having different domain information, this method consistently produced the same accuracy while other existing methods could not.

Zhidong Xue et al. [17] proposed another technique called ThreaDom, which infers protein domain boundary regions using multiple threading alignments. The key to this approach is that it can calibrate sequence alignment information and composite structure by generating a domain boundary profile from the multiple threading templates for exact domain prediction. ThreaDom correctly classifies 81 % of single-domain and multi-domain proteins when 78 % proteins have the domain linker allotted in the range of ± 20 residues. Finally, George et al. [18] developed SnapDRAGON, a 3D template-based approach for domain boundary prediction. It predicts domain boundary based on features from a secondary structure prediction and multiple alignments of protein sequences. SnapDRAGON utilizes the DRAGON method to generate a large set of alternative 3D models for a given multiple sequence alignment (MSA). Then it assigns domain boundaries automatically to each of the 3D model structures. Domain boundary assignment seen in the largest number of 3D models is selected. Model generation using this method leads to alternative 3D

model structures that differ in structure with associated boundary positions and have different domain contents. This technique used on NR dataset consisting of 414 multiple sequence alignments constitutes, 231 multiple-domain and 185 single protein chains registered an accuracy of 72.4 %.

3.2 Ab-Initio/ML-Based Methods

Sim et al. [2] proposed an ab-initio method for prediction of protein domain boundaries called PPRODO. PPRODO uses a feed-forward fully connected neural network with one hidden layer. A neural network is trained and tested for each residue in the protein sequence [19]. Amino acid residues in a protein sequence may mutate and this is more regular if the residues are close to domain boundaries. However, during the evolution some residues close to the domain boundaries may be conserved despite the usual movement of the domain. Analyzing the patterns in the position-specific scoring matrix can detect these features.

Cheng et al. [4] propose DOMpro that uses recursive neural networks to predict domain boundary using profiles, predicted secondary structure, and predicted relative solvent accessibility. This paper used the dataset from CATH database. The solvent accessibility and relative secondary structure are predicted for each sequence using ACCpro [20] and SSpro[21]. DOMPro can accurately predict the domain boundary and domains number for 25 % of the proteins that have two domains.

Yoo et al. [5] proposed the method DomNet that uses an enhanced general regression network (EGRN) specially created for managing high-dimensional protein sequences. DomNet uses a novel compact domain profile so that it can obtain more structural information efficiently from target sequences. The input features used by this method for training are predicted solvent accessibility information, predicted secondary structure, inter-domain linker index that detects the target protein sequence's possible domain boundaries and a compact domain profile. DomNet uses methods proposed by [22] for noise reduction, smoothing and searching vectors center by quantizing input vectors. DomNet reports the 71 % accuracy for proteins with multiple domains.

Ebina et al. [6] used a support vector machine (SVM) for prediction. This paper also used random forest to compute optimal input features which are then used to train SVM. Each amino acid residue is encoded into a 3000-dimensional vector. Various SVM classifiers were trained with different optimal feature candidate sets. SVM hyper-parameters were optimized using a SVMLab [23]. The proposed model named DROP, had sensitivity and precision values of 19.9 % greater than SVMs trained with non-optimized features using the same parameters. SVM was also used by Chakraborty et al. [24] based on input features composed of physiochemical properties of amino acids in protein sequence (obtained from AAIndex [25] database), predicted solvent accessibility and predicted secondary structure. Physiochemical properties of amino acid residues are linker index, hydrophobicity, linker propensity indices, polarity, and average flexibility indices. This method achieved

a precision, recall and accuracy of 0.79 %, 0.91 % and 78.58 % respectively, on the CASP10 dataset.

Eickholt et al. [14] developed DoBo, which also used SVM to classify the putative domain boundary signals. These signals are extracted from MSA generated by PSI-BLAST [17]. These MSA helps to detect assumed signals of domain boundary in a query protein by leveraging evolutionary information. MSAs often disclose the query protein's domain architecture by returning proteins comprised of domains analogous to the query protein sequence. DoBo has a recall and precision rate of 0.6. Finally, Bi-Qing Li et al. [26] also combined SVM with multiplefeature selection methods. This paper reported about 58–70 % higher specificity, 24–31 % greater MCC, and 28–40 % more accuracy than the DoMpro, Globplot, and Domcut methods but 20 % less sensitivity.

Hwan Hong et al. [1] proposed ConDo that used a 4-layer neural network for prediction of domain boundary. This method employed both short-range features such as sequence information, as well as long-range sequence information like evolutionary information and partially aligned sequences (PAS) in MSA. Long-range features are beneficial for deciding whether two residues belong to either separate domains or the same domain. Short-range features are residue position in a sequence, whether the residue is outside of the target chain, the number of residues in a sequence, sequence profile, predicted SA, and predicted SS. HHblits generates the sequence profile with UniRef20 database. SANN [27] predicted SA. PSIPRED [28] predicted SS. Neural networks' output layer has four units, which state whether or not the amino acid was within 20, 15, 10, or 5 amino acids from the correct domain boundary.

Jiang et al. [3] proposed DeepDom, a deep learning domain boundary prediction method that uses LSTM. DeepDom stacks multiple bi-directional LSTM layers to fit a non-linear high-order function with the aim of predicting the signal pattern of complex domain boundary. It uses a window sliding strategy to encode an input sequence into fixed-length protein fragments without considering the original length of the protein sequence. The majority of existing ab-initio domain boundary classifiers only permit users to provide and predict one protein sequence at one time. DeepDom does not perform the time-consuming and computationally intensive task of sequence profile generation method.

3.3 Hybrid Methods

Hybrid methods combine both ab-initio and template-based techniques. Walsh et al. [29] used bi-directional recurrent neural networks for predicting protein domain boundaries. The work also used structural classification of proteins (SCOP) and protein data bank (PDB) template profiles. Using template information improves the performance of ab-initio. Cheng et al. [30] describe DOMAC, a hybrid domain boundary prediction technique that integrates domain parsing, ab-initio, and homology modeling methods. This hybrid approach uses neural networks and the homology-based method to predict domain boundaries for proteins having homolo-

gous template structures in PDB to predict domain boundaries for new proteins.

4 METHODOLOGY

A protein sequence can be segmented using a number of techniques. The segmented bio-words can further get encoded using different techniques. For classification, a variety of deep learning models are available. In this section, we not only highlight the proposed BERTDom model, but also specify different combination of segmentation-encoding-classification techniques that were used in this paper for experimentation and comparison with BERTDom.

The high-level block diagram for BERTDom is given in Figure 1. In the first step, protein sequence is segmented into bio-words using wordPiece segmentation algorithm. Then every bio-word is encoded using BERT. Finally, the entire encoded protein sequence is fed to the stacked biLSTM classifier that predicts the domain boundary. Each of these step are discussed next. The state-of-the-art deep learning and NLP components of BERTDom model are also sufficiently discussed due to the multi-disciplinary nature of the current study.

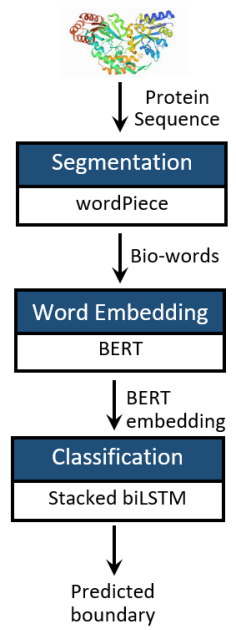

Figure 1. Architecture of BERTDom based on BERT and stacked biLSTMs for protein domain boundary prediction

4.1 WordPiece Algorithm for Bio-Word Segmentation

WordPiece algorithm [31] is a tokenizer that splits sentences into words and then words into sub-words. It is used in natural language processing (NLP) and deep learning architectures like BERT. WordPiece is trained for protein sequences and can therefore be used for segmenting protein sequence. The training parameters include the vocabulary size: 80 000, minimum words frequency: 2, and maximum sequence length: 256. WordPiece outputs a vocabulary file containing all words, sub-words, and individual characters in protein sequences. The trained wordPiece tokenizer is fed this vocabulary file along with the protein sequences and it segments them into bio-words.

4.2 Pre-Training of BERT Language Model for Protein Word Embeddings

Bidirectional encoder representations from transformers (BERT) is developed by Google AI researchers [7]. BERT consists of two steps: pre-training and fine-tuning. In pre-training, a large amount of unlabeled text is input to the BERT model for training where it learns the contextual relations between words and sentences in the language. BERT has two sub-models: masked language modeling (MLM) and next sentence prediction (NSP). MLM takes in a sentence with some masked words and it needs to predict the masked words. During fine-tuning, the last output layer of BERT is replaced by a new fully-connected layer that is trained for the specific task. Although each task is initialized with the same pre-trained weights in the non-final layers, the last layer of BERT is fine-tuned. The same pre-trained BERT weights can also be used to initialize other deep learning models for any sequence based prediction task. Protein domain boundary prediction can be modeled as a sequence based prediction task. This study proposes to use BERT for feature representations of protein sequences. Since there is no pre-trained BERT model available for protein bio-words, BERT had to be pre-trained for protein bio-word embeddings from scratch.

Figure 2 shows the architecture of one encoder state of BERT. BERT has multiple encoder states. Each encoder state has the same architecture. BERT processes sequence-based information by using a multihead attention mechanism. The input to BERT is a vector of words that have positional encoding information added to them. The self-attention layer takes the dot product of the input word with all query vectors of all other words in the sequence. A normalization layer is added after the self-attention layer. The next layer is the feed-forward neural network layer. The output of one encoder is passed as input to the next encoder state. The final output is the vector representation of input words such that the representation of each word has information of surrounding words baked into it.

The vocabulary file is converted to TFRecord format which is then used to pre-train BERT. The model configurations are given in Table 1. This BERT model's

configuration is the same as BERT-Medium uncased configuration – only difference is vocabulary size, which is changed from 30 500 to 80 000.

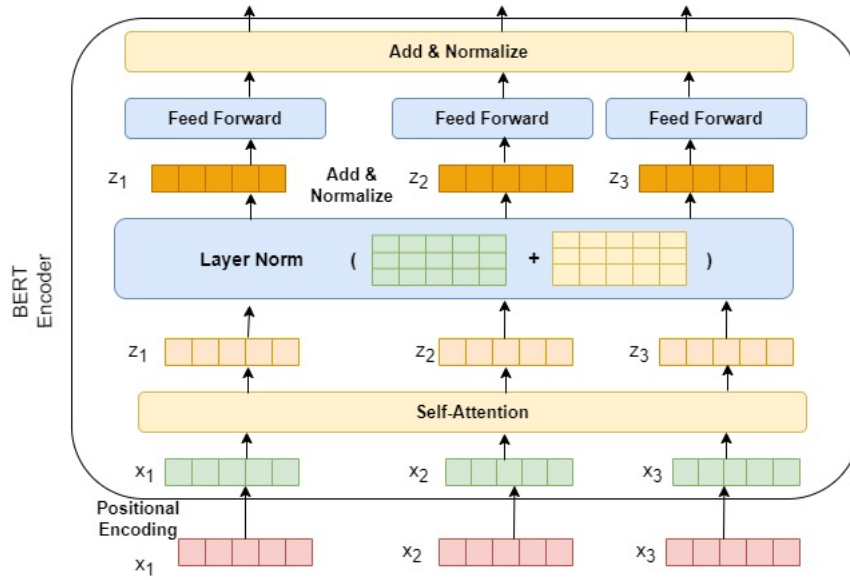

Figure 2. Architecture of BERT encoder [32]

Batch size of input examples	16
Maximum sequence length	256
Maximum predictions per sequence	35
Number of Training steps	30 000
Learning Rate	1e−4
Optimizer	Adam

Table 1. Hyper-parameters for pre-training BERT

The proteins can be of variable-length, so they are broken down into fixed-length protein sub-sequences using a sliding window strategy. The optimal values of window and stride were found to be 200 and 80 respectively. Each of these protein sub-sequences are tokenized using WordPiece tokenizer and then vectorized by the pre-trained BERT. BERT gives an embedding of 512 dimension for each token.

For fine-tuning BERT, BERTDom uses stacked biLSTM. However, we also performed experiments using 2 other deep learning techniques. These are also discussed in the following sub-sections.

4.3 BERTDom: Stacked Bi-LSTM and BERT's Language Model

For fine-tuning BERT, three models were used. The first model is stacked bidirectional LSTMs. LSTM [33] is a deep learning architecture that can process a sequence of data such as speech, text, or time-series. The true power of LSTMs lies in their ability to model longer sequences. LSTM is modified form of recurrent neural network (RNN) which were proposed for representation of sequence data. RNN suffer from the problem of vanishing gradient which occurs for long sequence of data. The gradient can become very small during back propagation in long sequences, this is called vanishing gradient problem [34]. LSTM overcome this problem by using a cell state for remembering only important information.

BiLSTM has one LSTM that processes sequence from start-to-end and another LSTM that processes the same sequence backwards. These two LSTMs are combined using the concatenation operator. Stacked biLSTM has multiple layers of biLSTM stacked n top of one another. The pre-trained BERT model is attached to stacked bidirectional LSTMs that has four bidirectional LSTM layers. The softmax is used as the activation function. The number of output units in last layer of each LSTM is equal to the maximum length of protein sequence which is 200.

The second deep learning model used for fine-tuning BERT is LSTM. The number of output units in the last layer is 200 with the softmax as the activation function.

Finally, the third deep learning model used for fine-tuning BERT is a deep feed-forward neural network. The network has 4 hidden layers with 1500 units and dropout values of 0.5, 0, 0, and 0.5 respectively. ReLU and sigmoid are used as activation functions. The number of output units in the last layer is equal to the length of the protein sequence – 200.

4.4 Feature Representation for Protein Bio-Word Using Protein-to-Vector (pro2vec)

In this study, for comparison purposes, we also used another feature representation method for segmented protein bio-words called pro-to-vector (pro2vec)[35] instead of BERT. For pro2vec model, word2vec algorithm called the skip-gram is used. Word2vec's skip-gram is used for learning the distributed representation for every protein word in proteins. We also trained word2vec from scratch on 185 000 protein sequences obtained from UniRef dataset [36] in sequence clusters with identity of 50 % (UniRef50). For classification, bidirectional LSTM is used. For segmentation, sentencePiece segmentation, [37] and K-mer segmentation techniques are used instead of wordPiece. The same window size and stride of 200 and 80 respectively, are used. Lastly, all protein word vectors of a protein sequence are combined together to form the embedding matrix of the protein sequence and then fed to a bidirectional LSTM for prediction of domain boundaries in a protein.

4.4.1 SentencePiece Segmentation

SentencePiece is a language independent tokenizer which is used when size of vocabulary is already known. It is trained directly from raw text using unigram language model (ULM). The sentencePiece library is used to implement this technique [38]. SentencePiece is an unsupervised method for tokenizing text.

4.4.2 K-mer Segmentation

K-mers are subsequences of length k in a protein sequence. For a given protein sequence, k-mer segmentation is used to divide them into bio-words. For example, the sequence MSLQ would have four monomers (M, S, L, and Q), three 2-mers (MS, SL, LQ), two 3-mers (MSL and SLQ) and one 4-mer (MSLQ). For length Z of a given protein sequence, we will get $Z - k + 1$ k-mers or bio-words.

4.5 Comparison with Other Methods

We have compared our proposed methods with existing template-based approaches such as Pfam [39], and FIEFDOM [16]. In addition to template-based methods we have also compared our proposed methods with statistical and machine learning approaches such as DomPro [4], PPRODO [2], and DROP [6]. DeepDom [3] is a recently proposed deep-learning-based method that uses LSTM for protein domain boundary prediction. DeepDom [3] has shown superior performance as compared to many template-based and statistical methods so we also compared our proposed methods with DeepDom.

5 EXPERIMENTAL SETUP

This section presents details of training and test data used in experiments. The evaluation measures are also discussed.

5.1 Training Dataset

For training, 46 000 domain boundary annotations of proteins from the CATH [40] version 4.2 database were collected. Uniprot database [41] is used for downloading corresponding sequences of these proteins. After downloading proteins, CD-HIT [42] tool is used to cluster similar proteins that meet the predefined 40 % similarity threshold. The representative protein sequences have sequence similarity less than 40 % with every other protein [43]. The similarity threshold (40 %) is used to make sure sufficient diverse data is available for the training of LSTM and BERT models.

5.2 Test Dataset

The proposed methods are tested on the proteins in the critical assessment of techniques for protein structure prediction (CASP) dataset which is a benchmark dataset. CASP protein domain prediction competition provided the annotations of domain boundaries of test proteins. Proteins in the training dataset that have at least 40 % similarity with any test proteins were removed from the training dataset. CASP provided three types of test datasets for bench-marking. These test datasets are listed below. Their details can be found in [3].

1. Free modeling (FM) target proteins from CASP 9.
2. Multi-domain proteins from CASP 9.
3. Discontinuous domain targets from CASP 8.

Dataset	# of Proteins	Single Domain	Multiple-Domain
Free Modeling	22	12	10
Multi-domain	14	0	14
Discontinuous domain	18	1	18

Table 2.

5.3 Dataset Used to Pre-Train Language Model – UniProt UniRef50

The UniRef50 protein dataset was used for pre-training language models (BERT and Word2vec) is obtained from UniProt Knowledge base (UniProtKB) with reference clusters (UniRef). UniRef gives clustered sequences' sets from the chosen UniParc records and UniProt. It removes protein sequences that are redundant and acquires whole coverage of the sequence space at 3 resolutions, which are UniRef50, UniRef90, and UniRef100. UniRef50 dataset was used for pre-training the language model. UniRef50 dataset contains 185 000 protein sequences.

5.4 Parameter Settings

DeepDom [3] is trained on 57 000 protein sequences while our proposed BERT model is trained on the dataset described above. Word2vec is trained on UniRef50 dataset with a window size of 10 and word vector dimension of 50 for ULM and K-Mer methods. The ULM is implemented using the sentencePiece library. It trains on UniRef dataset with a maximum vocabulary size of 50 000. Word2vec is also trained on a training dataset with a window size of 10.

5.5 Evaluation Measures

The proposed methodology is evaluated using benchmark classification evaluation measures, precision, recall, F-score, and accuracy. The formulas for these measures

are given as follows:

$$precision = \frac{TP}{TP + FP},$$

$$recall = \frac{TP}{(TP + FN)},$$

$$Accuracy = \frac{TP + TN}{FP + FN + TP + TN)},$$

$$F\text{-}score = \frac{2 * precision * recall}{precision + recall},$$

where, when a residue is predicted a domain boundary region, then it is checked if it is within ±20 residues of the actual domain boundary region. If yes, then it is a true positive (TP). If no, then it is a false positive (FP). When a residue is predicted outside the domain boundary region, then it is checked if it is within ±20 residues of the actual domain boundary region. If yes, then it is a false negative (FN). If no, then it is a true negative (TN).

6 RESULTS AND DISCUSSION

In this section, the results of the proposed methods are discussed for all datasets. Performance comparison between all methods is discussed in the following sections.

6.1 Performance on Free Modeling (FM) Targets

Table 3 presents the results of our proposed methods using free modeling (FM) targets from CASP9. Our proposed methods can be categorized into two main categories based on feature representation. The first is BERT encoder and the second is pro2vec. BERT is used as an encoder for protein sequences and then it is fine-tuned using three different deep learning models (LSTM, BiLSTM, and FCNN). BERTDom (BERT fined tuned with BiLSTM) performs best as compared to other models. The F-score is 0.58. Pro2vec is the second feature representation method in our experiments. Pro2vec was used with K-mer and unigram language model for segmenting a sequence into bio-words. Different values of k (3,4, and 5) have been tried, it is shown by results that 3-mer performs better than 4-mer and 5-mer segmentation. The F-score using pro2vec with 3-mer is also 0.58. The results of pro2vec with unigram model (0.57) are also close to pro2vec with 3-mer. The performance of BERT fined tuned with LSTM and FCNN is much inferior to BERT fine-tuned with biLSTM. The reason for this difference can be the bidirectional nature of biLSTM which takes into account the context from both directions.

	Accuracy	Precision	Recall	F-Score
BERT fine-tuned with LSTM	0.53	0.70	0.42	0.52
BERT fine-tuned with Stacked biLSTM (BERTDom)	0.74	0.74	0.47	**0.58**
BERT fine-tuned with Deep FCNN	0.71	0.69	0.43	0.53
Pro2Vec with 3-mer (biLSTM)	0.73	0.71	0.49	**0.58**
Pro2Vec with ULM (biLSTM)	0.76	0.84	0.43	0.57

Table 3. Comparison of proposed methods for FM dataset

	Accuracy	Precision	Recall	F-Score
BERT fine-tuned with LSTM	0.48	0.75	0.39	0.51
BERT fine-tuned with Stacked biLSTM (BERTDom)	0.76	0.82	0.45	0.58
BERT fine-tuned with Deep FCNN	0.74	0.79	0.38	0.51
Pro2Vec with 3-mer (biLSTM)	0.74	0.70	0.51	**0.59**
Pro2Vec with ULM (biLSTM)	0.76	0.84	0.41	0.55

Table 4. Comparison of proposed methods for multi-domain protein dataset

	Accuracy	Precision	Recall	F-Score
BERT fine-tuned with LSTM	0.50	0.75	0.43	**0.55**
BERT fine-tuned with stacked biLSTM (BERTDom)	0.70	0.82	0.33	0.47
BERT fine-tuned with Deep FCNN	0.70	0.81	0.32	0.46
Pro2Vec with 3-mer (biLSTM)	0.67	0.66	0.37	0.47
Pro2Vec with ULM (biLSTM)	0.68	0.79	0.28	0.41

Table 5. Comparison of proposed methods for DCD Dataset

		Precision	Recall	F-Score
Template based methods	Pfam [39]	0.32	0.49	0.39
	FIEFDOM [16]	0.23	0.18	0.2
Statistical and machine learning methods	DomPro [4]	0.50	0.18	0.26
	PPRODO [2]	0.33	0.49	0.39
	DROP [6]	0.43	0.18	0.25
	DeepDom [3]	0.89	0.41	0.56
Proposed methods	BERT fine-tuned with stacked biLSTM (BERTDom)	0.74	0.47	**0.58**
	Pro2Vec with 3-mer (biLSTM)	0.71	0.49	**0.58**

Table 6. Comparison of proposed method (BERT with stacked biLSTM) with other methods for FM dataset

		Precision	Recall	F-score
Template based	Pfam [39]	0.50	0.55	0.52
	FIEFDOM [16]	0.34	0.23	0.27
Statistical and machine learning methods	DomPro [4]	0.50	0.14	0.22
	PPRODO [2]	0.5	0.52	0.51
	DROP [6]	0.68	0.26	0.38
	DeepDom [3]	0.76	0.45	0.57
Proposed methods	BERT fine-tuned with stacked biLSTM (BERTDom)	0.82	0.45	0.58
	Pro2Vec with 3-mer (biLSTM)	0.7	0.51	**0.59**

Table 7. Comparison of proposed method (BERT with stacked biLSTM) with other methods for multi-domain dataset

6.2 Performance on Discontinuous Domain Targets (DCD)

Table 4 presents results on discontinuous domain targets. The results on this dataset are similar to results on the FM dataset. Pro2vec with 3-mer performs best with an F-score of 0.59, whereas, BERTDom has similar results with an F-score of 0.58. The rest of the models do not perform as well as these two models. Pro2vec, based on word2ec, learns a representation of bio-words based on the context. This contextual information helps in learning a better representation of the input data.

6.3 Performance on Multi-Domain Targets

Table 5 presents results using multi-Domain targets. BERTDom shows best results for multi-domain targets with an F-score of 0.55. This model has good precision as well as better recall as compared to other models. The rest of the models have good precision but low recall so the F-score of the rest of the models is less than BERT fine-tuned with LSTM. BERT fine-tuned with FCNN has inferior performance as compared to BERT fine-tuned with LSTM or biLSTM. The reason for this performance is the sequential nature of protein sequence data. LSTM and biLSTM are sequence-based models which remember context information.

6.4 Comparison with Other Methods

Table 6 and Table 7 present a comparison of our best performing proposed models (BERTDom and pro2vec with 3-mer) with existing work. Existing work can be divided into two categories. The first category is template-based methods. Our proposed method BERTDom outperforms template-based methods using F-score with a large margin. The F-score with BERTDom is 0.58 for FM dataset as shown in Table 6, whereas, Pfam [39] and FIEFDOM [16] have very low F-score of 0.39

Dataset			Precision	Recall	F-Score
FM	Template based methods	Pfam [39]	0.32	0.49	0.39
		FIEFDOM [16]	0.23	0.18	0.2
	Statistical and machine learning methods	DomPro [4]	0.50	0.18	0.26
		PPRODO [2]	0.33	0.49	0.39
		DROP [6]	0.43	0.18	0.25
		DeepDom [3]	0.89	0.41	0.56
	Proposed methods	BERT fine-tuned with stacked biLSTM (BERTDom)	0.74	0.47	**0.58**
		Pro2Vec with 3-mer (biLSTM)	0.71	0.49	**0.58**
Multi-domain	Template based	Pfam [39]	0.50	0.55	0.52
		FIEFDOM [16]	0.34	0.23	0.27
	Statistical and machine learning methods	DomPro [4]	0.50	0.14	0.22
		PPRODO [2]	0.5	0.52	0.51
		DROP [6]	0.68	0.26	0.38
		DeepDom [3]	0.76	0.45	0.57
	Proposed methods	BERT fine-tuned with stacked biLSTM (BERTDom)	0.82	0.45	0.58
		Pro2Vec with 3-mer (biLSTM)	0.7	0.51	**0.59**

Table 8. A Summary table for comparison of proposed method (BERT with stacked biLSTM) with other methods

and 0.2 respectively. Similarly, on the multi-domain dataset, our proposed methods have superior results as compared to template-based methods as shown in Table 7. We have also compared our proposed models with other statistical and machine learning models. Overall, our proposed models outperform the compared methods. DeepDom [3] performs best among the compared methods and our proposed models outperform DeepDom [3] as well. These results strengthen our belief that BERT and pro2vec give superior representations for protein sequences as compared to existing approaches. Table 8 presents results summary of comparison of our proposed methods with other methods. The best results are highlighted in bold. This table clearly shows the superior performance of our proposed deep learning methods for protein domain boundary prediction as compared to other approaches (template based, machine learning and statistical).

7 CONCLUSIONS

Protein domain boundary prediction is an important step in understanding the function of a protein. Most of the template-based methods have low accuracy so in recent years many computational approaches have been proposed for this problem. In this study, we have proposed a novel method BERTDom which trains the BERT model for the problem of protein domain boundary prediction. BERT is a popular model for the representation of text due to the sequential nature of the text. The protein sequence is also an example of sequence data so experimented with BERT for protein sequence data. The results are encouraging and show the potential of this multi-head attention-based model for protein sequence problems. The results are superior to many existing machine learning and template-based methods. We have also tried pro2vec for this problem. Pro2vec is inspired from word2vec for context-based words representation. The results with pro2vec are also superior as compared to exiting computational and template-based approaches.

The performance of deep learning models highly depends on the amount of training data. Google's BERT models are trained for at least 1 000 000 steps and are fed millions of documents, whereas we have trained the BERT model with only 10 000 steps and 185 000 sequences. The reason for the small training size is the lack of computational resources. Having said that, the results are promising. Thus, this study shows the potential of pre-trained BERT for protein domain boundary prediction even when trained on a small data. It is expected that if BERT is pre-trained with more data, the results can further improve.

REFERENCES

[1] Hong, S. H.—Joo, K.—Lee, J.: ConDo: Protein Domain Boundary Prediction Using Coevolutionary Information. Bioinformatics, Vol. 35, 2019, No. 14, pp. 2411–2417, doi: 10.1093/bioinformatics/bty973.

[2] Sim, J.—Kim, S. Y.—Lee, J.: PPRODO: Prediction of Protein Domain Boundaries Using Neural Networks. Proteins, Vol. 59, 2005, No. 3, pp. 627–632, doi: 10.1002/prot.20442.

[3] Jiang, Y.—Wang, D.—Xu, D.: DeepDom: Predicting Protein Domain Boundary from Sequence Alone Using Stacked Bidirectional LSTM. Biocomputing 2019: Proceedings of the Pacific Symposium, World Scientific, 2018, pp. 66–75, doi: 10.1142/9789813279827_0007.

[4] Cheng, J.—Sweredoski, M. J.—Baldi, P.: DOMpro: Protein Domain Prediction Using Profiles, Secondary Structure, Relative Solvent Accessibility, and Recursive Neural Networks. Data Mining and Knowledge Discovery, Vol. 13, 2006, No. 1, pp. 1–10, doi: 10.1007/s10618-005-0023-5.

[5] Yoo, P. D.—Sikder, A. R.—Taheri, J.—Zhou, B. B.—Zomaya, A. Y.: DomNet: Protein Domain Boundary Prediction Using Enhanced General Regression Net-

work and New Profiles. IEEE Transactions on NanoBioscience, Vol. 7, 2008, No. 2, pp. 172–181, doi: 10.1109/TNB.2008.2000747.

[6] Ebina, T.—Toh, H.—Kuroda, Y.: DROP: An SVM Domain Linker Predictor Trained with Optimal Features Selected by Random Forest. Bioinformatics, Vol. 27, 2011, No. 4, pp. 487–494, doi: 10.1093/bioinformatics/btq700.

[7] Devlin, J.—Chang, M. W.—Lee, K.—Toutanova, K.: BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In: Burstein, J., Doran, C., Solorio, T. (Eds.): Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2019). ACL, Vol. 1, 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.

[8] Wang, Y.—Zhang, H.—Zhong, H.—Xue, Z.: Protein Domain Identification Methods and Online Resources. Computational and Structural Biotechnology Journal, Vol. 19, 2021, pp. 1145–1153, doi: 10.1016/j.csbj.2021.01.041.

[9] Folkers, G. E.—van Buuren, B. N. M.—Kaptein, R.: Expression Screening, Protein Purification and NMR Analysis of Human Protein Domains for Structural Genomics. Journal of Structural and Functional Genomics, Vol. 5, 2004, No. 1, pp. 119–131, doi: 10.1023/B:JSFG.0000029200.66197.0c.

[10] Wikipedia Contributors: Protein Domain. Wikipedia, The Free Encyclopedia, `https://en.wikipedia.org/wiki/Protein_domain` (Retrieved 2022-02-11).

[11] Brenner, S. E.: Target Selection for Structural Genomics. Nature Structural and Molecular Biology, Vol. 7, 2000, No. 11, pp. 967–969, doi: 10.1038/80747.

[12] Wang, Y.—You, Z. H.—Yang, S.—Li, X.—Jiang, T. H.—Zhou, X.: A High Efficient Biological Language Model for Predicting Protein–Protein Interactions. Cells, Vol. 8, 2019, No. 2, Art. No. 122, doi: 10.3390/cells8020122.

[13] Marchler-Bauer, A.—Anderson, J. B.—DeWeese-Scott, C.—Fedorova, N. D.—Geer, L. Y.—He, S.—Hurwitz, D. I. et al.: CDD: A Curated Entrez Database of Conserved Domain Alignments. Nucleic Acids Research, Vol. 31, 2003, No. 1, pp. 383–387, doi: 10.1093/nar/gkg087.

[14] Eickholt, J.—Deng, X.—Cheng, J.: DoBo: Protein Domain Boundary Prediction by Integrating Evolutionary Signals and Machine Learning. BMC Bioinformatics, Vol. 12, 2011, No. 1, Art. No. 43, doi: 10.1186/1471-2105-12-43.

[15] Liu, J.—Rost, B.: Sequence-Based Prediction of Protein Domains. Nucleic Acids Research, Vol. 32, 2004, No. 12, pp. 3522–3530, doi: 10.1093/nar/gkh684.

[16] Bondugula, R.—Lee, M. S.—Wallqvist, A.: FIEFDom: A Transparent Domain Boundary Recognition System Using a Fuzzy Mean Operator. Nucleic Acids Research, Vol. 37, 2009, No. 2, pp. 452–462, doi: 10.1093/nar/gkn944.

[17] Xue, Z.—Xu, D.—Wang, Y.—Zhang, Y.: ThreaDom: Extracting Protein Domain Boundary Information from Multiple Threading Alignments. Bioinformatics, Vol. 29, 2013, No. 13, pp. i247–i256, doi: 10.1093/bioinformatics/btt209.

[18] Chivian, D.—Kim, D. E.—Malmström, L.—Schonbrun, J.—Rohl, C. A.—Baker, D.: Prediction of Casp6 Structures Using Automated Robetta Protocols. Proteins: Structure, Function, and Bioinformatics, Vol. 61, 2005, No. S7, pp. 157–166, doi: 10.1002/prot.20733.

[19] Altschul, S. F.—Madden, T. L.—Schäffer, A. A.—Zhang, J.—Zhang, Z.—Miller, W.—Lipman, D. J.: Gapped BLAST and PSI-BLAST: A New Generation of Protein Database Search Programs. Nucleic Acids Research, Vol. 25, 1997, No. 17, pp. 3389–3402, doi: 10.1093/nar/25.17.3389.

[20] Pollastri, G.—Baldi, P.—Fariselli, P.—Casadio, R.: Prediction of Coordination Number and Relative Solvent Accessibility in Proteins. Proteins, Vol. 47, 2002, No. 2, pp. 142–153, doi: 10.1002/prot.10069.

[21] Baldi, P.—Pollastri, G.: The Principled Design of Large-Scale Recursive Neural Network Architectures – DAG-RNNs and the Protein Structure Prediction Problem. Journal of Machine Learning Research, Vol. 4, 2003, pp. 575–602, https://www.jmlr.org/papers/volume4/baldi03a/baldi03a.pdf.

[22] Yoo, P. D.—Sikder, A. R.—Zhou, B. B.—Zomaya, A. Y.: Improved General Regression Network for Protein Domain Boundary Prediction. BMC Bioinformatics, Vol. 9, Suppl. 1, 2008, doi: 10.1186/1471-2105-9-S1-S12.

[23] Joachims, T.: Making Large-Scale Support Vector Machine Learning Practical. Advances in Kernel Methods: Support Vector Learning, MIT Press, 1999, pp. 169–184.

[24] Chakraborty, S.—Das, S.—Chatterjee, P.: Prediction of Domain Boundaries in Protein Sequences Using Predicted Secondary Structure and Physicochemical Properties of Amino Acids. 2014 International Conference on Circuits, Power and Computing Technologies (ICCPCT-2014), IEEE, 2014, pp. 1022–1026, doi: 10.1109/ICCPCT.2014.7054913.

[25] Kawashima, S.—Ogata, H.—Kanehisa, M.: AAindex: Amino Acid Index Database. Nucleic Acids Research, Vol. 27, 1999, No. 1, pp. 368–369, doi: 10.1093/nar/27.1.368.

[26] Li, B. Q.—Hu, L. L.—Chen, L.—Feng, K. Y.—Cai, Y. D.—Chou, K. C.: Prediction of Protein Domain with mRMR Feature Selection and Analysis. PLoS ONE, Vol. 7, 2012, No. 6, Art. No. e39308, doi: 10.1371/journal.pone.0039308.

[27] Joo, K.—Lee, S. J.—Lee, J.: Sann: Solvent Accessibility Prediction of Proteins by Nearest Neighbor Method. Proteins, Vol. 80, 2012, No. 7, pp. 1791–1797, doi: 10.1002/prot.24074.

[28] McGuffin, L. J.—Bryson, K.—Jones, D. T.: The PSIPRED Protein Structure Prediction Server. Bioinformatics, Vol. 16, 2000, No. 4, pp. 404–405, doi: 10.1093/bioinformatics/16.4.404.

[29] Walsh, I.—Martin, A. J. M.—Mooney, C.—Rubagotti, E.—Vullo, A.—Pollastri, G.: Ab Initio and Homology Based Prediction of Protein Domains by Recursive Neural Networks. BMC Bioinformatics, Vol. 10, 2009, No. 1, Art. No. 195, doi: 10.1186/1471-2105-10-195.

[30] Cheng, J.: DOMAC: An Accurate, Hybrid Protein Domain Prediction Server. Nucleic Acids Research, Vol. 35, 2007, No. Suppl_2, pp. W354–W356, doi: 10.1093/nar/gkm390.

[31] Schuster, M.—Nakajima, K.: Japanese and Korean Voice Search. 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012, pp. 5149–5152, doi: 10.1109/ICASSP.2012.6289079.

[32] Alammar, J.: The Illustrated BERT, ELMo, and Co. (How NLP Cracked Transfer

Learning). http://jalammar.github.io/illustrated-bert/ (Retrieved 2021-09-11).

[33] Hochreiter, S.—Schmidhuber, J.: Long Short-Term Memory. Neural Computation, Vol. 9, 1997, No. 8, pp. 1735–1780, doi: 10.1162/neco.1997.9.8.1735.

[34] Hochreiter, S.: The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, Vol. 6, 1998, No. 2, pp. 107–116, doi: 10.1142/S0218488598000094.

[35] Yi, H. C.—You, Z. H.—Cheng, L.—Zhou, X.—Jiang, T. H.—Li, X.—Wang, Y. B.: Learning Distributed Representations of RNA and Protein Sequences and Its Application for Predicting lncRNA-Protein Interactions. Computational and Structural Biotechnology Journal, Vol. 18, 2020, pp. 20–26, doi: 10.1016/j.csbj.2019.11.004.

[36] The UniProt Consortium: Protein Information Resource. European Bioinformatics Institute, SIB Swiss Institute of Bioinformatics, https://www.uniprot.org/uniref/?query=uniprot.

[37] Kudo, T.: Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates. In: Gurevych, I., Miyao, Y. (Eds.): Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). ACL, 2018, pp. 66–75, doi: 10.18653/v1/P18-1007.

[38] Kudo, T.—Richardson, J.: SentencePiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing. In: Blanco, E., Lu, W. (Eds.): Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP 2018). ACL, 2018, pp. 66–71, doi: 10.18653/v1/D18-2012.

[39] Bateman, A.—Coin, L.—Durbin, R.—Finn, R. D.—Hollich, V.—Griffiths-Jones, S.—Khanna, A.—Marshall, M.—Moxon, S.—Sonnhammer, E. L. L.—Studholme, D. J.—Yeats, C.—Eddy, S. R.: The Pfam Protein Families Database. Nucleic Acids Research, Vol. 32, 2004, No. Suppl_1, pp. D138–D141, doi: 10.1093/nar/gkh121.

[40] Dawson, N. L.—Lewis, T. E.—Das, S.—Lees, J. G.—Lee, D.—Ashford, P.—Orengo, C. A.—Sillitoe, I.: CATH: An Expanded Resource to Predict Protein Function Through Structure and Sequence. Nucleic Acids Research, Vol. 45, 2017, No. D1, pp. D289–D295, doi: 10.1093/nar/gkw1098.

[41] The UniProt Consortium: UniProt: A Worldwide Hub of Protein Knowledge. Nucleic Acids Research, Vol. 47, 2019, No. D1, pp. D506–D515, doi: 10.1093/nar/gky1049.

[42] Fu, L.—Niu, B.—Zhu, Z.—Wu, S.—Li, W.: CD-HIT: Accelerated for Clustering the Next-Generation Sequencing Data. Bioinformatics, Vol. 28, 2012, No. 23, pp. 3150–3152, doi: 10.1093/bioinformatics/bts565.

[43] Wang, D.—Zeng, S.—Xu, C.—Qiu, W.—Liang, Y.—Joshi, T.—Xu, D.: MusiteDeep: A Deep-Learning Framework for General and Kinase-Specific Phosphorylation Site Prediction. Bioinformatics, Vol. 33, 2017, No. 24, pp. 3909–3916, doi: 10.1093/bioinformatics/btx496.

Ahmad Haseeb received his M.Sc. degree in computer science in 2021 from the National University of Computer and Emerging Sciences, Pakistan. His current research interests include machine learning, and natural language processing.

Maryam Bashir is an Assistant Professor of computer science at the National University of Computer and Emerging Sciences. She earned her doctorate in computer science from the Northeastern University in Boston, USA. She is recipient of prestigious Fulbright Scholarship for Ph.D. studies in the USA. Her research interests include information retrieval, natural language processing, and evolutionary algorithms.

Aamir Wali has been teaching at the Department of Computer Science, FAST-National University of Computer and Emerging Sciences since 2004. He has his Ph.D. in computer science from the same university. His areas of interest include font development, writing systems, machine learning, image processing, human-computer interaction and virtual/augmented reality.