

Ensemble model for accuracy prediction of protein secondary structure

Srushti C. Shivaprasad¹, Prathibhavani P. Maruthi¹, Teja Shree Venkatesh¹, Venugopal K. Rajuk²

¹Department of Computer Science and Engineering, University Visvesvaraya College of Engineering (UVCE), Karnataka, India

²IEEE Fellow, Former Vice Chancellor, Bangalore University, Bengaluru, India

Article Info

Article history:

Received Feb 17, 2023

Revised Jul 24, 2023

Accepted Sep 26, 2023

Keywords:

Convolutional neural networks

CullPDB6133

Protein secondary structure prediction

Q8 accuracy

SVM classifier

ABSTRACT

Predicting a protein's secondary structure is crucial for understanding the working of proteins. Despite advancements over the years, the top predictors have achieved only 80% Q8 accuracy when sequence profile information is the sole input. An ensemble approach is proposed using convolutional neural network (CNN) and a classifier known as support vector machine (SVM) on both the partial and the whole CullPDB datasets. The protein secondary structure (PSS) has a complex hierarchical structure, as well as the ability to take into account the reliance between neighbouring labels. A detailed experiment yielding high levels of Q8 accuracy with scores of 97.91%, 85.13%, and 78.02% using 20%, 80%, and 100% respectively of the protein residues on the new predicted dataset CullPDB6133 which is better than the accuracies predicted by similar models. The proposed methodology highlights the use of CNN as a general framework, for efficiently predicting eight-state (Q8) accuracy of secondary protein structures with a low time and space complexity.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Srushti C. Shivaprasad

Department of Computer Science and Engineering, University Visvesvaraya College of Engineering (UVCE)

K. R. Circle, Bengaluru, Karnataka, India

Email: srushtichan@gmail.com

1. INTRODUCTION

Proteins are the driving force behind all biological processes since they represent the genetic code's immediate expression. To ensure the normal operation of every living being, proteins serve as catalysts for reactions in physiological processes like digestion, deoxyribonucleic acid (DNA) replication and cellular metabolism. Proteins also have a role in the development of complex structures like bone and collagen and the upkeep of systems like the immune system [1]. In a variety of biological issues and disease detection techniques connected to identifying drug resistance, proteins perform a variety of roles. It is crucial to comprehend its role in the development of efficient diagnostic techniques [2], drug target identification/discovery [3] and therapeutic interventions [4].

Proteins made of amino acid sequences are bound together by peptide bonds. All proteins that are known are composed of about 20 distinct amino acids. Each amino acid (AA) is composed of an atom carbon-centered backbone ($C\alpha$), a COOH functional group that represents the carboxyl group, a NH_2 group that represents an amino group and a side chain or R-group. A carbon atom (C) bonds to an oxygen atom (O) through a double bond ($C=O$) and to a hydroxyl group (OH) via a single bond in the COOH functional group. The nitrogen atom in the centre is bonded to two hydrogen atoms to form the amino group. Because each AA has a unique R-group, they all exhibit different chemical properties. Proteins perform a variety of tasks attributed to the variability in the R group among the 20 amino acids. The simplest R group, consisting of just

one hydrogen atom, is found in glycine. Determining the folded, or tertiary, structure of the thousands of newly found protein sequences each year is essential to understanding their biological significance [5].

Protein secondary structure (PSS): The primary *motivation* behind the protein secondary structure prediction (PSSP) is that most protein folding prediction methods rely on information about the PSS. This is because knowledge of the PSS is a crucial prerequisite for accurately forecasting the protein’s native or 3D (3-Dimensional) structure [6]. The accurate PSS identification helps comprehend the intricate dependencies between protein sequences and tertiary structures. Thus, PSS is relied upon for its application in protein structure modeling, protein function prediction, protein folding studies, annotating a protein’s functions, and changing proteins (protein engineering) [7].

The PSS encompasses the α -helix, the β -sheet (also known as a pleated sheet) which was first introduced in [8] and random coil. Helix (represented as H), strand (represented as E), and coil (represented as C) are the three states (Q3) that characterize PSS. The Q3 was then evolved into 8 states represented as Q8. The 8 states are α -helix (represented as H), β -strand (represented as E), isolated β -bridge (represented as B), 3_{10} -helix (represented as G), π -helix (represented as I), bend (represented as S), turn (represented as T), and Others (represented as L), to provide a more precise description of the PSS [9]. The more difficult Q8 prediction can provide more exact and highly detailed information on the proteins’ structural characteristics. Figure 1 [10] depicts a typical sequence and the corresponding secondary protein structure in 2D visualization.

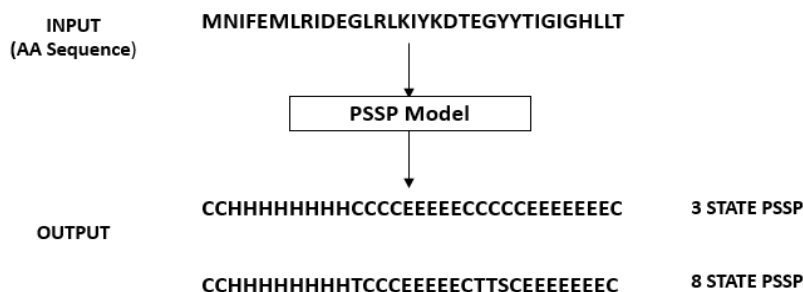


Figure 1. A typical example of structural protein sequences

The significance of PSSP in accurately determining the tertiary structure of proteins, coupled with the high costs and high time complexity of experimental methods, there is a growing demand for computational approaches to handle these predictions efficiently. Q3 and Q8 accuracies, which respectively assess the proportion (%) of residues for three-state and eight-state secondary structures (SS) are used to evaluate PSSP. The proposed study makes substantial contributions. The fundamental framework is an ensemble architecture of deep convolutional neural network (DCNN) with support vector machine (SVM) and is utilized to ascertain the Q8 accuracy of the PSS. Q8 accuracy obtained is higher than the models using SVM with only sequence features. The proposed technique with the custom CullPDB6133 dataset has an improved accuracy of 3% over earlier work.

The paper is organized into six sections. The scope and context of the study are highlighted in section 1. The literature survey is presented in section 2. In section 3, a comprehensive explanation of the materials and methods is detailed. Section 4 provides an in-depth description of the proposed model and algorithm. In section 5, experimental setup, data, outcomes and comparative analysis are presented. The summary of the research article is outlined in section 6.

2. RELATED WORK

PSSP is now feasible with advanced deep learning (DL) algorithms that outperform machine learning techniques. Due to this, there is clear productivity in accurate prediction. The algorithms used are based on convolutional neural network (CNN) [11], and recurrent neural networks (RNN) [12]. DeepCNF [13] makes use of deep hierarchical design to find the interdependency between neighboring secondary structure labels. To efficiently process local and non-local interactions between amino acids, MULFOLD-SS [14] uses hierarchical deep inception blocks. The ensemble model of generative stochastic network (GSN) architecture and CNN for PSSP has been proposed in the study [15]. To combat the loss of detail in the data brought on by CNN’s convolution and pooling stages, MUST-CNN [16] has made use of Multilayer-Shift-and-Stitch for PSSP. The method based on SVM RBF kernel was introduced in [17]. The inputs for this study are sequences of AA and position specific scoring matrix generally known as PSSM. The prediction of PSS using ensemble

techniques, such as DL models based on AdaBoost and Bagging is described in [18]. A two-stage hybrid classifier known as ROSE predicts PSS in two stages. It uses a 1D bidirectional recurrent neural network (BRNN) and SVM in the first stage and last stage respectively [19]. A novel technique is introduced for forecasting one-dimensional structural characteristics of proteins. It relies on an ensemble of various neural network models, including long short term memory (LSTM)-BRNN, residual network (ResNet), and fully-connected neural network (FC-NN), with input from predicted contact maps provided by SPOT-contact [20]. There are many servers that are used in predicting the PSS such as Porter [21], JPred4 [22]. Table 1 provides an overview of the most recent developments in predicting PSS.

Table 1. Table summarizing relevant research

Author & year of publication	Concept and model	Performance (Q8 Accuracy)	Advantages	Disadvantages
Yuan <i>et al.</i> [23] 2022	DLBLS_SS: PSSP using DL and broad learning system	73.35% on CB6133 dataset	Strong feature extraction potential with the ability to predict using both local and global optimal features, improving the identification of secondary structure (SS).	The model does not explore complex sequence-structure relationship.
Yang <i>et al.</i> [24] 2022	ShuffleNet_SS: The network's capacity to learn unusual classes is improved by using label distribution aware margin loss and modified 1D batch normalisation	ShuffleNet_SS (LDAM): On CB513 dataset: 71.87%. Loss used is label distribution aware margin (LDAM)	The domain of 8-state deep Secondary Structure Prediction (SSP) is incorporating the loss for imbalanced datasets.	Imbalanced classification methods to raise the accuracy of the rare classes is not considered.
Yuan <i>et al.</i> [25] 2023	Bidirectional temporal convolutional network (BTCN), BLSTM, with proposed network Multi-scale bidirectional temporal convolutional network (MSBTCN)	73.89% on CullPDB dataset	Strong stability and feature extraction capabilities, which successfully address the drawbacks of inadequate capture of long-range (distant) dependencies in sequences	When interacting with additional real sequence data, BTCN might overlook certain information whereas MSBTCN might add irrelevant information.
Srushti <i>et al.</i> Proposed system	Obtain well-curated dataset "CullPDB6133Filtered" Deep CNN to generate feature maps from the processed dataset and given as input to SVM classifier.	(i) 97.91% of Q8 accuracy on 20% of CullPDB6133 filtered dataset (ii) 85.13% on 80% of CullPDB6133 filtered dataset (iii) 78.02% on CullPDB6133 dataset	The proposed technique is the first to employ the filtered CullPDB6133 dataset and it has improved accuracy of 3% over previous research on the same dataset.	

The objective and goal of this research are to find the Q8 Accuracy of PSS when provided with an unprocessed protein dataset with low computational resources. Thus, the focus is on preparing a clean dataset or filtered dataset from the CullPDB6133 dataset that contains duplicate, to train a deep neural network for the classification using a blend of the CNN model, providing a model with a classification accuracy of more than 80% for PSS prediction using 20% of the CullPDB6133 filtered dataset.

3. METHOD

This section outlines the dataset and network architectures in the proposed system. The protein dataset contains the essential information required for training and evaluation of ensemble model. The network architectures shed light on the intricate designs that underpin the core functionality of the proposed system.

3.1. Dataset description

In order to determine a protein's secondary structure, one must have access to protein sequences, protein data, and protein databanks. Mass spectrometry, nuclear magnetic resonance (NMR) spectroscopy, exclusion chromatography is some of the techniques used to progressively and persistently uncover the structure of proteins. Newly identified proteins deposited in databanks like the protein data bank (PDB) [26] have been used as a default dataset by researchers for decades. When it comes to protein data, including 3D structure, the PDB is a de-facto standard.

To ensure consistency in benchmarking, the proposed system relied on the validated dataset from Princeton University hosted on the Princeton University official portal. The dataset utilized is the CullPDB6133, which has 6,133 proteins, with a total of 39,900 characteristic features and 700 amino acids along with 57 characteristics. The peptide chain length is indicated by the number 700, while 57 denotes the number of characteristics for each position of AA. Although it might vary with fewer or greater numbers of amino acids, the protein’s polypeptide chain typically consists of 200-300 amino acids.

In this dataset, a protein chain’s average length is 208 amino acids. Out of the total of 57 features, 22 pertain to the protein’s fundamental structure, and 9 to its secondary structure. Rather than relying on amino acid residues, protein profile residue was used. The original CullPDB6133 file uploaded by Princeton University contains duplicates. Thus, to fix the issue and provide a well-curated dataset, subsequently called “CullPDB6133Filtered”, the original dataset is transformed using principal component analysis (PCA) to reduce dimensionality.

3.2. Network architecture

The neural network layers which are utilized in the model are the CNN, MaxPool layer, Flatten Layer, Dense Layer, Activation Function, Dropout Layer, Learning rate as well as SVM. The description of each is mentioned below. The next section describes the model’s overview.

3.2.1. The convolutional layer

CNN’s components are often the same size as the final product of the computations. The hidden units perform a dot-product operation on the data, storing information about the connections between the data points. Each piece of information generated by a hidden unit is recorded on a feature map, and there will be as many feature maps as hidden units. Next, the existing feature maps undergo the pooling stage, which retrieves detailed information from the feature maps. The function *Conv2D(filters, kernel size, stride, padding, activation)* performs convolution operation.

Non-linearity is a crucial concept in CNN. The max-pool layer’s non-linear function is used in this model to guarantee that the greatest possible number of non-overlapping regions was chosen. The max-pooling procedure *MaxPooling2D(poolsize, strides, padding)* can obtain the best value of any region of interest (ROI) in a $m*n$ dimensional image, as explained by a function represented as k and stride size as illustrated in (1).

$$\frac{m-k}{sr+1} * \frac{(n-k)}{sr+1} \tag{1}$$

The input image is further down sampled by *max-pooling()* layer as shown in Figure 2. This is envisaged in [27].

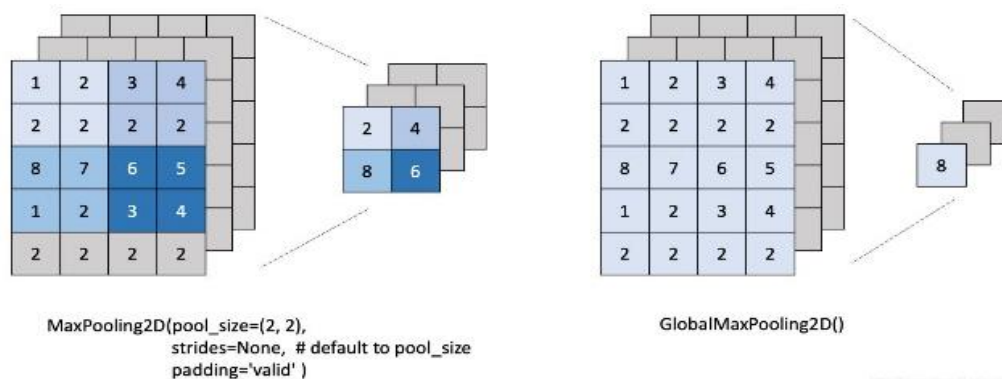


Figure 2. Max-pooling operation using a 2x2 filter

3.2.2. Dropout layer, learning rate

The dropout layer was first proposed in the study [28]. To avoid complex co-adaptations on the training data and improve generalization, the dropout method randomly excludes half of the feature detectors through every instance of training. The function *Dropout(dropout-rate)* prevents overfitting in neural networks. The use of a *learning rate* was initially suggested in [29]. Since the learning rate is a component of the overall weight update rule and is computed at the end of the backpropagation layer, it determines the extent of the feedback required. The value is between 0 and 1.

3.2.3. Activation function

The model becomes non-linear when an *activation function* which is a mathematical function, is applied to a neuron's output. Since they aid in the modelling of complicated interactions and provide neural networks the ability to learn and make non-linear judgments, activation functions are a crucial part of neural networks. The example would be to mention functions including Sigmoid, rectified linear unit (ReLU), Leaky ReLU, Tanh (Hyperbolic Tangent), Softmax [30] as activation functions in the model.

3.2.4. Fully connected layer

The layer referred to as the *FC layer* is a Fully Connected layer in which each individual neuron or node is coupled to every neuron in the preceding layer. They identify images using the convolved features, just as conventional NN. Loss functions are calculated and then propagated through backpropagation. The suggested model has five levels, all of which are coupled to one another and result in an output layer. *SoftMax* is the activation function used by each of the four nested layers. Each independent label seeks to predict the probability within 0 and 1 generated by one of the activation functions known as the sigmoid activation function. This is defined in (2).

$$\text{Sign}(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{1+e^x} \quad (2)$$

Where $x = \text{input vector}$

The deep CNN is an extension of basic CNN architecture with the addition of dilated layers of dropout regularization. A convolution, pooling, and FC layers make up the three main layers of a CNN. The image is then fed to the convolution layer, where feature extraction neurons are located. Convolution is a matrix of size $f * f$ with the input images results in an activation map; a constant called stride length which is used to move the filter across the images. When applied to an image of size $m * n$, convolution with a filter represented as fr , padding pa and stride represented as sr produces the corresponding result.

$$(\text{OUTimg}) = \frac{(m-fr+2pa)}{sr+1} * \frac{(n-f+2pa)}{(sr+1)} \quad (3)$$

Also, the training images have a huge impact on the filter's accuracy. Nonlinearity was introduced to the model via the standard *SoftMax*(xa) in (4) as:

$$\sigma_{(xa)} = \frac{e^{xa}}{\sum_{b=1}^n e^{xa}} \quad (4)$$

for $a = 1, \dots, n$; $x = (x_1, x_2, \dots, x_n) \in R^n$

where n = the sum of all the elements of the input vector x and x_i depicts each element of the vector space.

3.2.5. Flatten layer

Although the spatial structure of the CNN hides the nature of the underlying matrix multiplication, one may *flatten* this spatial structure to conduct the multiplication and then "reshape" it again using the elements' recognized spatial positions [31]. A flatten layer is a layer that converts multi-dimensional data into a one-dimensional array or vector. This is frequently done in order to get the data ready for more processing, such as feeding it into fully connected layers or other neural network topologies that need one-dimensional input.

3.2.6. Support vector machine

SVM are a type of classifier used for both binary classification and multiclass classification tasks. It employs separating hyperplanes (decision borders) to distinguish between classes. The SVM aims to maximise the separation between the classes to make the decision boundary as remote as possible. Some of the applications including image recognition, bioinformatics, computer vision, natural language processing (NLP), text and document analysis are shown to benefit from the use of SVM.

4. PROPOSED METHOD

In order to address spatial issues, CNN model has been developed. In order to retrieve useful characteristics from the data, CNN is employed to perform feature extraction and enrichment, thus transforming the data into a higher dimension. The dependability between AA residues across a long distance in AA sequences were captured with the help of the CNN. To prevent overfitting, an approach proposed in [32] is used. Later, SVM was used because of its capacity to classify high-dimensional data.

4.1. Overview of the model

An ensemble strategy enhances the performance and accuracy of the deep learning models. An extension of the CNN model is DCNN and with an additional SVM classifier, an ensemble model known as DCNN_SVM is proposed in this work. The Figure 3 gives the architecture of the DCNN which consists of convolution layer, fully connected layer, flatten layer, dropout layer, learning rate, activation function, max pool layer and these have been described in the previous section.

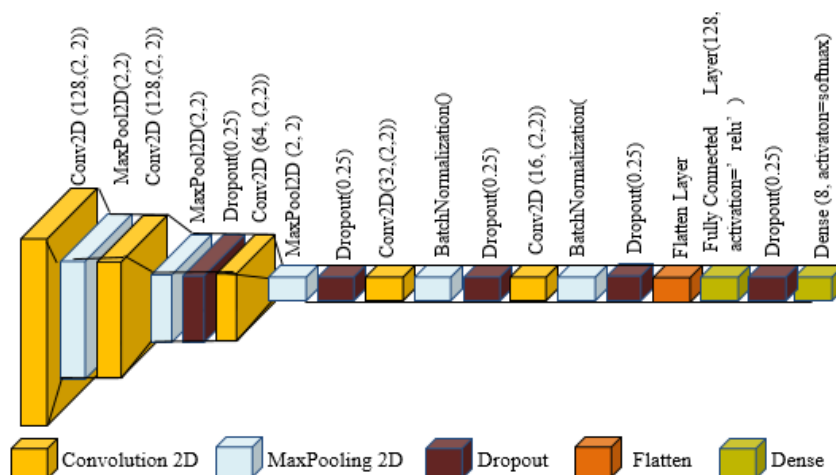


Figure 3. DCNN architecture

4.1.1. Data preprocessing

The initial stage of PSSP is to transform a set of proteins into feature-based representations from alphabetic strings of capital letters, which correspond to the twenty naturally occurring amino acids. In this study, the feature-based representation used for protein sequence is one hot encoding, where each AA is represented by a binary vector indicating the presence or absence of it. Principal component analysis (PCA) is one of the extensively used techniques for reducing the dimensionality of a dataset. PCA uses the covariance matrix, along with its eigenvectors and eigenvalues, to determine the major components in the data. These primary components are uncorrelated eigenvectors that each contribute to a portion of the data's variance.

Using sums of squares and cross products, the eigenvalues and eigenvectors of a square symmetric matrix are solved, and this is the mathematical method utilized in PCA. The direction of the first principal component is associated with the eigenvector connected to the biggest eigenvalue. The second principal component's direction is determined by the eigenvector connected to the second-largest eigenvalue. The maximum count of eigenvectors is equal to the number of rows (or columns) of this matrix, and the sum of the eigenvalues equals the trace of the square matrix. The function for the same is described below [33].

Function: PCA (protein dataset)

Begin

Step (i) Get the protein dataset

Step (ii) Subtract the mean

Step (iii) Compute the data sets' covariance matrix

Step (iv) Compute the covariance matrix's eigenvectors and eigenvalues

Step (v) Choose Components and form Feature Vector

Step (vi) Derive new dataset

Step (vii) Get old data

Return transformed protein dataset

End

4.1.2. Feature extraction

Because irrelevant features frequently have an impact on how well the machine learning (ML) classifier performs classification, the extraction of the significant features is a crucial step. The process which performs data analysis to obtain informative features from raw data is known as feature extraction. The model's training time is shortened and classification accuracy is increased with the right feature extraction technique.

4.2. Algorithm for the proposed model

The proposed algorithm for Q8 Accuracy prediction of PSS from an ensemble model of DCNN and support vector machine (DCNN_SVM) is given below. The proposed algorithm consists of 2 algorithms. They are *Preprocessing* and *Build_DCNN_SVM_Model*. Algorithm *Preprocessing* performs preprocessing operation on the protein dataset. It has two procedures: procedure *DataPreprocessing* and procedure *Find*. In the preprocessing operation defined in procedure *DataProcessing*, the dataset is handled for missing values and categorical data for every instance in the dataset. The output is the preprocessed protein dataset which is an input to the procedure *Find*. The *Find* procedure selects the appropriate dataset (CullPDB6133 or CullPDB6133Filtered). In order to reduce dimensionality and duplicates in the dataset, PCA is applied to the original protein dataset to obtain transformed dataset known as CullPDB6133Filtered. The algorithm 1 for the same is mentioned below.

Algorithm 1: Preprocessing

```

Begin
  Input: protein dataset
  Output: protein dataset

  Procedure DataPreprocessing (protein dataset)
  Begin
    for every instance in the dataset do
      Handle missing values
      Handle categorical data
    End for
    Return protein dataset
  End //End procedure DataPreprocessing

  Procedure Find(protein dataset)
  Begin
    Input: Protein dataset
    Output: Protein dataset // preprocessed or transformed protein dataset

    Switch (protein dataset)
    Begin
      Case 1: Return protein dataset
      break
      Case 2: //Reduce dimensionality by applying PCA to original dataset
      PCA(protein dataset)
      Return transformed protein dataset
    End //End switch
  End //End procedure Find
End //End Algorithm Preprocessing

```

The algorithm *Build_DCNN_SVM_model* builds the model by taking the protein dataset generated from the previous algorithm as input. It consists of two sub-procedures *DCNN* and *Forward_SVM*. An ensemble architecture of DCNN with SVM is used to ascertain the Q8 accuracy of SSP. The introduction of drop-out and batch normalization layers with One-Hot Encoding yields improved results in terms of Q8 accuracy and the model under consideration achieves higher Q8 accuracy than using only SVM [34] with sequence features.

The process for training entails using a few of the initial settings of 125,005 trainable parameters. Among the main problems with this strategy is the padding it requires for shorter sequences, nevertheless affecting the loss on the complete sequences. The outputs from the padding region are unique in shape for each case. The algorithm for the proposed model is highlighted for Q8 accuracy as illustrated in the Algorithm 2. The DCNN network consists of two 128-filter length convolution layers and one each of 64, 32, 16-filter lengths convolution layers for classification. All of the layers were generated using a 2x2 kernel. In this design, the 2x2 max-pooling layer is taken into account and the stride length is set to 2. The last pooling layer's 2D output will generate feature maps and is flattened in a flattening layer, turning it into a 1D layer. Padding is the same for all convolution layers. To divide the data into two groups, a SoftMax activation function is applied to a fully linked layer of size 1024. Adam is the optimization function that is employed. The value of learning rate (lr) is set to 0.0001.

The sub-procedure will return detailed information from the generated feature maps which will then be forwarded to the sub-procedure *Forward_SVM*. In this sub-procedure, the output from the sub-procedure DCNN will then be used to compute the Q8 accuracy of PSSP. The Q8 accuracy is computed when the condition $\text{argmax}()$ prediction < 0.5 is met. The window size is selected to be above 11 as the typical length of an α -helix is about 11 residues and the β -strand is approximately six residues. Various even sizes of 11 through 23 were examined, with 17 offering the best accuracy. The model summary is presented in Figure 4.

Algorithm 2: Build_DCNN_SVM_model

```

Begin
  Input: protein dataset
  Output: Q8 Accuracy

  Procedure BuildModel (protein dataset)
  Begin
    Sub-procedure DCNN( protein dataset)
    Begin
      Input: protein dataset
      Output: Generated feature maps
      For every input in dataset do
        Add Conv2D layer, Maxpooling Layer
        i=0
        while i<3 do
          Step (i)      Add Conv2D Layer, Maxpooling Layer
          Step (ii)     Generate Feature Maps
          Step (iii)    Activation (Relu)
          Step (iv)     Padding (same)
          Step (v)      Dropout (25%)
        End while
        Add Conv2D Layer, BatchNormalization
        Add Conv2D Layer, BatchNormalization
        Add Flatten Layer, Add Dense Layer
        Dropout (25%)
        Add Dense (1) and activation (softmax)
      End //End for
      Return predicted feature maps
    End //End sub-procedure DCNN

    //Forward to SVM model

    Sub-procedure Forward_SVM(predicted feature maps)
    Begin
      Input: protein dataset
      Output: Q8 Accuracy

      If argmax() prediction <0.5
        Compute Q8 Accuracy
      Else
        BuildModel ( protein dataset)
      End if
      Return Q8 Accuracy
    End // End subprocedure Forward_SVM
  End //End procedure BuildModel
End //End algorithm
  
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 700, 17, 128)	10880
activation (Activation)	(None, 700, 17, 128)	0
max_pooling2d (MaxPooling2D)	(None, 350, 8, 128)	0
conv2d_1 (Conv2D)	(None, 350, 8, 128)	65664
activation_1 (Activation)	(None, 350, 8, 128)	0
max_pooling2d_1 (MaxPooling2D)	(None, 175, 4, 128)	0
dropout (Dropout)	(None, 175, 4, 128)	0
conv2d_2 (Conv2D)	(None, 175, 4, 64)	32832
activation_2 (Activation)	(None, 175, 4, 64)	0
max_pooling2d_2 (MaxPooling2D)	(None, 87, 2, 64)	0
dropout_1 (Dropout)	(None, 87, 2, 64)	0
conv2d_3 (Conv2D)	(None, 87, 2, 32)	8224

Figure 4. Model summary

5. EXPERIMENTATION AND DISCUSSION

Since the available experimental approaches to address the PSSP problem are exceedingly expensive in terms of both money and time the attempt to do so was substantial. PSSP seeks to provide as precise structure predictions as possible, given only the blueprints. The metric Q8 accuracy measures how well a protein's secondary structure was predicted using an eight-state model. The principal aim of this study is to implement and design an ensemble of deep CNN and SVM model capable of predicting the PSS based on its foundational structure. CullPDB6133 and a sliced version, CullPDB6133Filtered is utilized to as the dataset for this study. CullPDBFiltered dataset is obtained by applying PCA algorithm to the original CullPDB6133 dataset. In order to facilitate training and evaluation, the dataset is partitioned as per Table 2.

Table 2. Distribution of the CullPDB6133 dataset

CullPDB6133 dataset	80% of CullPDB6133Filtered	20 % of CullPDB6133Filtered
Training: 80% vs Val: 20%	Training: 80% vs Val: 20%	Training: 80% vs Val: 20%

The experiment was performed using two approaches. First, the CullPDB6133 dataset was used completely and second, only partial slices of them are used. The model was subsequently trained in phases, viz: **Case 1:** 80% of the sequences are sampled. For the training, 80% of the sampled sequences is used, whereas 20% is used for testing.

Case 2: 20% of the sequences are sampled. For the training, 80% of the sampled sequences is used, and the remaining (20%), reserved for testing.

To ensure that the suggested model can correctly interpret protein structures including helices, sheets, and loops, it is trained on proteins.

5.1. Experimental setup

The following describes the experimental setting for the aforementioned findings.

Hardware: The model is trained on a Windows 10 computer featuring an Intel(R) Pentium(R) Core i7 8th Generation processor, operating at a clock frequency of 2.30GHz, and utilizes a dedicated graphics accelerator card, the GeForce GTX 1060.

Input features: A two-part input feature for a sequence is considered. They are position-specific scoring matrices (PSSM) and one-hot vectors for a sequence. One-hot vectors of length 21 (20 different types of amino acids, and an unknown AA) represent each AA in the protein sequence. The PSSM is a representation of the occurrence with which different types of amino acids appear at various locations in the protein sequence.

5.2. Model performance metric

The efficiency of the suggested model for predicting the Q8 accuracy of PSS was evaluated using the quantitative standard metrics stated in (5). A system's accuracy (Acc) can be thought of as the proportion of properly labelled instances relative to the sum of instances. The corresponding equation is mentioned.

$$Acc = \frac{TP+TN}{TP+FP+FN+TN} \quad (5)$$

5.3. Results and discussion

5.3.1 Model performance

The accuracy of the proposed model at various epochs is mentioned in Table 3. The train and validation accuracy when using 80% of the CullPDB6133Filtered dataset is 83.91% and 85.13% with the highest at the 20th epoch. Using the filtered dataset, the model is trained for 4 epochs with an early stoppage hyperparameter. Keeping the learning rate constant (0.00001), with a bias factor of 0.01 and momentum at 0.009, a training and validation accuracy of 98.62% and 97.91% is recorded as illustrated in Figure 5.

It can be noted that for the complete CullPDB6133 dataset, although the model is trained for 38 epochs using early stoppage, the highest accuracy was recorded at the 35th epoch, and the model improved no further even with the patience factor of 3, which was the main reason the accuracy vs epoch curve stopped at 35th epoch. This behavior is illustrated in Figures 6 and 7 respectively. Fifty-five training epochs are used to perfect the DCNN_SVM on the CullPDB6133 dataset (on CPU in approximately 14 hours). It can be inferred that the approach in the proposed system yielded an improvement over the previous research because of hyperparameter tuning. The learning rates and epochs alongside variations of batch normalization were carefully chosen.

Table 3. Model accuracy across epochs

CullPDB6133	#Epochs	Training Acc. (%)	Val Acc. (%)
	10	68.93	70.01
	20	73.67	76.29
	30	75.99	77.56
	35	77.90	78.02
	38	77.59	76.89
20% of CullPDB6133Filtered	#Epochs	Training Acc. (%)	Validation Acc. (%)
	4	98.62	97.91
	20	98.20	97.13
80% of CullPDB6133Filtered	#Epochs	Training Acc. (%)	Validation Acc. (%)
	4	68.45	70.23
	20	83.91	85.13

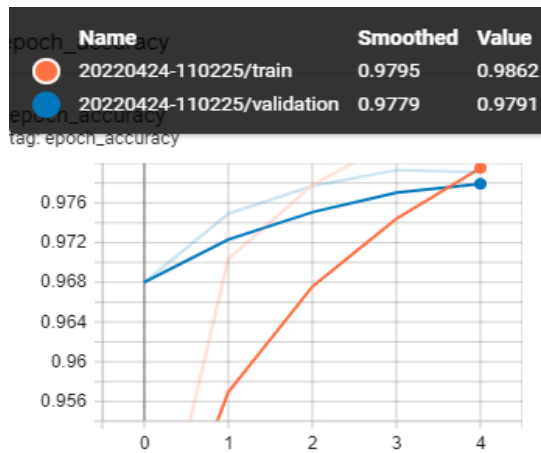


Figure 5. Accuracy vs Epoch of the DCNN_SVM (20% of CullPDB6133Filtered)

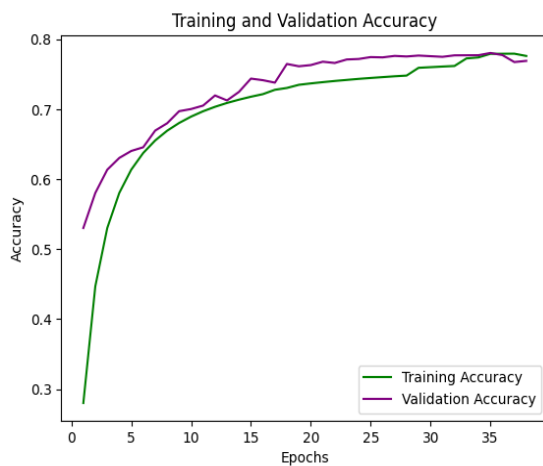


Figure 6. The DCNN_SVM model Q8 accuracy

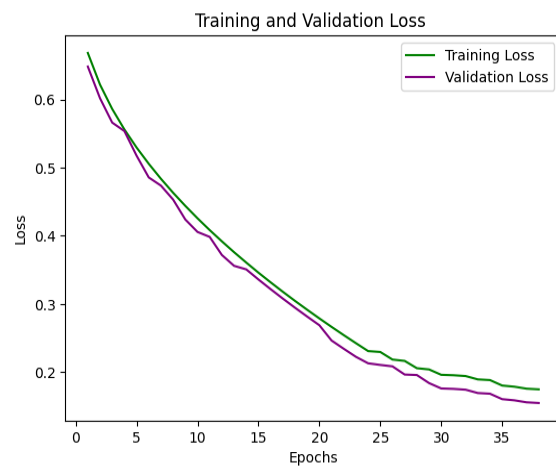


Figure 7. DCNN_SVM model loss

5.3.2. Performance analysis

The model’s performance analysis with other authors is presented in Table 4. The results of the proposed research are highlighted in bold. Chopra and Bender [35] employed cellular automaton for PSSP. It was provided with AA sequence as an input and the rules evolved via a genetic algorithm for updating states to compute Q3 accuracy. The accuracy obtained was 56.51% for the CB513 dataset. Because it tried to replicate the folding determinants prevalent in nature, accuracy was reduced. The approximate Q8 accuracy presented by Asgari and Mofrad [36] on PDB dataset is around 93%. The proposed model has better accuracy since it uses an ensemble of DCNN and SVM.

Table 4. Comparison with other authors

Year	Authors	Dataset/Methods	Accuracy
2007	Chopra <i>et. al</i> [35]	CB513	56.51%
2015	Asgari and Mofrad [36]	PDB	93%±0.06
2023 (20% of the dataset)	Proposed (DCNN_SVM)	CullPDB6133Filtered	97.91%
2023 (80% of the dataset)	Proposed (DCNN_SVM)	CullPDB6133Filtered	85.13%
2023 (100% of the dataset)	Proposed (DCNN_SVM)	CullPDB6133	78.02%

The performance comparison with other models is tabulated in Table 5. The proposed model DCNN_SVM is compared with DeepACLSTM [37], PS8-Net [38], OneHotEncoding with LSTM [39] which are trained with CullPDB6133 dataset. They have achieved Q8 accuracy of 74.2%, 76.89% and 77.8%. In DeepACLSTM, asymmetric convolutional neural networks (ACNNs) with BLSTM are combined and the advantage of the protein feature matrix's feature vector dimension is taken. The long-distance interdependencies between AAs are captured by BLSTM neural networks, while ACNNs extract the intricate local contexts of AA. The PS8-Net module operates with skip connections to collect global information during SSP by extracting long-term interdependencies from deeper layers and retrieving local contexts from previous levels. In Onehot encoding and LSTM-based method, the model uses the OneHotEncoding approach. The entire protein sequence is transformed using this technique into the input feature vector and it is then provided as input to the LSTM model for PSSP as DCNN_SVM uses feature maps generated from DCNN as input to SVM and then computes Q8 accuracy of PSSP.

Table 5. Proposed model performance comparison with other models on CullPDB6133

Sl. No	Model	Dataset	Accuracy (%)
1	DeepACLSTM [37]	CullPDB6133	74.2(Q8)
2	PS8-Net [38]	CullPDB6133	76.89%(Q8)
3	OneHotEncoding and LSTM [39]	CullPDB6133	77.8(Q8)
4	*DCNN_SVM [Proposed]	CullPDB6133Filtered	97.91(Q8)

The performance comparison of different models with the said model is represented in Figure 8. The average of the Q8 Accuracy of DeepACLSTM and OneHotEncoding and LSTM-based model across 25 epochs is plotted. The labels proposed_80% and proposed_20% refer to the DCNN_SVM model trained using 80% and 20% of the CullPDB6133Filtered dataset.

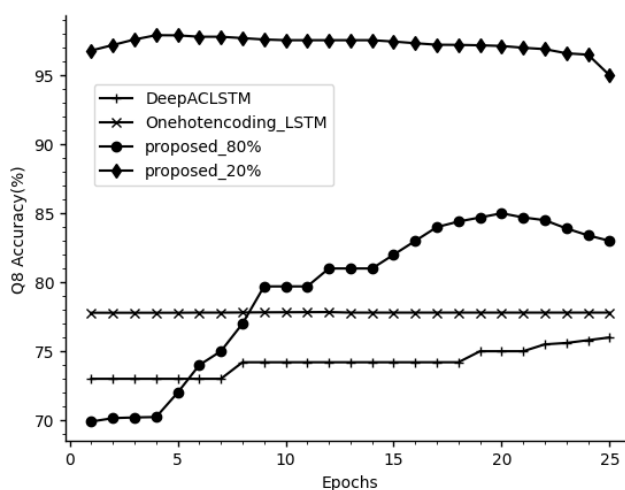


Figure 8. Performance comparison with other models on CullPDB6133

6. CONCLUSION

In the proposed work, an ensemble model deep CNN with SVM (DCNN_SVM) for predicting Q8 accuracy of PSS was proposed. To store detailed contextual information, two stacked layers of deep neural networks alongside an SVM classifier block was proposed. It was able to extract protein information using the

DCNN. The proposed model was able to attain an accuracy of 97.91% on the filtered dataset of 20% of the CullPDB6133Filtered dataset. According to the results of several testing, DCNN_SVM suggested method, was found to be generalizable, and suited for both sequence-labelling tasks and PSS activities.

Despite using lower computation and processing resources, accuracy comparable to certain cutting-edge techniques was reached. However, the work is limited as the performance of the proposed approach on other available large PSS datasets could not be investigated. The future enhancement will be to incorporate other benchmarked datasets and high-end computing systems to predict Q8 accuracy in PSS.





REFERENCES

- [1] A. Kessel and N. Ben-Tal, "Introduction to proteins," in *Introduction to proteins: structure, function, and motion*, 2nd Ed., CRC Press, 2018.
- [2] Z. Ding, N. Wang, N. Ji, and Z.-S. Chen, "Proteomics technologies for cancer liquid biopsies," *Molecular Cancer*, vol. 21, no. 1, p. 53, Feb. 2022, doi: 10.1186/s12943-022-01526-8.
- [3] T. J. Hedl *et al.*, "Proteomics approaches for biomarker and drug target discovery in ALS and FTD," *Frontiers in Neuroscience*, vol. 13, 2019, doi: 10.3389/fnins.2019.00548
- [4] R. Shelin and S. Meenakshi, "Rise of bacterial small proteins and peptides in therapeutic applications," *Protein Pept Lett*, vol. 30, no. 2, pp. 126–136, 2023, doi: 10.2174/0929866530666230118144723.
- [5] M. Zamani and S. C. Kremer, "Protein secondary structure prediction through a novel framework of secondary structure transition sites and new encoding schemes," in *IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, IEEE, 2017, pp. 1–7, doi: 10.1109/CIBCB.2016.7758118.
- [6] M. N. Nguyen, J. M. Zurada, and J. Rajapakse, "Toward better understanding of protein secondary structure: Extracting prediction rules," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 8, no. 3, pp. 858–864, 2010, doi: 10.1109/TCBB.2010.16.
- [7] J. Zhou, H. Wang, Z. Zhao, R. Xu, and Q. Lu, "CNNH_PSS: protein 8-class secondary structure prediction by convolutional neural network with highway," *BMC Bioinformatics*, vol. 19, no. 4, p. 60, May 2018, doi: 10.1186/s12859-018-2067-8.
- [8] L. Pauling, R. B. Corey, and H. R. Branson, "The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain," *Proceedings of the National Academy of Sciences*, vol. 37, no. 4, pp. 205–211, 1951, doi: 10.1073/pnas.37.4.205.
- [9] W. Kabsch and C. Sander, "Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features," *Biopolymers: Original Research on Biomolecules*, vol. 22, no. 12, pp. 2577–2637, 1983, doi: 10.1002/bip.360221211.
- [10] D. P. Ismi and R. Pulungan, "Deep learning for protein secondary structure prediction: Pre and post-AlphaFold," *Computational and Structural Biotechnology Journal*, vol. 20, pp. 6271–6286, 2022, doi: 10.1016/j.csbj.2022.11.012.
- [11] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE Transactions on neural networks and learning systems*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022, doi: 10.1109/TNNLS.2021.3084827.
- [12] W. Fang, Y. Chen, and Q. Xue, "Survey on research of RNN-based spatio-temporal sequence prediction algorithms," *Journal on Big Data*, vol. 3, no. 3, pp. 97–110, 2021, doi: 10.32604/jbd.2021.016993.
- [13] S. Wang, J. Peng, J. Ma, and J. Xu, "Protein secondary structure prediction using deep convolutional neural fields," *Scientific reports*, vol. 6, no. 1, pp. 1–11, 2016, doi: 10.1038/srep18962.
- [14] C. Fang, Y. Shang, and D. Xu, "MUFOLD-SS: New deep inception-inside-inception networks for protein secondary structure prediction," *Proteins: Structure, Function, and Bioinformatics*, vol. 86, no. 5, pp. 592–598, 2018, doi: 10.1002/prot.25487.
- [15] J. Zhou and O. Troyanskaya, "Deep supervised and convolutional generative stochastic network for protein secondary structure prediction," in *International Conference on Machine Learning*, PMLR, pp. 745–753, 2014.
- [16] Z. Lin, J. Lanchantin, and Y. Qi, "MUST-CNN: a multilayer shift-and-stitch deep convolutional architecture for sequence-based protein structure prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, Feb. 2016, doi: 10.1609/aaai.v30i1.10007.
- [17] Y. Wang, J. Cheng, Y. Liu, and Y. Chen, "Prediction of protein secondary structure using support vector machine with PSSM profiles," in *2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference*, May 2016, pp. 502–505, doi: 10.1109/ITNEC.2016.7560411.
- [18] R. A. Ghamdi, A. Aziz, M. Alshehri, K. R. Pardasani, and T. Aziz, "Deep learning model with ensemble techniques to compute the secondary structure of proteins," *The Journal of Supercomputing*, vol. 77, pp. 5104–5119, 2021, doi: 10.1007/s11227-020-03467-9.
- [19] Y. Görmez and Z. Aydın, "ROSE: A novel approach for protein secondary structure prediction," in *Trends in Data Engineering Methods for Intelligent Systems*, J. Hemanth, T. Yigit, B. Patrut, and A. Angelopoulou, Eds., in *Lecture Notes on Data Engineering and Communications Technologies*. Cham: Springer International Publishing, 2021, pp. 455–464, doi: 10.1007/978-3-030-79357-9_45.
- [20] 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, 2019, doi: 10.1093/bioinformatics/bty1006.
- [21] G. Pollastri and A. McLysaght, "Porter: a new, accurate server for protein secondary structure prediction," *Bioinformatics*, vol. 21, no. 8, pp. 1719–1720, Apr. 2005, doi: 10.1093/bioinformatics/bti203.
- [22] A. Drozdetskiy, C. Cole, J. Procter, and G. J. Barton, "JPred4: a protein secondary structure prediction server," *Nucleic acids research*, vol. 43, no. W1, pp. W389–W394, 2015, doi: 10.1093/nar/gkv332.
- [23] L. Yuan, X. Hu, Y. Ma, and Y. Liu, "DLBLS_SS: protein secondary structure prediction using deep learning and broad learning system," *RSC Advances*, vol. 12, no. 52, pp. 33479–33487, 2022, doi: 10.1039/D2RA06433B.
- [24] W. Yang, Z. Hu, L. Zhou, and Y. Jin, "Protein secondary structure prediction using a lightweight convolutional network and label distribution aware margin loss," *Knowledge-Based Systems*, vol. 237, p. 107771, Feb. 2022, doi: 10.1016/j.knsys.2021.107771.
- [25] L. Yuan, Y. Ma, and Y. Liu, "Ensemble deep learning models for protein secondary structure prediction using bidirectional temporal convolution and bidirectional long short-term memory," *Frontiers in Bioengineering and Biotechnology*, vol. 11, 2023, doi: 10.3389/fbioe.2023.1051268.





- [26] "Protein data bank: the single global archive for 3D macromolecular structure data," *Nucleic acids research*, vol. 47, no. D1, pp. D520–D528, 2019, doi: 10.1093/nar/gky949.
- [27] A. V. Ikechukwu and S. Murali, "i-Net: a deep CNN model for white blood cancer segmentation and classification," *International Journal of Advanced Technology and Engineering Exploration*, vol. 9, no. 95, 2022, doi: 10.19101/IJATEE.2021.875564.
- [28] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv: 1207.0580*, 2017.
- [29] W. Jin, Z. J. Li, L. S. Wei, and H. Zhen, "The improvements of BP neural network learning algorithm," in *WCC 2000 - ICSP 2000. 2000 5th International Conference on Signal Processing Proceedings. 16th World Computer Congress 2000*, vol. 3, Aug. 2000, pp. 1647–1649, doi: 10.1109/ICOSP.2000.893417.
- [30] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," in *Adaptive computation and machine learning*. Cambridge, Massachusetts: The MIT Press, 2016.
- [31] C. C. Aggarwal, "Convolutional neural networks," in *Neural Networks and Deep Learning: A Textbook*, C. C. Aggarwal, Ed., Cham: Springer International Publishing, 2018, pp. 315–371. doi: 10.1007/978-3-319-94463-0_8.
- [32] A. V. Ikechukwu, S. Murali, R. Deepu, and R. C. Shivamurthy, "ResNet-50 vs VGG-19 vs training from scratch: A comparative analysis of the segmentation and classification of Pneumonia from chest X-ray images," *Global Transitions Proceedings*, vol. 2, no. 2, pp. 375–381, 2021, doi: 10.1016/j.glt.2021.08.027.
- [33] L. C. Paul, A. A. Suman, and N. Sultan, "Methodological analysis of principal component analysis (PCA) method," *International Journal of Computational Engineering and Management*, vol. 16, no. 2, pp. 32–38, March 2013.
- [34] S. Xie, Z. Li, and H. Hu, "Protein secondary structure prediction based on the fuzzy support vector machine with the hyperplane optimization," *Gene*, vol. 642, pp. 74–83, Feb. 2018, doi: 10.1016/j.gene.2017.11.005.
- [35] P. Chopra and A. Bender, "Evolved cellular automata for protein secondary structure prediction imitate the determinants for folding observed in nature," *In silico biology*, vol. 7, no. 1, pp. 87–93, 2007.
- [36] E. Asgari and M. R. Mofrad, "Continuous distributed representation of biological sequences for deep proteomics and genomics," *PLoS one*, vol. 10, no. 11, p. e0141287, 2015, doi: 10.1371/journal.pone.0141287.
- [37] Y. Guo, W. Li, B. Wang, H. Liu, and D. Zhou, "DeepACLSTM: deep asymmetric convolutional long short-term memory neural models for protein secondary structure prediction," *BMC Bioinformatics*, vol. 20, no. 1, pp. 1–12, 2019, doi: 10.1186/s12859-019-2940-0.
- [38] M. A. R. Ratul, M. T. Elahi, M. H. Mozaffari, and W. Lee, "PS8-Net: A deep convolutional neural network to predict the eight-state protein secondary structure," in *2020 Digital Image Computing: Techniques and Applications (DICTA)*, Nov. 2020, pp. 1–3, doi: 10.1109/DICTA51227.2020.9363393.
- [39] V. Enireddy, C. Karthikeyan, and D. V. Babu, "OneHotEncoding and LSTM-based deep learning models for protein secondary structure prediction," *Soft Computing*, vol. 26, no. 8, pp. 3825–3836, 2022, doi:10.1007/s00500-022-06783-9.

BIOGRAPHIES OF AUTHORS






Srushti C. Shivaprasad     has received the Bachelor's (B.E) degree from the Department of CSE, at RNSIT, Bengaluru under Visvesvaraya Technological University (VTU), Belgavi, India in 2009, and the M.Tech Degree from The Oxford College of Engineering, Bengaluru under VTU in the year 2012 and currently pursuing Ph.D. in CSE from University Visvesvaraya College of Engineering, Bangalore University, India. Since 2012, she has worked as an Assistant Professor for 9 years. She has authored 3 research articles. Her area of interest includes deep learning and proteomics. She can be contacted at email: srushtichan@gmail.com.






Dr. Prathibhavani P. Maruthi     is working as an Assistant Professor in the Department of Computer Science and Engineering, University of Visvesvaraya College of Engineering, Bangalore. Prathibhavani holds a Ph.D. (CSE - Engg.) from VTU, Belagavi in the area of Wireless Sensor networks. She obtained the post-graduation in the Department of Information Technology from University Visvesvaraya College of Engineering, Bangalore University, Bengaluru. She has more than 14 years of teaching and research experience. She has taught several courses for both UG and PG levels in the area of WSN, Electronic Circuits, Adhoc Networks, Mobile Computing, Advanced Digital Communication, Mobile Device Forensics, Cyber Space and Probability Stochastic and Queuing Theory. Her research area is Wireless Sensor Network. She has guided several projects on wireless sensor network for both UG and PG students. She has published more than 24 technical national, international conferences and journals in the area of Wireless Sensor Network. She is a life member of IAENG, ISTE. She is associated with the Old Dominion University, Virginia, Norfolk USA as research project coordinator of Department of ISE, Acharya Institute of Technology, Bangalore from the year 2010 to 2019. She has guided several students in ODU program and sent several students to ODU, USA to get benefit of internship from the year 2010 to 2019. She can be contacted at email: prathibhavani.pm@uvce.ac.in.



Teja Shree Venkatesh    has received Bachelor's (BE) degree from Alpha College of Engineering in 2019 and masters (M.Tech) degree in CSE from University Visvesvaraya College of Engineering, Bangalore University, India. Her research interest includes artificial intelligence machine learning, deep learning. She can be contacted at email: tejasremoni17@gmail.com.



Dr. Venugopal K. Rajuk    is an IEEE Fellow and ACM Distinguished Educator. He was the former Vice-Chancellor of Bangalore University. He was former Principal, University Visvesvaraya College of Engineering, Bangalore University, Bangalore. He obtained Bachelor of Engineering from University Visvesvaraya College of Engineering and Master degree in Computer Science and Automation from Indian Institute of Science Bangalore. He was awarded Ph.D. in Economics from Bangalore University and Ph.D. in Computer Science from Indian Institute of Technology, Madras. He has a distinguished academic career and has degrees in Electronics, Economics, Law, Business Finance, Public Relations, Communications, Industrial Relations, Computer Science and Journalism. He has authored and edited 84 books on Computer Science, which include Petrodollar and the World Economy, C Aptitude, Mastering C, Microprocessor Programming, Mastering C++, Soft computing for Data Mining. During his five decades of service at UVCE, he has over 1100 research papers to his credit and holds 38 patents. His research interests include computer networks, parallel and distributed systems, digital signal processing and data mining, computer networks, IoT, and computer networks. He can be contacted at email: venugopalkr@gmail.com.