Tamjeed Journal of Healthcare Engineering and Science Technology E-ISSN: 2583-7281
Volume **1** (1): 2023, page 35-42, https://doi.org/10.59785/tjhest.v1i1.4 https://tamjed.com/

Detection of Human Protein Structures by Select Deep Learning Models and Dynamic Systems

Dr.B.Santhosh Kumar[1]

[1]Professor, Department of Computer Science &Engineering, Guru Nanak Institute of Technology Hyderabad, Telangana, India
Email: b.santhoshkumar@gmail.com; drsanthoshkumar81@gmail.com

Dr.Badamasi Sani Mohammed[2]

[2]Lecturer, Department of Economics, Al-Qalam University Katsina, Nigeria
E-mail: sanibadamasi9@gmail.com; sanibadamasi@auk.edu.ng

Abstract

Peptide bonds bind amino acids together to form proteins. Because of the way amino acids fold, proteins have a three-dimensional structure. Increased data, or samples, improves machine learning models that learn from examples. ANNs, a single-layer learning system, require sophisticated systems and time to learn massive volumes of data over time. Today, the multi-layer deep learning technique is preferred. Deep learning uses artificial neural networks. This part of the work provides technical details about the deep learning methods used. This research's deep learning methods include CNN, Hidden Markov Model (HMM), Recurrent Weighted Average (RWA), and Conditional Random Fields (CRF). The training and testing results of these models are presented in this work. For this work, the CNN, HMM, RWA, and CRF algorithms were all compared using the CB513 dataset. In addition, each method was analyzed and contrasted with other research conducted previously. In this particular research endeavour, the respective success rates of the four models that were built for predicting the protein secondary structure were as follows: 82.54%, 81.06%, 81.10%, and 81.48%. The models and working environment produced in this study can be used to predict protein secondary structure quickly. In deep learning experiments, data amount affects learning. Increasing data will test the study's models.

Keywords: Deep Learning models, Dataset, Protein structures, algorithms, ANNs

1. Introduction

Proteins play an essential role in almost every biological process of a living organism. Their unique amino acid sequences determine proteins' functions and three-dimensional structures [1]. Proteins with different amino acid sequences have different functions [2]. Therefore, knowing the amino acid sequence of proteins is an essential step in explaining their biological activity. Proteins are divided into four levels according to their structure; primary, secondary, tertiary, and quaternary. The polypeptide chain, formed by the peptide bonds of amino acids together, is defined as the primary structure. Any change in this polypeptide chain affects the protein's three-dimensional structure and, therefore, its activity. The secondary structure is formed due to regional folding in the polypeptide chain; this folding occurs with hydrogen bonds formed between the carboxyl and amino groups of amino acids. The tertiary structure is the structure formed by the interactions between the R groups of amino acids in the secondary structure, in which proteins

acquire their three-dimensional structure. The structure in which the proteins consisting of more than one polypeptide chain are found is defined as the quaternary structure. Knowing the protein's three-dimensional structure is very important for understanding the function of genes, detecting diseases caused by errors in protein folding and drug design, and understanding the protein's function [1,3]. Experimental techniques such as X-ray diffraction, nuclear magnetic resonance (NMR) and electron crystallography are used to determine the three-dimensional structures of proteins. However, determining these structures with laboratory studies is very costly and challenging, and it is impossible to use these techniques for every protein [3]. Estimating the three-dimensional structures of proteins from the primary structure that forms them is considered a difficult problem [4]. Therefore, the prediction of secondary structure from the amino acid sequence (primary structure) is important in understanding protein structure and function. Over the years, various methods have been used to predict protein secondary structure. Computerised computation techniques have been widely used to solve this problem with the increase in protein data information in the protein data bank (PDB).

Machine learning approaches such as artificial neural networks, support vector machines, and genetic algorithms are among the methods used in this research area. Besides machine learning techniques, hybrid methods have been developed in which more than one successful method is used together to increase the success rate in protein secondary structure prediction [4]. In recent years [5], with the achievement of successful prediction results with deep learning methods such as CNN and HMM, it has been aimed to increase prediction success in studies where these methods are used together. With its success in solving complex problems and the opportunity to work with large data sets, deep learning methods have been frequently preferred in studies in the field of bioinformatics, as in many other fields [5-7]. With the studies in this field, Convolutional Neural Networks and Recursive Neural Networks, which are deep learning algorithms, are among the essential methods used in protein secondary structure prediction [8-9]. Recent studies have revealed that deep learning models increase the success of protein secondary structure prediction, as in many different complex problems. In their tests on different datasets, Li and Yu used multiscale CNN and bidirectional recurrent gate unit layers, which they called Deep convolutional recurrent neural network, 69.7% for the CB513 dataset, 76% for the CASP10 dataset. 9 and 73.1% Q8 success for the CASP11 dataset [10]. They also stated that this is the first known study in the literature to use CRF for PYT. In the study by Guo et al., an eight-class protein secondary structure prediction study was performed using bidirectional long-short-term memory and asymmetric CNN [11]. According to the study results, this algorithm achieved a Q8 success of 75% in the CASP10 dataset and 73% in the CASP11 dataset. Wang et al. trained with the CNN and RWA layers network, using the 13-window PSSM matrix as training input and achieved 80.18% Q3 success [10-11]. This work mentions information about the deep learning algorithm developed and its parameters. CNN, HMM, RWA and CRF deep learning models were studied to predict protein secondary structure, and their performances were compared. The CB513 dataset was used in the training and testing phases.

2. Materials and Methods

2.1. Data set

CB513, an open dataset of 513 proteins and 84,119 amino acids created by Cuff and Barton [12], was used in this work. The CB513 dataset, which was arranged using PSSM feature vectors obtained using PSI-BLAST and HHBlits alignment methods and structural profile matrices obtained by the DSPRED method, was obtained from the study of Zhang, et al.[13]. In the dataset, each amino acid is represented by 539 attributes. These features show the interaction of 49 features with the amino acids around the target amino acids, obtained with two 20xN sized PSSM matrices and three 3xN sized structural profile matrices representing the secondary structure of the protein, as a result of aligning the amino acid sequence of each protein in the CB513 dataset with the specified methods. It was stated that it was obtained by using 11 unit-long windows to measure. Proteins are represented as 20xN, as 20 types of amino acids are known in nature. N denotes the number of amino acids present in the target protein, as the number of amino acids in proteins can differ. Similarly, in tri-class protein secondary structure prediction studies, the secondary structure of proteins is represented by matrices of size 3xN.

The dataset used consisted of seven cross-validation sets of 513 proteins, each layer then used for testing. K-fold cross-validation is used to test the performance of the model. Each model will be trained and tested seven times using different datasets.

2.2. Development Environment

2.2.1. Google Colaboratory: Google Colaboratory is a cloud-based platform that offers free GPU access for applications that require powerful hardware equipment, especially machine learning, data analysis, and deep learning. It allows these applications to be developed via an internet browser with Python [14]. The applications realised in this work study were carried out using GPU on this platform. For GPU execution, Colaboratory defines 12 hours of continuous use to users; a temporary restriction is applied at the end of this period. At the end of the restriction, GPU usage is allowed again. However, the mentioned time was sufficient

for the execution of the studies, as the use of GPU speeds up the working time quite a bit.

2.2.2. Libraries used: This work developed five different deep learning networks on the Collaboratory platform using Python programming language to predict tri-class protein secondary structure. Various libraries were used for data transfer, preprocessing, designing the deep learning network, training, testing, and evaluation stages. During the implementation of the applications, mainly; NumPy, Pandas, Scikit-Learn, Seaborn, Matplotlib and Keras libraries are used.

NumPy is an open-source library that allows mathematical operations with arrays and matrices. Pandas is a data analysis tool that can efficiently perform many operations, such as reading data from different file types, analysing it, and writing it back to the file. NumPy and Pandas libraries were used for loading and editing the data to be used in the study.

Scikit-Learn is a machine learning library that can work integrated with different libraries. This study obtained the calculations required to measure model successes using this library. The Seaborn and Matplotlib libraries are referenced for data visualization and graphing.

Keras is an open-source deep-learning library written in Python programming language that can run over Tensorflow or Theano libraries. All deep learning networks mentioned in this work were created using the Keras library.

2.3. Developed Deep Learning Models: In this work, four different deep learning networks [15], CNN, HMM, RWA and CRF, were created to predict the secondary structure from the primary structure of the protein mentioned in the second chapter. The creation of models and network structures are shared under this title. To use it in the training of the models, a connection was established between Google Drive et al., where the data is located, and the data was transferred to the program. Since 7-fold cross-validation was applied for the dataset used, the data for training were resized as Nx49x11 for each cross-validation set, with N being the number of rows of the dataset, to represent the 11 window sizes and 49 features identified for each of the amino acids. 10% of the training data allocated for each floor is reserved as a validation set to be used during the training. After the data is prepared for training, the network layers and parameters are defined, and the model is trained. After the model's training, the success and loss functions of the training and validation tests were examined. The test data outputs were estimated with the trained network, evaluation metrics were calculated, and graphs were prepared to compare the secondary structure classes and predicted classes. The training process for all models requires the same operations. The flow chart showing the network training of the models is shown in Figure 1.

2.3.1. CNN model [16]: The Convolutional Neural Networks model, developed to perform protein secondary structure prediction, consists of dilution layers between three convolution layers, flattening and fully connected layers. In the convolutional layers of the CNN network used in this study, 128, 64, 32 filters, 5, 3, and 3 kernels and ReLu activation functions were used, respectively. L2 regulation with a value of 0.001 has been added as kernel regulation in convolution layers. A value of 0.20 was determined for the tracking layers located between the convolutional layers. To estimate the three classes representing the secondary structure of the protein in the last layer, the output size was determined as three, and the softmax function was used. After the network structure and parameters were determined, 124195 parameters were trained, and the model's training was completed at the end of 20 epochs. The learning coefficient used in training was 0.0001, Adam's optimisation algorithm, and the heap size was 64.

2.3.2. HMM model: The second model [17] for estimating secondary structure classes was developed using Hidden Markov Model. The input layer consists of two dense layers, the two HMM layers and the output layer at the end. The two HMM layers in the network structure contain 64 neurons, and the dense layer contains 32 neurons and uses the relu activation function. As stated in the CNN model, the size of the output layer is three, and the activation function is softmax. HMM layers can return the sequence. The input data transmitted to this layer is transmitted to the next layer. The sequence rotation feature is used in the first layer. Sequence rotation is not used in this layer, as there is a dense layer after the second layer, and the input size does not match the sequence size. After the network layers are determined, the training parameters are determined; learning coefficient 0.0001, optimisation algorithm Adam and heap size 64. A total of 15299 parameters were trained, and the model's training was completed at the end of 15 training rounds.

2.3.3. **RWA model:** The other model developed for the tertiary structure classification study consists of two RWA layers [18], a dense and an output layer. Similar to the HMM model, this model has two RWA layers, while the sequence rotation feature is used in the first layer. One hundred twenty-eight neurons are defined in the first layer and 64 neurons in the second layer. Parameters and activation functions are used in the network layers of the RWA model. The activation function in the RWA layers is determined as tanh, relu in the fully connected dense layer, and softmax in the output layer. 123267 parameters in the model are trained using the Adam optimisation algorithm in 20 steps, with a learning coefficient of 0.0001 and a heap size of 64.

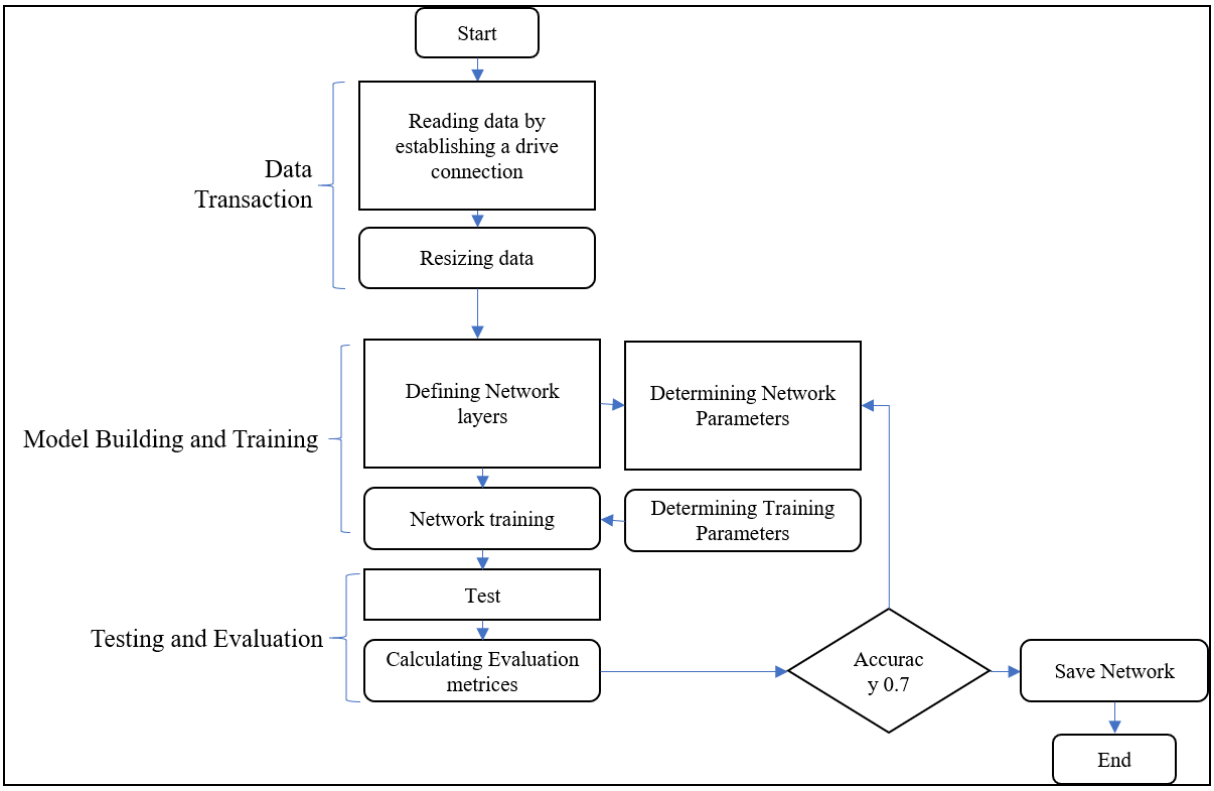


Figure 1: Network training flowchart

2.3.4. CRF model: The fourth model developed is the Gated Iterative Unit model. In this model, the input layer, two CRF layers [19], followed by the dilution layer and dense layers, are used. The CRF model includes CRF layers consisting of 100 and 50 neurons in the first two layers. After the CRF layers, there is the Dropout layer that applies 0.20 dilution. Finally, a dense layer of 50 neurons and a final layer with three outputs. In this model, tanh is used as the activation function in the layers. This is because the default function of the CRF layers is tanh. When the activation function is changed, the model has to work on the CPU and the training time is considerably slower. The model contains 75153 trainable parameters, and training is carried out in 20 steps. It is trained using the Adam optimisation algorithm and a learning coefficient of 0.0001. The heap size is set to 64.

2.4. Evaluation Metrics

After the training of the models was carried out, the testing phase started. To compare test results effectively, metrics used in classification problems were calculated. The details of the metrics are shared under this title. While evaluating the classification problems, the values produced by the model as a result of the estimation are examined in four cases. These; true positive (True Positive, TP) for correctly predicted positive classes, false positive (False Positive, FP) for falsely predicted positive classes, true negative (True Negative, TN) for correctly predicted negative classes, and falsely predicted negative classes It is called a false negative (False Negative, FN). Accuracy, sensitivity and precision, among the model evaluation criteria calculated using these conditions, were preferred to compare the models in this study.

- **Success rate**: Accuracy is the most common measure used to evaluate the overall success of classification models, determined by the ratio of the number of samples correctly predicted by the model to the total number of samples.

Success rate

$$= TP + TN/TP + FP + TN + FN \quad (4.1)$$

- **Sensitivity:** Sensitivity represents the ratio of correctly predicted positive classes (TP) to all cases that need to be positively predicted (TP+FN).

$$\text{Sensitivity} = TP/TP + FN \quad (4.2)$$

- **Precision:** It is defined as the criterion precision (precision), which is calculated by the ratio of correctly predicted positive classes (TP) to all positively predicted cases (TP+FP).

$$\text{precision} = TP/TP + FP \quad (4.3)$$

- **F1 score**: The F1 Score is an evaluation criterion obtained by calculating the harmonic mean of sensitivity and precision values. It is often preferred,

especially in sets where the data distribution between classes is not equal.

$$F1\ score = 2 \times \frac{Sensitivity \times Precision}{Sensitivity + Precision} \quad (4.4)$$

The mentioned performance evaluation metrics, precision, sensitivity and F1 score, are calculated separately for each class in multi-class estimation studies.

3.0 Results and Discussion

This work created four different deep learning models for three-class secondary structure prediction of proteins, and the models were compared between them. The CB513 dataset was used to train and test the models. Precision, sensitivity, F score and success rate values were calculated for each model, and the results were evaluated. The models were trained on the GPU using the Google Collaboratory service, and the training times were compared. While calculating the averages of the success rates and evaluation metrics of the four models created in this study, the weighted average of the results obtained with each of the 7-fold cross-validation sets applied in the data set was taken.

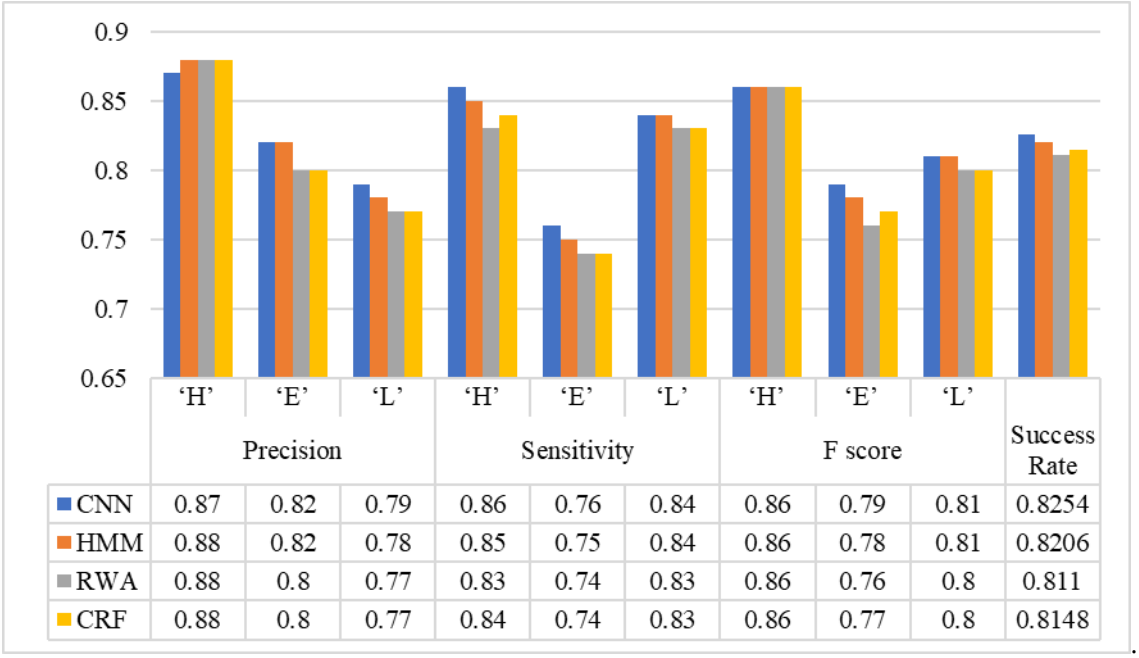
As mentioned in the previous sections, four different deep learning models developed in this study were trained using the same data set. After completing the training, the graphs showing the development of the training success rate and the loss function during the specified training step were examined, and the training compliance was checked. Graphs of varying success rates and loss functions of Convolutional Neural Network (CNN), Hidden Markov Model (HMM), Recurrent Weighted Average (RWA) and Conditional Random Field (CRF) models during training for each cross-validation set Figure 2.

Figure 2 includes the test results of the CNN model. The mean F score values for the H, E and L classes were calculated as 0.86, 0.79 and 0.81, respectively. When the results are examined, it is seen that the helix class represented as 'H' is predicted more successfully than other classes. The average success rate of the model was calculated as 0.8254. Figure 2 shows the complexity matrix calculated due to training and testing the set with the first cross-validation of the CNN model. When the matrix is examined, it is seen that 3079 of 3553 class 'H' data, 1939 of 2672 class 'E' data and 3591 of 4272 class 'L' data in the test set are predicted correctly.

Figure 2 contains the evaluation metrics calculated according to the test results performed for each cross-validation set of the HMM model. The mean F score values for the H, E and L classes were calculated as 0.86, 0.78 and 0.81, respectively. When Figure 2 is examined, it is seen that the best-predicted class is 'H', and the average success rate of the model is 0.8206. Figure 2 shows the complexity matrix calculated due to training and testing the HMM model with the first set of cross-validation. When the matrix is examined, it is seen that 3050 of 3553 class 'H' data, 1833 of 2672 class 'E' data and 3652 of 4272 class 'L' data in the test set are predicted correctly. Complexity matrices calculated as a result of tests with other cross-validation sets are given in the Appendices. While the average success rate was calculated as 0.8110 in the RWA model, it was the model with the lowest performance among the developed models. The mean F score values for the H, E and L classes were calculated as 0.86, 0.76 and 0.80, respectively. As in other models, it is seen that the prediction rate of the 'H' class is higher than the other classes. (Figure 2)

Figure 2 shows the complexity matrix calculated due to training and testing the RWA model with the third set of cross-validation. When the matrix is examined, it is seen that 2873 of 3553 class 'H' data, 1840 of 2672 class 'E' data and 3699 of 4272 class 'L' data in the test set are predicted correctly.

The average success rate of the CRF model was calculated as 0.8148. The mean precision values for the H, E and L classes were calculated as 0.86, 0.77 and 0.80, respectively. It is seen that the successful prediction rate of class H is higher than other classes. (Figure 2)



	Precision			Sensitivity			F score			Success Rate
	'H'	'E'	'L'	'H'	'E'	'L'	'H'	'E'	'L'	
CNN	0.87	0.82	0.79	0.86	0.76	0.84	0.86	0.79	0.81	0.8254
HMM	0.88	0.82	0.78	0.85	0.75	0.84	0.86	0.78	0.81	0.8206
RWA	0.88	0.8	0.77	0.83	0.74	0.83	0.86	0.76	0.8	0.811
CRF	0.88	0.8	0.77	0.84	0.74	0.83	0.86	0.77	0.8	0.8148

Figure 2: Cross-validation of Deep learning Model results on average

Figure 3 shows the complexity matrix calculated as a result of training and testing the CRF model with the first set of cross-validation. When the matrix is examined, it is seen that 3010 of 3553 class 'H' data, 1840 of 2672 class 'E' data and 3640 of 4272 class 'L' data in the test set are predicted correctly.

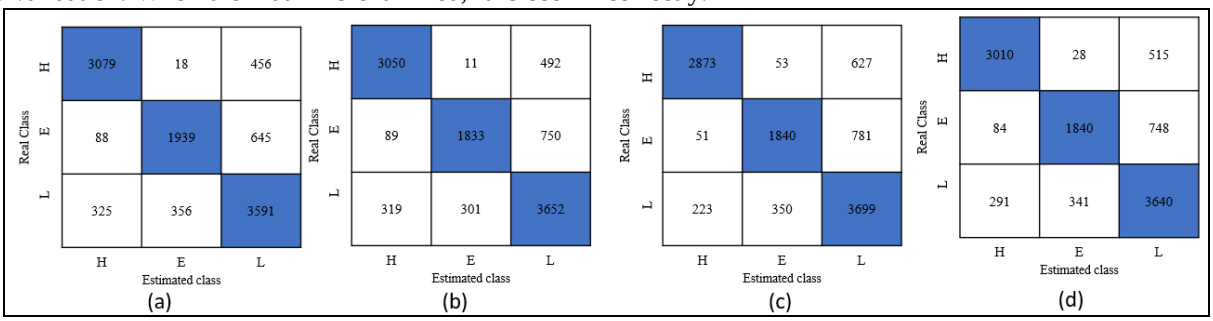


Figure 3: First cross-validation set complexity matrix for

(a) CNN, (b) HMM, (c) RWA and (d) CRF model

Table 2. shows the total training time, average F score, average success rate and standard deviation values of the success rates of the models. The training time for each cross-validation set is approximately 1 min in the CNN model. 57 sec., approx. 2 min in RWA model. 48 sec., approx. 2 min in CRF model. 18 sec. in progress. Since training in the HMM model takes longer to perform with the GPU than with the CPU, the training is performed using the CPU and takes approximately 7 min for a cross-validation set. 16 sec. It took.

Table 2: Comparison of the performances of the models

	Training time (minutes)	F-Score	success rate	Standard deviation
CNN	13.39	0.82	0.8254	0.01
HMM	50.52	0.82	0.8206	0.0081
RWA	19.36	0.81	0.811	0.0087
CRF	16.18	0.81	0.8148	0.0087

4. Conclusions & Recommendations

In this work, deep learning methods were used for secondary structure prediction of proteins, which are an essential part of living organisms. Although protein structure can be determined from the primary structure, it is difficult to predict the entire structure from the primary structure alone. Therefore, secondary structure prediction is an important and challenging step in predicting the protein's three-dimensional structure. In this study, CNN, HMM, RWA and CRF models, which are deep learning models, were used to predict the secondary structure of the protein from the primary structure. The study was carried out using the CB513 dataset with the network models specified in the Google Collaboratory environment. When the test results are compared over the success rate, the CNN model was the most successful deep learning model with 82.54%, while the RWA was the least successful deep learning model with a value of 81.1%. Compared to F scores, CNN and HMM models achieved 82% better results than the other two models by 1%.

When the total training times are examined among the models performed on the GPU, it is 13 min. 39 sec. CNN has been the fastest running network. RWA is 19 min. 36 sec. It was the slowest running model. As a result of the test, the HMM model was performed with the CPU since it runs slower on the GPU than it works on the CPU, and the total training time is 50 minutes. 52 sec. also yielded results. As a result, the success rates of the models used were close to each other. It has been seen that all four models used in this study can be used when predicting protein secondary structure with deep learning methods. In other studies in this field, it has been observed that there is a great difference between the training periods depending on the development environment and method used. In other studies, while training took days, time was obtained based on minutes in the study conducted.

For this reason, it is seen that the models and working environment realised in this study can be used in protein secondary structure prediction studies where fast results are essential. It is known that the effect of the amount of data on learning is important in deep learning studies. The success of the models proposed in the study can be tested by increasing the amount of data.

Reference

[1]. Chowdhury, Nilkanta & Bagchi, Angshuman. (2015). An Overview of DNA-Protein Interactions. Current Chemical Biology. 09. 1-1. 10.2174/2212796809666151022202255.

[2]. Alhalmi, Abdulsalam & Ali, Nafaa & Abdulrahman, Amer. (2020). Intracellular Protein Biosynthesis: A Review. Asian Journal of Biochemistry Genetics and Molecular Biology. 2. 10-18. 10.9734/AJBGMB/2020/v5i230125.

[3]. Taha, Kamal & Iraqi, Youssef & Aamri, Amira. (2019). Predicting protein functions by applying predicate logic to biomedical literature. BMC Bioinformatics. 20. 10.1186/s12859-019-2594-y.

[4]. Malik, Ali & Malik, Emad & Mohammed Ali Al-Shammaa, Nawal & Al-Rubaei, Zeinab. (2010). A Comparative Biochemical Study of Proteins Profile in Iraqi Children and Adolescent with ?-Thalassemia. 19. 19-23. 10.31351/vol19iss2pp19-23.

[5]. Hasic, Haris & Buza, Emir & Akagic, Amila. (2017). A Hybrid Method for Prediction of Protein Secondary Structure Based on Multiple Artificial Neural Networks. 10.23919/MIPRO.2017.7973605.

[6]. Dorn, Marcio & Breda, Ardala & Norberto de Souza, Osmar. (2008). A Hybrid Method for the Protein Structure Prediction Problem. 47-56. 10.1007/978-3-540-85557-6_5.

[7]. Cheng, Jianlin & Tegge, Allison & Baldi, Pierre. (2008). Machine Learning Methods for Protein Structure Prediction. Biomedical Engineering, IEEE Reviews in. 1. 41 - 49. 10.1109/RBME.2008.2008239.

[8]. Yoo, Paul & Zhou, Bing & Zomaya, Albert. (2008). Machine Learning Techniques for Protein Secondary Structure Prediction:An Overview and Evaluation. Current Bioinformatics. 3. 74-86. 10.2174/157489308784340676.

[9]. Bui, Quoc-Chinh & Katrenko, Sophia & Sloot, Peter. (2010). A hybrid approach to extract protein–protein interactions. Bioinformatics (Oxford, England). 27. 259-65. 10.1093/bioinformatics/btq620.

[10]. Zhang, Buzhong & Li, Jinyan & Lü, Qiang. (2018). Prediction of 8-state protein secondary structures by a novel deep learning architecture. BMC Bioinformatics. 19. 10.1186/s12859-018-2280-5.

[11]. Liimatainen, Kaisa & Huttunen, Riku & Latonen, Leena & Ruusuvuori, Pekka. (2021). Convolutional Neural Network-Based Artificial Intelligence for Classification of Protein Localization Patterns. Biomolecules. 11. 264. 10.3390/biom11020264.

[12]. Azgınoğlu, Nuh & Aydin, Zafer & Celik, Mete. (2019). Developing Structural Profile Matrices for Protein Secondary Structure and Solvent Accessibility Prediction. Bioinformatics. 35. 10.1093/bioinformatics/btz238.

[13]. ZHANG, YONGQING & Qiao, Shaojie & Ji, Shengjie & Li, Yizhou. (2020). DeepSite: bidirectional LSTM and CNN models for predicting DNA–protein binding. International Journal of Machine Learning and Cybernetics. 11. 10.1007/s13042-019-00990-x.

[14]. Tariq, Tayyaba & Frezund, Javed & Farhan, Muhammad & Latif, Rana & Mehmood, Azka.

(2020). Structure Analysis of Protein Data Bank Using Python Libraries. 201-209. 10.1109/IBCAST47879.2020.9044525.

[15]. Stepniewska-Dziubinska, Marta & Zielenkiewicz, Piotr & Siedlecki, Pawel. (2018). Development and evaluation of a deep learning model for protein-ligand binding affinity prediction. Bioinformatics (Oxford, England). 34. 10.1093/bioinformatics/bty374.

[16]. Das, Bihter & Toraman, Suat. (2020). Classifying Protein Sequences Using Convolutional Neural Network. Bitlis Eren Üniversitesi Fen Bilimleri Dergisi. 9. 1663-1671. 10.17798/bitlisfen.662816.

[17]. Bystroff, Christopher & Krogh, Anders. (2008). Hidden Markov Models for Prediction of Protein Features. Methods in molecular biology (Clifton, N.J.). 413. 173-98. 10.1007/978-1-59745-574-9_7.

[18]. Hakala, Kai & Kaewphan, Suwisa & Bjorne, Jari & Nlp, Farrokh & Moen, Hans & Tolvanen, Martti & Salakoski, Tapio & Ginter, Filip. (2020). Neural Network and Random Forest Models in Protein Function Prediction. IEEE/ACM Transactions on Computational Biology and Bioinformatics. PP. 1-1. 10.1109/TCBB.2020.3044230.

[19]. Johansen, Alexander & Sønderby, Casper & Sønderby, Søren & Winther, Ole. (2017). Deep Recurrent Conditional Random Field Network for Protein Secondary Prediction. 73-78. 10.1145/3107411.3107489.

[20]. Rashid, Shamima & Saraswathi, Saras & Kloczkowski, Andrzej & Suresh, Sundaram & Kolinski, Andrzej. (2016). Protein secondary structure prediction using a small training set (compact model) combined with a Complex-valued neural network approach. BMC Bioinformatics. 17. 10.1186/s12859-016-1209-0.