IGPRED-MultiTask: A Deep Learning Model to Predict Protein Secondary Structure, Torsion Angles and Solvent Accessibility

Yasin Görmez and Zafer Aydin

Abstract—Protein secondary structure, solvent accessibility and torsion angle predictions are preliminary steps to predict 3D structure of a protein. Deep learning approaches have achieved significant improvements in predicting various features of protein structure. In this study, IGPRED-Multitask, a deep learning model with multi task learning architecture based on deep inception network, graph convolutional network and a bidirectional long short-term memory is proposed. Moreover, hyper-parameters of the model are fine-tuned using Bayesian optimization, which is faster and more effective than grid search. The same benchmark test data sets as in the OPUS-TASS paper including TEST2016, TEST2018, CASP12, CASP13, CASPFM, HARD68, CAMEO93, CAMEO93_HARD, as well as the train and validation sets, are used for fair comparison with the literature. Statistically significant improvements are observed in secondary structure prediction on 4 datasets, in phi angle prediction on 2 datasets and in psi angel prediction on 3 datasets compared to the state-of-the-art methods. For solvent accessibility prediction, TEST2016 and TEST2018 datasets are used only to assess the performance of the proposed model.

Index Terms—Feature extraction or construction, machine learning, protein structure predicition, bioinformatics, deep learning

1 Introduction

Proteins are the building blocks of living organisms. Understanding the function of proteins is one of the main goals of biological sciences. Because there is a strong relationship between the 3-D structure of a protein and its function, protein structure determination has been studied for many years.

Solving the 3-D structure of a protein using experimental methods such as NMR, Cryo-EM and X-ray crystallography is expensive and time consuming. Because of this reason computational estimation of protein structure has been considered as an efficient alternative. As a preliminary step for predicting the 3-D structure, information about structural properties such as secondary structure, torsion angles and solvent accessibility is widely used, which provides important constraints about the full 3-D structure. Therefore, accurate prediction of these properties is an important step that is typically achieved by a machine learning algorithm trained using a dataset of experimentally solved structures.

Studies on improving the accuracy of machine learning based methods have concentrated on two major directions: extracting better input features and developing more advanced model architectures. The first methods developed for protein secondary structure prediction (PSSP) used amino acid sequence only and reached a Q3 accuracy of 60%. Subsequently, multiple sequence alignment algorithms were used to extract input features and the Q3 accuracy increased to 80-82% [1], [2]. When structural profiles were also used as input, the Q3 accuracy reached to 84-85% [3], [4]. Aydin *et al.* systematically developed structural profiles for different sequence identity cutoff rates and obtained 83-84% accuracy for distant templates and 89-91% for close templates [5]. Studies show that using various profiles which come from several alignment algorithms improves the accuracy of classification [1], [6].

Related to the second category of methods, to date, many machine learning models have been developed for PSSP. Jones proposed position specific scoring matrices (PSSM) and neural network achieving a Q3 accuracy of 78.3% on the CASP3 dataset [7]. Faraggi *et al.* improved Q3 accuracy to 83.8% with the multistep neural network algorithm [8]. Magnan and Baldi obtained 80% Q3 accuracy with bi-directional recurrent neural networks [9]. Drozdetskiy *et al.* proposed neural network architecture with bootstrap framework and obtained a Q3 accuracy of 82% [10]. Wang *et al.* proposed deep convolutional neural fields called DeepCNF and obtained Q3 accuracies of 85.4%, 82.3%, 84.4%, 84.7% and 84.5% for CullPDB, CB513, CASP10, CASP11 and CAMEO datasets, respectively [11]. Heffernan *et al.* proposed long short-term memory bidirectional recurrent neural networks and obtained a Q3 accuracy of 84.48% on TS1199 dataset [12]. Aydin *et al.* showed that dimension reduction and feature selection algorithms increased the performance of PSSP [13]. Fang *et al.* proposed a deep inception inside inception neural network for PSSP and achieved Q3 accuracies of 85.98%, 83.59%, 80.59% on CASP10, CASP11 and CASP12 datasets, respectively [14]. Torrisi *et al.* applied an ensemble model

- *Yasin Görmez is with Management Information System, Sivas Cumhuriyet University, Sivas 58050, Turkey. E-mail: yasingormez@cumhuriyet.edu.tr.*
- *Zafer Aydin is with Computer Engineering Department, Abdullah Gül University, Kayseri 38080, Turkey. E-mail: zafer.aydin@agu.edu.tr.*

Manuscript received 24 November 2021; revised 23 May 2022; accepted 8 July 2022. Date of publication 18 July 2022; date of current version 3 April 2023. (Corresponding author: Yasin Görmez.)
Digital Object Identifier no. 10.1109/TCBB.2022.3191395

which uses convolutional neural networks and cascaded bi-directional recurrent neural networks. They obtained Q3 accuracies of 85.48% and 82.89% on CAMEO and CASP13 datasets, respectively [15]. Klausen *et al.* achieved 82.4%, 85.4% and 85.7% as the Q3 accuracies on CASP12, CB513 and TS115 datasets, respectively with a model that uses a combination of convolutional and long short-term memory networks [16]. Hanson *et al.* obtained Q3 accuracies of 85.99%, 86.18% and 89.06% for TEST2016, TEST2018 and CAMEO93 datasets, respectively using an ensemble of recurrent and residual convolutional neural networks [17]. Kumar *et al.* achieved 85.4%, 83.7%, 81.5% 85.4% as the Q3 accuracies on CASP10, CASP11, CB513 and CB6133 datasets, respectively using a combination of convolutional and bi-directional recurrent neural networks [18]. Zhou *et al.* obtained 82.1% and 84.8% as the Q3 accuracies on CB513 and CullPDB datasets, respectively with a model that also uses a combination of convolutional and long short-term memory networks [19]. Xu *et al.* proposed an ensemble of convolutional neural networks, transformers layer and bi-directional long short-term memory achieving Q3 accuracies of 87.79%, 86.64%, 89.06%, 85.47% on TEST2016, TEST2018, CAMEO and CASP12 datasets, respectively [20]. Uddin *et al.* obtained 77.73%, 76.09%, 74.78%, 74.17% and 72.25% as the Q8 accuracies using a model based on self-attention module called SAINT on TEST2016, TEST2018, CASP13, CASP12 and CASP-FM datasets, respectively [21].

Protein solvent accessibility (SA) prediction is another popular task for protein structure prediction (PSP). Thompson and Goldstein introduced a Bayesian probabilistic method for 2-state SA prediction and achieved a 75% accuracy [22]. Li and Pan developed a novel method for the SA threshold of 20% and reached a 75.3% accuracy with a correlation coefficient of 0.44 [23]. Naderi Manesh *et al.* obtained better than 70% accuracy for 2-state and 60% for 3-state prediction using an information theory based system [24]. Ahmad and Gromiha applied neural networks for several thresholds of SA and reached an 88% accuracy in 2-state prediction [25]. Yuan *et al.* obtained 70.1% accuracy for single sequence input and 73.9% accuracy for multiple sequence alignment input using support vector machines (SVM) [26]. Ahmad and Gromiha showed that neural networks are better than existing models to predict accessible surface area (ASA) [27]. Ahmad *et al.* presented a neural network model to predict ASA based on neighborhood information and showed that 23.7 mean absolute error (MAE) can be obtained even when no information about neighbors is included [28]. Adamczak *et al.* obtained a 15.3–15.8 MAE with proposed recurrent neural network-based regression model for relative solvent accessibility (RSA) prediction. They showed that, their proposed model outperformed the existing classification methods when predictions are converted into 2-state classes [29]. Kim and Park proposed SVM and PSSM for RSA. They also introduced a three-dimensional local descriptor that contains information about the expected remote contacts by both the long-range interaction matrix and neighbor sequences. They obtained 78.7%, 80.7%, 82.4% and 87.4% accuracies in 2-state classification for the accessibility thresholds of 25%, 16%, 5%, and 0%, respectively [30]. Nguyen and Rajapakse also used SVM for RSA prediction and obtained 90.4% and 90.2% accuracies on the Manesh and RS126 datasets respectively [31]. Sim *et al.* obtained 64.1% accuracy in 3-state SA prediction for thresholds of 9% for buried/intermediate and 36% for intermediate/exposed, respectively. They also obtained accuracies of 86.7%, 82.0%, 79.0% 78.5% in 2-state SA prediction (i.e., buried/exposed) for thresholds of 0, 5, 16 and 25%, respectively using fuzzy k-nearest neighbor method [32]. Faraggi *et al.* introduced guide learning for RSA prediction, which reduces MAE by 2-4 [33]. Joo *et al.* proposed a nearest neighbor method for SA prediction and obtained 80.89% and 67.58% accuracies on CASP8 datasets for 2-state and 3-state classifications, respectively. They also showed that increasing the dataset size effects accuracies positively [34]. Mirabello and Pollastri obtained 80% accuracy for SA prediction using bi-directional recurrent neural networks [2]. Deng *et al.* obtained 68% accuracy in 3-state solvent accessibility prediction on CASP11 dataset using a stacked autoencoder based deep learning model [35]. Zhang *et al.* proposed stacked deep bi-directional recurrent neural network to capture long-range interactions and obtained an 8.8 and 8.2 MAE scores on CB502 and Manesh215 datasets, respectively [36]. Kaleel *et al.* obtained 80% accuracy in 2-state SA prediction with proposed method based on a combination of convolutional neural networks and a bidirectional recurrent neural network [37].

Torsion angle prediction (TAP) is also one of the important steps for PSP. Kuang *et al.* showed that SVM and neural networks have improved the accuracy for three and four state TAP [38]. Keskin *et al.* showed that short-range interactions and long-range interactions affects the torsion angles considerably [39]. Wu and Zhang pro-posed a composite machine learning algorithm based on neural networks to predict real-valued torsion angles and obtained a MAE of 10 [40]. Xue *et al.* showed that real-valued prediction of torsion angles are more useful than discrete-valued prediction of torsion angles for PSP [41]. Cheung *et al.* obtained 82.1% accuracy for 4-state TAP, which uses Bayesian inference [42]. Lyons *et al.* achieved 9 and 35 degrees of MAE using stacked sparse deep autoencoder model for predicting phi and psi angles, respectively [43]. Heffernan *et al.* improved the accuracies of TAP, SA and PSSP by using iterative deep learning [44]. Li *et al.* achieved 20 and 29 degrees of MAE for phi and psi angles, respectively using four different deep learning architectures [45]. Gao *et al.* showed that grid based deep neural networks can obtain 2-6% higher accuracy for TAP [46]. Fang *et al.* proposed deep residual inception neural networks for TAP and obtained 5 and 2 degrees of im-provement for psi and phi angles, respectively [47]. Gao *et al.* obtained 18.32 MAE for phi and 27.15 MAE for psi angle using a combination of deep learning and clustering [48]. Mataeimoghadam *et al.* achieved a significantly more accurate MAE score than an existing system using fewer features and simpler neural networks [49]. Xu *et al.* proposed a sampling based post-processing method and they showed that this method can increase the accuracy of prediction models [50].

Considering the studies in the literature, deep learning methods are used effectively to predict structural properties of proteins. In addition, due to strong relationship between different types of structural properties, multi-task learning, which predicts multiple structural properties simultaneously,

is also shown to improve prediction accuracy [20], [51]. Furthermore, in recent studies on 3D structure prediction, interactions between amino acid pairs are predicted (e.g., as contact maps or distance maps) and employed as input to more sophisticated energy minimization algorithms [52]. Distances between pairs of amino acids are effective to determine the severity of these interactions. If two amino acids are close to each other in 3D space (i.e., once the protein folds into its structure), they may potentially interact even if these amino acids are far from each other in sequence. Though utilized successfully for predicting 3D structure of proteins, there is limited work that employs interaction information between amino acid pairs for predicting structural properties of proteins [17].

In this paper, a novel deep learning model that includes convolutional neural networks (CNN), graph convolutional networks (GCN) and bi-directional long short-term memory (biLSTM) is proposed. Convolutional neural networks used in this study employ inception modules and 1D convolutions, which is an effective technique for processing time-series data. GCN allows the incorpo-ration of pairwise amino acid interactions as input by representing the protein sequence as a graph. BiLSTM is a recurrent neural network architecture that allows capturing long-range interactions between amino acids of a protein. At the output layer, secondary structure (SS), RSA, and torsion angles (TA) are predicted using a multi-task strategy. Bayesian optimization technique is used to optimize the hyper-parameters of the proposed model. Sequence profiles, physico-chemical properties of amino acids, structural profiles and a no seq label are used as input features. The novelties of our model include the use of GCN jointly with convolutional and recurrent neural network architectures and utilization of structural profile information along with sequence profiles and physico-chemical properties.

2 Materials and Methods

2.1 Problem Definition

In this study, machine learning models are developed for protein secondary structure prediction (PSSP), relative solvent accessibility prediction (RSA), and torsion angle prediction (TAP). PSSP aims to assign a secondary structure class to each amino acid of a protein. It can be predicted as 8-states or 3-states. In this work, the 8-state representation is transformed to 3-states. For this purpose, H, G and I are assigned to H, E and B are assigned to E and " ", S and T are assigned to L. SA is the area that is accessible to solvent such as water and RSA is the SA normalized by the maximum accessible surface area. Similar to secondary structure, SA and RSA information is derived for each amino acid separately. It can be predicted as a real-valued quantity or it can be categorized and predicted as a discrete label. The present work predicts real-valued RSA for each amino acid. TA, also known as dihedral angle, is the angle between two successive backbone planes. Proteins have three types of backbone dihedral angles: phi, psi and omega. The omega angle is generally close to 180° (trans case) or 0° (cis case). Therefore real-valued phi and psi angles are commonly predicted for each amino acid, which is also performed in this paper.

TABLE 1
Number of Proteins and Amin Acids in Benchmark Datasets

Dataset	Number of Proteins	Number of Amino Acids
TEST2016	1212	287733
TEST2018	250	50889
CAMEO93	93	22901
CAMEO93_HARD	15	4375
CASP12	55	10283
CASP13	32	5354
CASPFM	56	8100
HARD68	45	6447
VALİDATION	983	215803
TRAIN	10042	2235849

2.2 Benchmark Datasets

In this study, a total of 10 benchmark datasets are used, which were also used in the OPUS-TASS [20] paper and are downloaded from the link (https://github.com/thuxugang/opus_tass). The name of the datasets are TEST2016, TEST2018, CAMEO93, CAMEO93_HARD, CASP12, CASP13, CASPFM, HARD68, validation and train. Eight of them, which include TEST2016, TEST2018, CAMEO93, CAMEO93_-HARD, CASP12, CASP13, CASPFM, HARD68, are used as test sets to assess the performance of the proposed model and compare with the state-of-the-art by doing a single round of model training and testing. Training set is used for model training and validation set is used as test data for hyper-parameter optimization, which require repeated rounds of model training and testing for each hyper-parameter configuration. Note that test sets are employed after the optimum hyper-parameters are found. The number of proteins and amino acids for benchmark datasets are given in Table 1.

In the experimental phase, the proposed model is tested on each test set individually (i.e., test sets are not combined during evaluation). These are the test data sets used in the OPUS-TASS paper [20] and were derived by choosing recent proteins deposited to protein data bank (PDB) by applying some selection criteria so that test proteins are not too similar to train proteins or those that are released in CASP competitions or those that are formed to include difficult targets. That's why we used the same data sets for fair comparison with the state-of-the-art. The percentages of the sample size (i.e., amino acids) in test sets to training set are 12.87%, 2.28%, 1.02%, 0.20%, 0.46%, 0.24%, 0.36%, 0.29% and 9.65% for TEST2016, TEST2018, CAMEO93, CAMEO93_HARD, CASP12, CASP13, CASPFM, HARD68 and validation data sets respectively. Although these percentanges seem to be small for most of the test sets, the smallest set contains sufficient number of samples (i.e., 4375 amino acids) to statistically evaluate the model performance because it is evaluated at amino acid level.

In addition to the original training set, a separate training set is also derived for each test set to make the experimental conditions even more difficult. For this purpose, pairwise BLAST alignments with a stringent E-value cut-off of 0.05 are performed between the original training set of 10042 proteins and the test sets or a union of test sets such as the union of CASP sets. Then, for each test set or test set group, training set proteins for which the E-value of the alignment

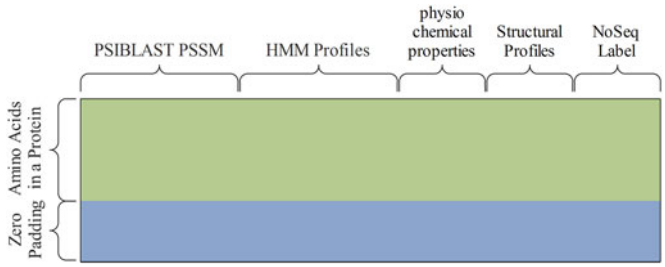

Fig. 1. Feature matrix representation for a protein.

is less than 0.05 are eliminated and the remaining set of proteins is taken as the training set specialized for that test set (or test set group). The number of proteins and amino acids in the reduced versions of the training sets is tabulated below.

In this table, for instance, Train-TEST2016 represents the training set derived for TEST2016. The same training set is derived both for CAMEO93 and CAMEO93_HARD and another single training set for CASP12, CASP13, CASPFM, and HARD68. CAMEO93 contains CAMEO93_HARD as a subset. Therefore CAMEO93 is used as the test set for the BLAST alignments to derive the training sets both for CAMEO93 and CAMEO93_HARD. Similarly, the union of CASP12, CASP13, and CASPFM is taken as the test set for the BLAST alignments to derive the training sets for CASP12, CASP13, CASPFM, and HARD68. Note that the majority of the proteins in HARD68 set are CASP proteins (a total of 45). Therefore for simplicity its training set is computed using CASP proteins as test set in BLAST alignments.

2.3 Feature Generation

Deriving a representative feature set is one of the most important steps for accurate estimation of structural elements of a protein. Based on the studies in the literature, combining different types of features has been shown to improve the accuracy of prediction. In this study, a rich feature set is derived for each amino acid, which contains 20 scores from PSIBLAST alignments, 30 scores from HHblits alignments, 7 scores reflecting physico-chemical properties, 5 scores from structural profiles, and a noseq label as shown in Fig. 1.

This type of feature set was originally proposed by Fang *et al.* [14]. The set of 20 PSIBLAST features are taken as the corresponding column of the position specific scoring matrix (PSSM) generated using PSIBLAST, where each target protein is aligned with the proteins in the NR database (dated as July 2020). In PSIBLAST alignments, the number of iterations is set to 3, e-value threshold to 10 and inclusion threshold to 0.001. The 30 features obtained from HHblits alignments include 20 PSSM-based features, 7 scores related to transition probabilities and 3 entropy-based features. These features are derived from the HMM-profile model obtained by running HHblits for each target against the Uniclust30 database. Details for computing the 30 features from HHblits alignments can be found in Gormez *et al.* [53]. Structural profiles consist of three values of secondary structure and two values of solvent accessibility scores for each amino acid. These structural profiles were computed using the HHBlits alignment algorithm. Details of structural profile computation for secondary structure classes can be found in Gormez *et al.* [53] with the only exception that Uniclust30 database is used in the first stage for sequence alignment instead of NR20 database. The computation of structural profiles for solvent accessibility labels is similar, which uses solvent accessibility labels of the template proteins. A total of 7 physico-chemical features are used to represent each amino acid including volume of side chains, polarity, polarizability, hydro-philicity, hydrophobicity, net charge index of side chains and solvent accessible surface area. In addition to those features, a noseq label feature is included to denote whether a given row of feature matrix contains feature data (a noseq label of zero) or not (a noseq label of one). This feature is included because deep learning models are designed to take a fixed sized input in time dimension (i.e., they expect the number of amino acids to be the same). If the number of amino acids in a target protein is less than the sequence length parameter (which is set to 700), zero padding is applied for the remaining time steps and the noseq label will be set to 1 for zero padded section. Otherwise, it will be equal to 0. If the number of amino acids in a protein is more than 700, the amino acid sequence will be divided into multiple parts where each part contains 700 amino acids (with the last part completed into 700 amino acids by zero padding if necessary). Then each part is treated as a separate protein. Details of how noseq label is assigned including cases in which the number of amino acids in target is greater than the sequence length parameter is explained in Gormez et al [53]. Fig. 1 shows the feature data matrix for a protein with less than 700 amino acids (with zero padding applied). In this matrix, columns represent features and rows represent amino acids. Each protein corresponds to a data sample in each mini-batch of neural network model training.

2.4 Proposed Model

In this study, a novel multi-task architecture based on convolutional neural networks (CNN), graph convolutional networks (GCN) and recurrent neural networks with bidirectional long short-term memory (BiLSTM) was proposed. Several GCN, CNN, BiLSTM and fully connected layers are connected to each other in different ways. At the end of the model there are 3 output layers: a fully connected layer with softmax activation function to predict secondary structure, a fully connected linear layer with one node to predict solvent accessibility and a fully connected linear layer with two nodes to predict phi and psi angels. Fig. 2 summarizes the architecture of this model.

Each CNN module of this architecture consists of 5 different 1-D convolutional layers fed in parallel with kernel sizes (1,M), (3,M), (5,M), (9,M), and (15,M) where M represents number of features derived for each amino acid. Note that this is a form of an inception network. Except for the kernel sizes, all convolutional layers in the same module are identical. Four operations are applied to each layer in sequential order: convolution operation, batch normalization layer, activation layer with ReLu function and dropout. The outputs obtained from each dropout operation are concatenated to form the output of a CNN module, the architecture of which is depicted in Fig. 3 of Gormez and Aydin [53].

A GCN module of our model consist of two inputs. These are feature data matrix and a graph representing interactions

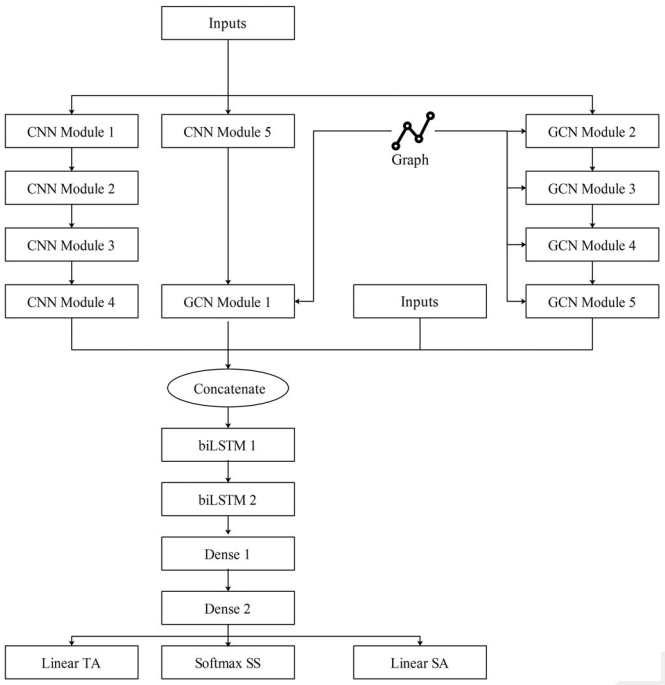

Fig. 2. The proposed model architecture.

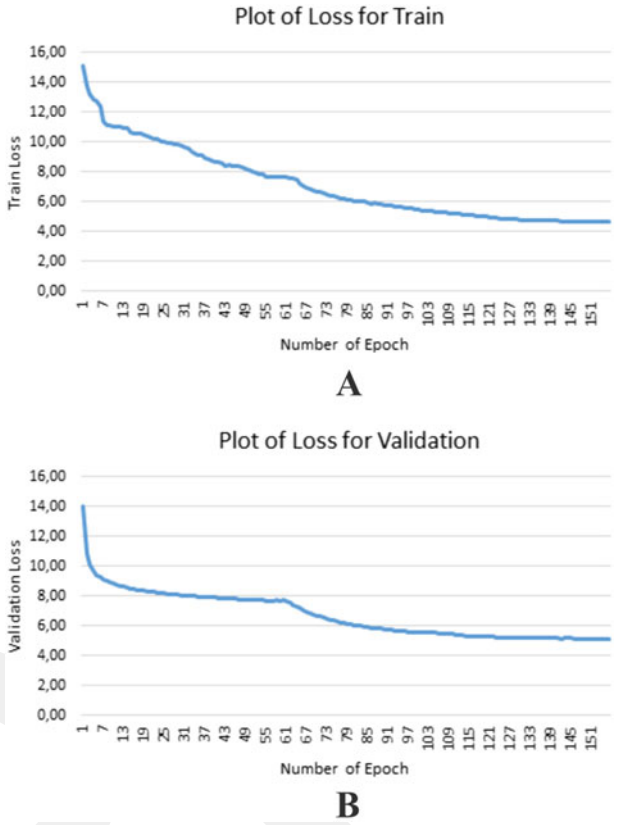

Fig. 3. Loss curves for IGPRED-MultiTask. (A) Train loss (B) Validation loss.

between amino acid pairs. In each GCN module, a multi-graph convolutional layer (mGCN) is used as the model architecture [54], followed by batch normalization and a dropout layer. In this paper, we assume that amino acids in different protein chains do not interact with each other. Therefore, a graph is constructed separately for each protein as an additional input feature matrix, which is sent as input to mGCN layers. For this purpose, an unweighted adjacency matrix W having dimension N $\times$ N is generated for each protein, where N represents the length of the amino acid sequence. W_{ij} is the $(i,\ j)^{th}$ element element of W with i being the row index and j denoting the column index. $W_{ij} = 1$, if there is an interaction between the i^{th} and the j^{th} amino acids, and $W_{ij} = 0$ otherwise. This matrix is also symmetric, (i.e., $W_{ij} = W_{ji}$). In this paper, short-range interactions are considered only when constructing the adjacency matrix. An amino acid is assumed to interact with its local neighbors only. The neighbors include those amino acids within the symmetric window taken around the central amino acid of interest. To represent the neighborhood relationship, the number of connections (nconn) parameter is used, which denotes the number of interactions an amino acid makes with its neighbors on each side (including nconn amino acids that come after and nconn amino acids that come before). Once the adjacency matrix is constructed, for each dataset, a tensor of size M $\times$ N $\times$ N is generated and sent as input to the GCN module, where M represents the number of proteins in the dataset and N is the maximum number of allowed amino acids in a protein, which is set to 700.

In addition to CNN and GCN layers, our model also contains BiLSTM layers. This is due to the fact that each amino acid interacts with its local neighbors that come before and after this amino acid (i.e., the interactions are two-sided). As in the CNN module, in each BiLSTM module, the three operations, which include batch normalization, activation layer with ReLu function and dropout were followed after BiLSTM layers in sequential order. In each layer return_sequences parameter was set to true, thus long-range interaction information can be captured and transferred to the next layers.

As can be seen in the model architecture, three parallel blocks are concatenanted at the end of the first part of the proposed model. The first block only consists of CNN modules, the third block only consists of GCN modules and the second block consist of both CNN and GCN modules. By this way, amino acid features are embedded using only CNN modules, only GCN modules and a combination of them. It can be anticipated that faulty prediciton of a block may be fixed by the others. Several experiments have been performed to analyze the effect of each module. The results of these experiments are shared in the supplementary document.

After the BiLSTM modules, two dense layers and three output layers follow implementing a multi-task architecture. There is an output layer for each of the prediction tasks including the prediction of torsion angle, secondary structure and solvent accessibility. However, for the datasets, which do not have solvent accessibility information, the corresponding output layer is removed. A softmax layer with sparse_categorical_crossentropy loss function is used to estimate 3-state secondary structure information. Linear layers with mean_absolute_error loss function are used to estimate real-valued torsion angle and solvent accessibility information. Adam is used as the optimization algorithm for estimating the weight coefficients of neural networks, where beta_1 parameter is set to 0.95 and beta_2 parameter to 0.99.

TABLE 2
Number of Proteins in Reduced Versions of the Training Set

Reduced training set	Number of proteins	Number of amino acids
Train-TEST2016	8850	1877909
Train-TEST2018	9903	2195453
Train-CAMEO93	10024	2235849
Train-CAMEO93_HARD	10024	2235849
Train-CASP12	9936	2208407
Train-CASP13	9936	2208407
Train-CASPFM	9936	2208407
Train-HARD68	9936	2208407
Train-Val	8999	1952387

TABLE 3
Hyper-parameter Ranges Used for Optimization

Parameter	Lowest	Highest
learning rate	10^{-6}	10^{-1}
n_filters_conv	20	200
batch size	2^0	2^7
epoch	10	200
dropout rate	0	0.6
n_denses	100	1500
nconn	0	75
out_dim_gcn	20	200
n_unit_lstm	20	200

2.5 Hyper-Parameter Optimization

In this study, it is a well known fact that, choosing the right hyper-parameters is important for a machine learning model to perform accurately. Therefore hyper-parameters of a machine learning model are optimized by selecting different value combinations iteratively and choosing the particular combination that gives the highest performance. In grid search optimization, a parameter grid that contains a finite set of values for each hyper-parameter is defined, and then optimum hyper-parameter configuration is found by evaluating the performance of the model for each value combination. One disadvantage of grid search is such that some of the intermediate values that are not represented in the parameter grid (and hence missed) may indeed cause the model to perform better if they were included in the parameter grid. To overcome this limitation, a denser parameter grid should be selected. However, the computational cost grows exponentially as the parameter grid contains more and more values. Considering the fact that the proposed models contain many hyper-parameters, grid search would not be the best approach to optimize them. Therefore in this article, the hyper-parameters of the proposed neural network models are optimized using the Bayesian optimization algorithm. Studies show that Bayesian optimization technique outperforms the traditional optimization algorithms [55], [56]. It is also advantageous due to the fact that it can sample intermediate values in the parameter ranges. Table 3 shows the lowest and highest values of the hyper-parameters that are optimized.

In this table, *n_filters_conv* represents the number of filters for CNN layers, *n_denses* represents the number of hidden units for fully connected dense layers, *out_dim_gcn* represents the number of output units for GCN layers and *n_unit_lstm* represents the number of units for BiLSTM layers.

3 Experiments and Results

In this study, the proposed deep learning model based on CNN, mGCN and BiLSTM is employed to predict secondary structure, solvent accessibility and torsion angles for each amino acid of a given protein. For this purpose, a total of ten benchmark datasets are used. The training set is used to train models, the validation set is used as a test set to optimize hyper-parameters and the remaining eight sets are used to measure performance of the proposed model. As explained in Section 2.5, a total of nine hyper-parameters are optimized within the specified parameter ranges. A separate parameter value is defined and optimized for the number of filters for each of the five CNN modules, the number of output dimensions for the mGCN modules, the number of units for the two BiLSTM modules and the number of hidden neurons for the two fully connected dense layers. Table 2 shows the optimum values of the hyper-parameters that are found using Bayesian optimization. These values are then used to train the neural network models. In this table, the values of *n_filters_conv*, *n_dense*, *out_dim_gcn* and *n_unit_lstm* are shown for each module following their sequential order in Fig. 2. For instance, a total of five *n_filters_conv* parameters are defined and optimized for CNN Modules 1-5 and the optimized values are presented in this order in Table 4 (i.e., 175 is the optimum value for this parameter for CNN Module 1).

After hyper-parameter optimization, the proposed model is trained on the original training set and predictions are computed on a total of eight test sets (see Section 2.2 for data sets). Accuracy (ACC) for 3-state secondary structure and mean absolute error (MAE) for torsion angles were used in the literature to evaluate the model performance [14], [19], [20], [40], [45]. Therefore, the results of these experiments are summarized in Table 5, in which ACC for 3-state secondary structure and MAE for torsion angles are computed on eight test sets. Note that for each benchmark in Table 5, a single train and a single test operation are performed. In these experiments, we obtain testing results on multiple independent data sets, which provides an estimate of the variation in performance results with respect to different data set conditions including the difficulty of the dataset.

TABLE 4
Optimum Hyper-Parameters for the Proposed Deep Learning Model

Hyper-parameter types	Optimum hyper-parameters
learning rate	0.000210060076
n_filters_conv	175, 86, 64, 112, 125
batch size	2^6
epoch	156
dropout rate	0.2
n_dense	675, 280
nconn	16
out_dim_gcn	96, 81, 79, 32, 28
n_unit_lstm	58, 30

TABLE 5
Comparison of Our Model with the State-Of-The-Art Methods for Secondary Structure and Torsion Angle Predictions

Models	Accuracy SS3	MAE psi	MAE phi
		TEST2016	
SPOT-1D[a]	87.16%	16.27	23.26
OPUS-TASS	87.79%	15.78	22.46
IGPRED-MultiTask*	87.98%	15.13	22.29
IGPRED-MultiTask	**88.29%**	**15.04**	**21.81**
		TEST2018	
MUFOLD[a]	84.78%	17.78	27.24
NetsurfP-2.0[a]	85.31%	17.90	26.63
SPOT-1D[a]	86.18%	16.89	24.87
OPUS-TASS	86.84%	16.40	24.06
IGPRED-MultiTask*	87.35%	15.85	23.37
IGPRED-MultiTask	**87.64%**	**15.76**	**23.22**
		CASP12	
MUFOLD	83.36%	——	——
SPOT-1D [a]	84.82%	18.44	26.90
OPUS-TASS	85.47%	18.08	25.98
IGPRED-MultiTask*	86.57%	17.63	24.81
IGPRED-MultiTask	**86.61%**	**17.57**	**24.78**
		CASP13	
SPOT-1D [a]	86.53%	18.48	26.73
OPUS-TASS	87.62%	17.89	25.93
IGPRED-MultiTask*	88.27%	17.09	24.61
IGPRED-MultiTask	**88.41%**	**17.05**	**24.47**
		CASFM	
SPOT-1D [a]	82.37%	19.39	30.10
OPUS-TASS	83.40%	18.85	28.00
IGPRED-MultiTask*	84.18%	18.29	27.21
IGPRED-MultiTask	**84.24%**	**18.27**	**27.17**
		CAMEO93	
SPOT-1D [a]	87.72%	16.89	23.02
OPUS-TASS	89.06%	16.56	21.98
IGPRED-MultiTask*	89.27%	**16.19**	21.74
IGPRED-MultiTask	**89.28%**	16.25	**21.65**
		CAMEO93 HARD	
SPOT-1D [a]	82.31%	18.75	31.02
OPUS-TASS	82.56%	18.52	30.17
IGPRED-MultiTask*	**84.11%**	17.60	**27.57**
IGPRED-MultiTask	84.09%	**17.58**	27.64
		HARD68	
SPOT-1D [a]	83.79%	18.35	27.77
OPUS-TASS	83.78%	18.03	27.16
IGPRED-MultiTask*	84.61%	17.51	26.54
IGPRED-MultiTask	**84.81%**	**17.38**	**26.32**

[a]*Results are taken from the paper of the OPUS-TASS method.*

In this table, IGPRED-MultiTask is our proposed model and is trained using the original training set. IGPRED-MultiTask* represents our proposed model trained on the reduced versions of the original training set (see Section 2.2), which is a more difficult experimental setting. Our results are compared with the state-of-the-art methods OPUS-TASS [20], SPOT-1D [17], NetsurfP-2.0 [16] and MUFOLD [14], whenever possible for secondary structure and torsion angle predictions. Based on these results, our model (both IGPRED-MultiTask* and IGPRED-MultiTask) outperforms all the other state-of-the-art methods in all test sets and in all performance metrics. Note that IGPRED-MultiTask has the same experimental conditions as the other state-of-the-art methods (in terms of the training set used). Therefore, it is more convenient to compare IGPRED-MultiTask directly with the state-of-the-art. On the other hand, it is promising to observe that IGPRED-MultiTask* is also better than the state-of-the-art. Since IGPRED-MultiTask* is evaluated on the reduced training sets derived for each test set, the results obtained for IGPRED-MultiTask* can be regarded as the actual performance of our proposed model in the most stringent experimental conditions.

TABLE 6
P-Values Between IGPRED-MultiTask* and OPUS-TASS

Dataset	SS	PHI	PSI
TEST2016	0.001	0.001	0.204
TEST2018	0.045	0.030	0.016
CASP12	0.017	0.357	0.047
CASP13	0.307	0.293	0.131
CASPFM	0.161	0.342	0.280
CAMEO93	0.603	0.342	0.720
CAMEO93 HARD	0.029	0.213	0.005
HARD68	0.2187	0.412	0.441

The reason for the improved performance over the state-of-the-art can be due to the following two factors. The first one can be related to the model architecture, which utilizes deep learning models including CNN, mGCN and BiLSTM modules jointly. For instance, one difference between our model and OPUS-TASS is the utilization of mGCN modules by our model, which is also not present in other state-of-the-art methods. The second can be due to the structural profile features employed as input to our model, which may have provided additional useful information. These factors are further analyzed in Supplementary Section in more detail.

As explainined before IGPRED-MultiTask is trained on original training set and predicitons are computed on test sets using this model. Fig. 3 shows the loss curves of IGPRED-MultiTask for secondary sturucture prediction on training set and validation set. These curves show the loss until the optimum number of epochs after which the training is stopped. According to these figures, model training is performed successfully reaching the optimum validation loss. For the IGPRED-MultiTask* the behavior of losses is observed to be similar to IGPRED-MultiTask.

Based on the results presented in Table 5, the improvements obtained by IGPRED-MultiTask* over the OPUS-TASS method (which is selected as the best method among state-of-the-art methods) are statistically significant according to a two-tailed Z-test at $p <= 0.05$ for TEST2016, TEST2018, CASP12 (excluding the phi angle predictions), and CAMEO93_HARD (excluding the phi angle predictions). The results on the remaining test sets or prediction tasks can be regarded as comparable. Table 6 shows the p-values that are computed for the Z-test experiment that compares IGPRED-MultiTask* and OPUS-TASS results.

Regarding solvent accessibility, RSA labels are available in TEST2016, TEST2018, validation and training set only. Since our model can also predict the RSA information, we evaluated the solvent accessibility prediction performance of our model on TEST2016 and TEST2018 data sets. The

TABLE 7
Q3 Accuracies of IGPRED-MultiTask* for Regions

Dataset	H	E	L	mean acc	acc begin-3	Acc end-3
TEST2016	80.52%	73.77%	76.21%	76.58%	87.32%	85.71%
TEST2018	80.01%	74.12%	75.10%	76.81%	86.11%	85.43%
CASP12	79.87%	73.15%	75.22%	76.37%	86.01%	84.12%
CASP13	82.26%	73.86%	77.12%	78.12%	86.12%	83.20%
CASPFM	75.26%	69.15%	79.10%	74.70%	82.11%	80.16%
CAMEO93	83.13%	72.91%	80.01%	78.14%	87.33%	85.14%
CAMEO93 HARD	74.19%	68.94%	80.06%	74.55%	81.06%	79.87%
HARD68	75.90%	69.29%	78.56%	73.86%	81.76%	80.03%

mean absolute error of IGPRED-MultiTask* is obtained as 14.09 for TEST2016 and 15.01 for TEST2018 data sets. We are not able to compare these results with the state-of-the-art because to the best of our knowledge there is no solvent accessibility prediction result presented in the literature for TEST2016 and TEST2018.

To give some details about error regions, Q3 accuracies are calculated for the amino acid that is at the beginning and the amino acid that is at the end of secondary structural segments, which are shown in Table 7 for IGPRED-MultiTask*. This calculation is done for helix (H), strand (E) and loop (L) segments separately and also for all the segments, which is presented in "mean acc" column of Table 7. In addition to this analysis, Q3 accuracy is calculated for the amino acids that are at the beginning and at the end of the proteins in validation set. For this purpose, three aminoacids that are at the beginning and three amino acids that are at the end of each protein are selected. According to these results the proposed model has significantly lower accuracy at the terminals of secondary structure segments and slightly lower accuracy around the N-terminal and C-terminal of the proteins as compared to the accuracies obtained in Table 5. This shows that it is more difficult to predict secondary structure information at the terminals of secondary structure segments and the at the terminal regions of amino acid chains. This can be due to the fact that these are transition regions for secondary structure elements. As local windows are taken around each amino acid to form the feature vectors, the composition of those vectors at segment ends will include information from multiple segments. Furthermore, the feature vectors at the terminals of segments may be located closer to class boundaries of secondary structure elements. All these factors may make the prediction task more difficult at the terminal regions of structural segments.

In addition to evaluating the performance of our model on benchmark data sets and comparing with the state-of-the-art, we performed several other experiments to analyze the capabilities of our model in different conditions. For this purpose, we derived new versions of our original model that contain certain module blocks while excluding others. We also considered removing structural profile matrices from the input feature set. Detailed architecture of these models and their experimental results are shown in Supplementary Table 1. Based on the experimental results, it can be concluded that using all modules (including CNN, mGCN, and BiLSTM modules), multi-task learning and structural profiles have contribution in improving the accuracy of protein structure prediction tasks studied in this work. In Supplementary Table 2, we include detailed performance metrics of the proposed model for protein secondary structure prediction.

4 Conclusion

In this study, a novel deep learning model based on graph convolutional networks, convolutional neural networks and recurrent neural networks with bidirectional long short-term memory architecture that employs a multi-task learning strategy was proposed to predict secondary structure, solvent accessibility and torsion angles of a protein. The proposed model outperformed the state-of-the-art methods on all of the benchmark data sets and prediction tasks.

In this work, only short-range interactions between amino acids are considered by the graph convolutional and convolutional network layers. Although long-range interactions are also captured by recurrent neural networks to certain degree, specific interactions between amino acid pairs are not explicitly modeled by the BiLSTM layers. As a future work, long-distance interactions will also be included by feeding predicted contact maps or distance maps as inputs to graph convolutional networks.

Acknowledgments

The experiments reported in this paper were performed at Tubitak Ulakbim, High Performance and Grid Computing Center (TRUBA resources). The result of this paper will be used in the Yasin Görmez's thesis that topic is 'Developing Deep Learning Models for Protein Structure Prediction'. The IGPRED-Multitask method is available at PSP server, which can be accessed by visiting http://psp.agu.edu.tr.

References

[1] Z. Aydin, A. Singh, J. Bilmes, and W. S. Noble, "Learning sparse models for a dynamic bayesian network classifier of protein secondary structure," *BMC Bioinf.*, vol. 12, no. 1, May 2011, Art. no. 154, doi: 10.1186/1471-2105-12-154.

[2] C. Mirabello and G. Pollastri, "Porter, paleale 4.0: High-accuracy prediction of protein secondary structure and relative solvent accessibility," *Bioinformatics*, vol. 29, no. 16, pp. 2056–2058, Aug. 2013, doi: 10.1093/bioinformatics/btt344.

[3] D. Li, T. Li, P. Cong, W. Xiong, and J. Sun, "A novel structural position-specific scoring matrix for the prediction of protein secondary structures," *Bioinformatics*, vol. 28, no. 1, pp. 32–39, Jan. 2012, doi: 10.1093/bioinformatics/btr611.

[4] G. Pollastri, A. J. Martin, C. Mooney, and A. Vullo, "Accurate prediction of protein secondary structure and solvent accessibility by consensus combiners of sequence and structure information," *BMC Bioinf.*, vol. 8, no. 1, pp. 201, Jun. 2007, doi: 10.1186/1471-2105-8-201.

[5] Z. Aydin, N. Azginoglu, H. I. Bilgin, and M. Celik, "Developing structural profile matrices for protein secondary structure and solvent accessibility prediction," *Bioinformatics*, vol. 35, no. 20, pp. 4004–4010, Oct. 2019, doi: 10.1093/bioinformatics/btz238.

[6] Z. Aydin, D. Baker, and W. S. Noble, "Constructing structural profiles for protein torsion angle prediction," in *Proc. Int. Conf. Bioinf. Models Methods Algorithms*, 2015, pp. 26–35. doi: 10.5220/0005208500260035.

[7] D. T. Jones, "Protein secondary structure prediction based on position-specific scoring matrices," *J. Mol. Biol.*, vol. 292, no. 2, pp. 195–202, Sep. 1999, doi: 10.1006/jmbi.1999.3091.

[8] E. Faraggi, T. Zhang, Y. Yang, L. Kurgan, and Y. Zhou, "SPINE X: Improving protein secondary structure prediction by multistep learning coupled with prediction of solvent accessible surface area and backbone torsion angles," *J. Comput. Chem.*, vol. 33, no. 3, pp. 259–267, 2012, doi: 10.1002/jcc.21968.

[9] C. N. Magnan and P. Baldi, "SSpro/ACCpro 5: Almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity," *Bioinformatics*, vol. 30, no. 18, pp. 2592–2597, Sep. 2014, doi: 10.1093/bioinformatics/btu352.

[10] A. Drozdetskiy, C. Cole, J. Procter, and G. J. Barton, "JPred4: A protein secondary structure prediction server," *Nucleic Acids Res.*, vol. 43, no. W1, pp. W389–W394, Jul. 2015, doi: 10.1093/nar/gkv332.

[11] S. Wang, J. Peng, J. Ma, and J. Xu, "Protein secondary structure prediction using deep convolutional neural fields," *Sci. Rep.*, vol. 6, no. 1, Jan. 2016, Art. no. 1, doi: 10.1038/srep18962.

[12] R. Heffernan, Y. Yang, K. Paliwal, and Y. Zhou, "Capturing non-local interactions by long short-term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers and solvent accessibility," *Bioinformatics*, vol. 33, no. 18, pp. 2842–2849, Sep. 2017, doi: 10.1093/bioinformatics/btx218.

[13] Z. Aydin, O. Kaynar, and Y. Görmez, "Dimensionality reduction for protein secondary structure and solvent accesibility prediction," *J. Bioinform. Comput. Biol.*, vol. 16, no. 5, Aug. 2018, Art. no. 1850020, doi: 10.1142/S0219720018500208.

[14] C. Fang, Y. Shang, and D. Xu, "MUFOLD-SS: New deep inception-inside-inception networks for protein secondary structure prediction," *Proteins Struct. Funct. Bioinforma.*, vol. 86, no. 5, pp. 592–598, 2018, doi: 10.1002/prot.25487.

[15] M. Torrisi, M. Kaleel, and G. Pollastri, "Deeper profiles and cascaded recurrent and convolutional neural networks for state-of-the-art protein secondary structure prediction," *Sci. Rep.*, vol. 9, no. 1, Aug. 2019, Art. no. 1, doi: 10.1038/s41598-019-48786-x.

[16] M. S. Klausen *et al.*, "NetSurfP-2.0: Improved prediction of protein structural features by integrated deep learning," *Proteins Struct. Funct. Bioinforma.*, vol. 87, no. 6, pp. 520–527, 2019, doi: 10.1002/prot.25674.

[17] J. Hanson, K. Paliwal, T. Litfin, Y. Yang, and Y. Zhou, "Improving prediction of protein secondary structure, backbone angles, solvent accessibility and contact numbers by using predicted contact maps and an ensemble of recurrent and residual convolutional neural networks," *Bioinformatics*, vol. 35, no. 14, pp. 2403–2410, Jul. 2019, doi: 10.1093/bioinformatics/bty1006.

[18] P. Kumar, S. Bankapur, and N. Patil, "An enhanced protein secondary structure prediction using deep learning framework on hybrid profile based features," *Appl. Soft Comput.*, vol. 86, Jan. 2020, Art. no. 105926, doi: 10.1016/j.asoc.2019.105926.

[19] S. Zhou, H. Zou, C. Liu, M. Zang, and T. Liu, "Combining deep neural networks for protein secondary structure prediction," *IEEE Access*, vol. 8, pp. 84362–84370, 2020, doi: 10.1109/ACCESS.2020.2992084.

[20] G. Xu, Q. Wang, and J. Ma, "OPUS-TASS: A protein backbone torsion angles and secondary structure predictor based on ensemble neural networks," *Bioinformatics*, vol. 36, no. 20, pp. 5021–2026, 2020, doi: 10.1093/bioinformatics/btaa629.

[21] M. R. Uddin, S. Mahbub, M. S. Rahman, and M. S. Bayzid, "SAINT: Self-attention augmented inception-inside-inception network improves protein secondary structure prediction," *Bioinformatics*, vol. 36, no. 17, pp. 4599–4608, Nov. 2020, doi: 10.1093/bioinformatics/btaa531.

[22] M. J. Thompson and R. A. Goldstein, "Predicting solvent accessibility: Higher accuracy using bayesian statistics and optimized residue substitution classes," *Proteins Struct. Funct. Bioinforma.*, vol. 25, no. 1, pp. 38–47, 1996.

[23] X. Li and X.-M. Pan, "New method for accurate prediction of solvent accessibility from protein sequence," *Proteins Struct. Funct. Bioinforma.*, vol. 42, no. 1, pp. 1–5, 2001.

[24] H. Naderi-Manesh, M. Sadeghi, S. Arab, and A. A. M. Movahedi, "Prediction of protein surface accessibility with information theory," *Proteins Struct. Funct. Bioinforma.*, vol. 42, no. 4, pp. 452–459, 2001.

[25] S. Ahmad and M. M. Gromiha, "NETASA: Neural network based prediction of solvent accessibility," *Bioinformatics*, vol. 18, no. 6, pp. 819–824, Jun. 2002, doi: 10.1093/bioinformatics/18.6.819.

[26] Z. Yuan, K. Burrage, and J. S. Mattick, "Prediction of protein solvent accessibility using support vector machines," *Proteins Struct. Funct. Bioinforma.*, vol. 48, no. 3, pp. 566–570, 2002, doi: doi: 10.1002/prot.10176.

[27] S. Ahmad and M. M. Gromiha, "Design and training of a neural network for predicting the solvent accessibility of proteins," *J. Comput. Chem.*, vol. 24, no. 11, pp. 1313–1320, 2003, doi: 10.1002/jcc.10298.

[28] S. Ahmad, M. M. Gromiha, and A. Sarai, "Real value prediction of solvent accessibility from amino acid sequence," *Proteins Struct. Funct. Bioinforma.*, vol. 50, no. 4, pp. 629–635, 2003, doi: 10.1002/prot.10328.

[29] R. Adamczak, A. Porollo, and J. Meller, "Accurate prediction of solvent accessibility using neural networks–based regression," *Proteins Struct. Funct. Bioinforma.*, vol. 56, no. 4, pp. 753–767, 2004, doi: 10.1002/prot.20176.

[30] H. Kim and H. Park, "Prediction of protein relative solvent accessibility with support vector machines and long-range interaction 3D local descriptor," *Proteins Struct. Funct. Bioinforma.*, vol. 54, no. 3, pp. 557–562, 2004, doi: 10.1002/prot.10602.

[31] M. N. Nguyen and J. C. Rajapakse, "Prediction of protein relative solvent accessibility with a two-stage SVM approach," *Proteins Struct. Funct. Bioinforma.*, vol. 59, no. 1, pp. 30–37, 2005, doi: 10.1002/prot.20404.

[32] J. Sim, S.-Y. Kim, and J. Lee, "Prediction of protein solvent accessibility using fuzzy k-nearest neighbor method," *Bioinformatics*, vol. 21, no. 12, pp. 2844–2849, Jun. 2005, doi: 10.1093/bioinformatics/bti423.

[33] E. Faraggi, B. Xue, and Y. Zhou, "Improving the prediction accuracy of residue solvent accessibility and real-value backbone torsion angles of proteins by guided-learning through a two-layer neural network," *Proteins Struct. Funct. Bioinforma.*, vol. 74, no. 4, pp. 847–856, 2009, doi: 10.1002/prot.22193.

[34] K. Joo, S. J. Lee, and J. Lee, "SANN: Solvent accessibility prediction of proteins by nearest neighbor method," *Proteins Struct. Funct. Bioinforma.*, vol. 80, no. 7, pp. 1791–1797, 2012, doi: 10.1002/prot.24074.

[35] L. Deng, C. Fan, and Z. Zeng, "A sparse autoencoder-based deep neural network for protein solvent accessibility and contact number prediction," *BMC Bioinf.*, vol. 18, no. 16, Dec. 2017, Art. no. 569, doi: 10.1186/s12859-017-1971-7.

[36] B. Zhang, L. Li, and Q. Lü, "Protein solvent-accessibility prediction by a stacked deep bidirectional recurrent neural network," *Biomolecules*, vol. 8, no. 2, Jun. 2018, Art. no. 2, doi: 10.3390/biom8020033.

[37] M. Kaleel, M. Torrisi, C. Mooney, and G. Pollastri, "PaleAle 5.0: Prediction of protein relative solvent accessibility by deep learning," *Amino Acids*, vol. 51, no. 9, pp. 1289–1296, Sep. 2019, doi: 10.1007/s00726-019-02767-6.

[38] R. Kuang, C. S. Leslie, and A.-S. Yang, "Protein backbone angle prediction with machine learning approaches," *Bioinformatics*, vol. 20, no. 10, pp. 1612–1621, Jul. 2004, doi: 10.1093/bioinformatics/bth136.

[39] O. Keskin, D. Yuret, A. Gursoy, M. Turkay, and B. Erman, "Relationships between amino acid sequence and backbone torsion angle preferences," *Proteins Struct. Funct. Bioinforma.*, vol. 55, no. 4, pp. 992–998, 2004, doi: 10.1002/prot.20100.

[40] S. Wu and Y. Zhang, "ANGLOR: A composite machine-learning algorithm for protein backbone torsion angle prediction," *Plos One*, vol. 3, no. 10, Oct. 2008, Art. no. e3400, doi: 10.1371/journal.pone.0003400.

[41] B. Xue, O. Dor, E. Faraggi, and Y. Zhou, "Real-value prediction of backbone torsion angles," *Proteins Struct. Funct. Bioinforma.*, vol. 72, no. 1, pp. 427–433, 2008, doi: 10.1002/prot.21940.

[42] M.-S. Cheung, M. L. Maguire, T. J. Stevens, and R. W. Broadhurst, "DANGLE: A bayesian inferential method for predicting protein backbone dihedral angles and secondary structure," *J. Magn. Reson.*, vol. 202, no. 2, pp. 223–233, Feb. 2010, doi: 10.1016/j.jmr.2009.11.008.

[43] J. Lyons *et al.*, "Predicting backbone cα angles and dihedrals from protein sequences by stacked sparse auto-encoder deep neural network," *J. Comput. Chem.*, vol. 35, no. 28, pp. 2040–2046, 2014, doi: 10.1002/jcc.23718.

[44] R. Heffernan *et al.*, "Improving prediction of secondary structure, local backbone angles and solvent accessible surface area of proteins by iterative deep learning," *Sci. Rep.*, vol. 5, no. 1, Jun. 2015, Art. no. 1, doi: 10.1038/srep11476.

[45] H. Li, J. Hou, B. Adhikari, Q. Lyu, and J. Cheng, "Deep learning methods for protein torsion angle prediction," *BMC Bioinf.*, vol. 18, no. 1, Sep. 2017, Art. no. 417, doi: 10.1186/s12859-017-1834-2.

[46] J. Gao, Y. Yang, and Y. Zhou, "Grid-based prediction of torsion angle probabilities of protein backbone and its application to discrimination of protein intrinsic disorder regions and selection of model structures," *BMC Bioinf.*, vol. 19, no. 1, Feb. 2018, Art. no. 29, doi: 10.1186/s12859-018-2031-7.

[47] C. Fang, Y. Shang, and D. Xu, "Prediction of protein backbone torsion angles using deep residual inception neural networks," *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 16, no. 3, pp. 1020–1028, May 2019, doi: 10.1109/TCBB.2018.2814586.

[48] Y. Gao, S. Wang, M. Deng, and J. Xu, "RaptorX-Angle: Real-value prediction of protein backbone dihedral angles through a hybrid method of clustering and deep learning," *BMC Bioinf.*, vol. 19, no. 4, May 2018, Art. no. 100, doi: 10.1186/s12859-018-2065-x.
[49] F. Mataeimoghadam *et al.*, "Enhancing protein backbone angle prediction by using simpler models of deep neural networks," *Sci. Rep.*, vol. 10, no. 1, Nov. 2020, Art. no. 1, doi: 10.1038/s41598-020-76317-6.
[50] G. Xu, Q. Wang, and J. Ma, "OPUS-Refine: A fast sampling-based framework for refining protein backbone torsion angles and global conformation," *J. Chem. Theory Comput.*, vol. 16, no. 2, pp. 1359–1366, Feb. 2020, doi: 10.1021/acs.jctc.9b01054.
[51] Y. Qi, M. Oja, J. Weston, and W. S. Noble, "A unified multitask architecture for predicting local protein properties," *PLoS One*, vol. 7, no. 3, Mar. 2012, Art. no. e32235, doi: 10.1371/journal.pone.0032235.
[52] J. Jumper *et al.*, "Highly accurate protein structure prediction with alphafold," *Nature*, vol. 596, no. 7873, pp. 583–589, Aug. 2021, doi: 10.1038/s41586-021-03819-2.
[53] Y. Görmez, M. Sabzekar, and Z. Aydın, "IGPRED: Combination of convolutional neural and graph convolutional networks for protein secondary structure prediction," *Proteins Struct. Funct. Bioinforma.*, vol. 89, no. 10, pp. 1277–1288, 2021, doi: 10.1002/prot.26149.
[54] "Keras deep learning on graphs," 2020. [Online]. Available: https://vermamachinelearning.github.io/keras-deep-graph-learning/
[55] D. R. Jones, "A taxonomy of global optimization methods based on response surfaces," Art. no. 39.
[56] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei, and S.-H. Deng, "Hyperparameter optimization for machine learning models based on Bayesian optimization," *J. Electron. Sci. Technol.*, vol. 17, no. 1, pp. 26–40, Mar. 2019, doi: 10.11989/JEST.1674-862X.80904120.

Yasin Görmez received the graduate degree from Computer Engineering Department, Meliksah University, and the MSc degrees with high honor from the Electrical and Computer Engineering Department, Abdullah Gul University, in 2015 and 2017, respectively. He is currently working toward the PhD degree with the Electrical and Computer Engineering Department, Abdullah Gul University. He is a research assistant in management information sytems with Cumhuriyet University, Sivas, Turkey.

Zafer Aydin received the BSc and MSc degrees with high honor from the Electrical and Electronics Engineering Department, Bilkent University, in 1999 and 2001, respectively, and the PhD degree from the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta GA, in 2008. He then enrolled in the PhD program of the same department and worked as a teaching assistant for one year. Starting from 2002, he worked as a graduate research assistant with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta GA. As a result of maintaining an interest in bioinformatics research, he worked as a post-doctoral fellow for three years in Noble Research Lab, which is part of the Genome Sciences Department, University of Washington, Seattle, WA. From September 2011 to Februray 2014, he worked as an assistant professor with the Electrical and Electronics Engineering Department, Bahcesehir University, Istanbul, Turkey. He continued his career as an assistant professor with Computer Enginering Department, Abdullah Gul University, Kayseri, Turkey until receiving his tenure in March 2021. Currently, he works as an associate professor with the same department. His research interests include machine learning, bioinformatics, computational biology, health informatics, and biomedical engineering.