# Evaluating Machine Learning Models for Predicting Protein Secondary Structures

Paul Winpenny
*University of Southampton*
*Email: plw1g21@soton.ac.uk*

## 1. Introduction

Proteins are complex molecules that perform several functions through their structures, which are determined by the sequence of amino acids, known as the primary structure. The folding of these linear sequences into three-dimensional shapes is done by interactions among the amino acids, leading to the formation of secondary structures, such as alpha-helices, beta-strands, and coils.

Understanding the structure of proteins allows researchers to understand their functions. Accurate predictions can reveal potential interaction sites for medical and biotechnological applications, such as guiding drug design.

## 2. Background

Research into predicting secondary structure has been a focus of computational biology for decades. A significant advancement was made with Qian and Sejnowski's 1988 paper, which outlined a neural network approach that improved the success rates of secondary structure prediction from 53% to 64.3% [1]. This improvement was achieved through the use of a cascading network architecture, which allows for deep learning from the sequence data by reevaluating its outputs at each step.

The cascading network is a neural network architecture where each layer's output is not only passed forward to the next layer but also the output of one network is passed into another as the input. This adds additional complex decision-making, reevaluating its previous outputs at each step, enabling more advanced learning from the sequence data. For these networks, the sigmoid activation function is important and is used across each of the layers. This function is defined as

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

This function efficiently maps input values into a range between 0 and 1, mimicking neurons firing in biological systems. Such an activation function is particularly useful in layers of a neural network, as it allows each neuron to decide whether to pass on its signal. The sigmoid function was likely chosen due to the available activation functions at the time.

Alongside the sigmoid function, the backpropagation algorithm is important in the training process of neural networks. This algorithm optimises the network's weights and bases through error reduction. It first calculates the loss/error between the predictions and actual targets. Then it propagates the loss backward through the network, adjusting the weights to minimize errors. While the method had been around for years, the importance of this method was only realised in Rumelhart's 1986 paper [2]. In this study, backpropagation was used to effectively train multi-layer networks, which improved on models that could only handle linearly separable tasks. As protein structures rely on both local and global context when being created, the ability of neural networks to learn from complex patterns in data is important for predictions.

For Qian and Sejnowski's network, a likely optimiser would be Stochastic Gradient Descent (SGD), an important algorithm that was used in early research into machine learning [3]. As an optimiser, SGD updates the weights incrementally for each batch of training data.

Although Qian and Sejnowski's network marked a great improvement over previous models, the field of computational biology has witnessed advancements since their work. The evolution of machine learning

technologies has led to the development of more sophisticated models for predicting protein structures.

Notably, the use of convolutional neural networks (CNNs), which excel in recognizing patterns in sequential data, has been particularly effective [4]. For example, a recent study featuring a combination of Convolutional Neural Networks and Gated Recurrent Neural Networks (MCNN-GRNN) achieved an accuracy of 82% in predicting secondary protein structures [5].

In addition to deep learning approaches, other machine learning techniques have also shown promising results in this area. Support Vector Machines (SVMs) have been found to be effective in predicting different protein structures. In particular, they can work well with limited training data [6]. Another key study by Ding and Dubchak (2001) applied SVMs to classify proteins into different folding types based on their amino acid sequences. The accuracies reported in this study vary by the complexity of the dataset and the specific folding types, but they generally demonstrate the high effectiveness of SVMs in protein structure prediction tasks [7].

Moreover, Random Forests, a learning method involving multiple decision trees to make more accurate predictions, have been employed to handle the inherently complex nature of biological data [8]. An example of this is seen in the work by Shen and Chou, where Random Forests were used to predict the localization of proteins with an accuracy of about 90%, indicating its effectiveness in biological prediction [9].

While the cascaded model proposed by Qian and Sejnowski's paper was innovative for the time, modern advancements continue to improve the accuracy of prediction. As such, testing new methods that incorporate older and more modern techniques could improve the accuracy of biological networks.

## 3. Methodology

### 3.1. Initial Reimplementation

The reimplementation process begins with the preparation of the training and test datasets provided by the 1988 paper [1]. Each amino acid and its corresponding label in the protein sequence are encoded using a one-hot encoding scheme. This results in a binary vector representation for each amino acid.

After, to capture local information from the sequence, sliding windows of 13 amino acids are constructed. To ensure that the central amino acid in each window is supported by its neighbors, padding consisting of six zero-vectors is added to both the beginning and the end of the sequence. This padding ensures that windows at the front and back of the sequence have the same number of groups as those in the middle.

Once the sliding windows are prepared, training of the cascaded network begins. The first network in the model features a single hidden layer composed of 40 units. This layer utilizes the sigmoid activation function. The output layer consists of three units, each predicting a specific type of secondary protein structure: alpha-helices, beta-strands, or random coils. Each output unit uses a softmax activation function, which converts the raw output scores from the network into probabilities.

The output from the first network then serves as the input to a second network. It is identical in architecture to the first, with one hidden layer of 40 units and 3 units on the output layer. This is done in accordance with the paper's cascaded network model.

### 3.2. Single Improvement: Addition of Conv1D Layer

Next, the goal was to improve the network with one alteration. This enhancement of the neural network involves the addition of a Conv1D layer, which complements the existing dense layers. The newly integrated Conv1D layer has 64 filters with a kernel size of 3. This layer was added to the first network of the cascading model, with the second network being unchanged. With this addition, there were additional changes to facilitate the Conv1D adding more complexity to the network:

- L2 Regularization: Applied on the Conv1D layer to reduce overfitting by penalizing large weights.
- Batch Normalization: Follows the Conv1D layer to stabilize training by normalizing the activations.
- MaxPooling1D: Reduces the output dimensions after Conv1D, focusing on the most important features.

These adjustments are intended to improve the model's learning capabilities but also try to keep the

model close to the original. The original dense layer with 40 units is retained to preserve the original structure of the network, and the two layers are both activated by the sigmoid function.

### 3.3. Improvements- SVM

To explore alternative machine learning methodologies for secondary structure prediction, an SVM model was implemented. To optimise the performance, the model was configured with the following parameters:

- Kernel: The Radial Basis Function (RBF) kernel was chosen for its effectiveness in handling non-linear data structures. While proteins are affected by the local context of their amino acids, the global context, which includes distant interactions affecting structure formation, is also important. The RBF kernel is able to identify these relationships.
- Regularization Parameter (C): Set to 1 to balance the trade-off between achieving a low error on the training data and minimizing model complexity for better generalization.
- Gamma: It is set to ”scale”, which adjusts gamma based on the number of units to prevent overfitting.
- Probability Estimates: The model is set to compute the probability that each prediction is correct.

The data used for training the SVM are pre-processed in the same manner as in the original model, encoded and, segmented into sliding windows.

### 3.4. Improvement- Random Forest

A Random Forest model has also been tested. First, a small grid search is run on to find the optimal parameters, including the number of estimators/trees in the forest, the maximum depth of each tree, etc. Once this is done, trees are created in the forest, which are made by taking random samples. The trees are then built. Each tree makes a prediction and then through majority voting, the final output is determined.

### 3.5. Testing on unseen data

To evaluate the performance of the models on data not previously used in the 1988 paper, the RS126 dataset was used. The RS126 Dataset contains 150 protein sequences [10]. Upon retrieving the data from the text file, it underwent the same encoding process as the previous datasets. The method of parsing the data was amended due to the different format the RS126 dataset is stored in. Each model is run and the percentage accuracy is returned.

## 4. Results

The percentages for the accuracy of specific structure predictions were generated through observation of a normalised confusion matrix (An example is provided in the program files).

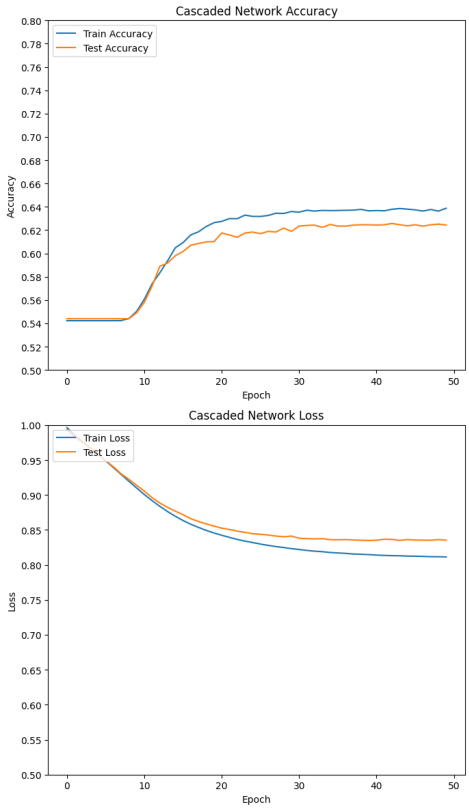


Figure 1. Model accuracy and loss of the initial model over epochs.

Figure 1 displays the accuracy and loss metrics for the initial model across training epochs. The top panel of the graph shows the training and testing accuracy, both of which increase over time, indicating that the model is learning effectively. The initial sharp increase followed by a plateau suggests that most learning occurs in the early stages. However, the test accuracy starts to flatten as the epochs increase, hinting

at potential overfitting or the model's limitations in generalizing beyond the training data.

The bottom panel shows the loss during training and testing. Both training and test loss decrease sharply in the initial epochs and then gradually plateau, mirroring the trends seen in the accuracy graph. The convergence of training and test loss, particularly after the initial drop, demonstrates the model's stability in learning without overfitting significantly.

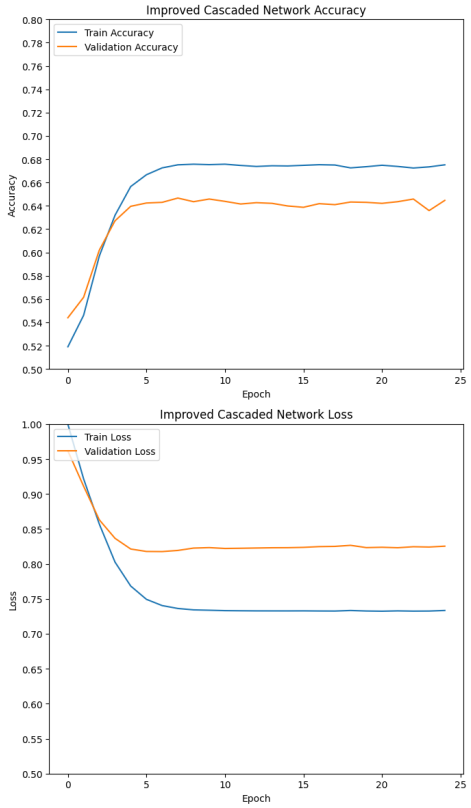


Figure 2. Model accuracy and loss of the Conv1D model over epochs.

The top panel of Figure 2 shows a rapid improvement in both training and validation accuracy right from the initial epochs, stabilizing at a high level early in the training process. This rapid convergence suggests that the Conv1D architecture is highly effective for this dataset, with its design potentially allowing for quicker and more robust feature extraction compared to the initial model.

The bottom panel depicts the training and validation loss, which shows a sharp decline in the initial epochs and then levels off.

It is important to note that the x-axis extends up to 25 epochs. This was done to reduce overfitting. Extending the training beyond certain epochs did not result in significant improvements, and instead, there was a tendency for the accuracy to plateau or decrease slightly.

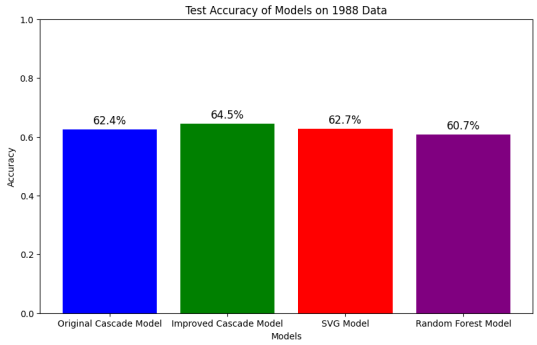


Figure 3. Comparison of model accuracy on original dataset

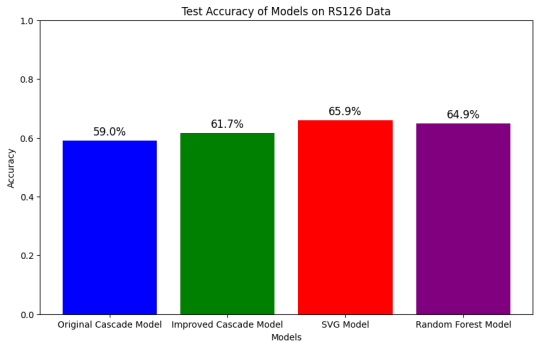


Figure 4. Comparison of model accuracy on RS126 dataset

Figure 3 and Figure 4 display the test accuracies of various computational models on two datasets. The Improved Cascade Model achieved the highest accuracy on the original 1988 dataset, while on the RS126 dataset, the SVM model showed the highest performance, followed closely by the Random Forest model. These results indicate that the cascading networks are effective for the 1988 dataset, but not as generalizable as the SVM and Random Forest model.

| Model | Overall | Beta-sheet | Alpha-helix | Coil |
|---|---|---|---|---|
| Initial Model | 62.4% | 31% | 42% | 84% |
| Conv1D Model | 64.5% | 29% | 34% | 88% |
| SVM Model | 62.7% | 28% | 38% | 88% |
| Random Forest | 60.7% | 17% | 23% | 94% |

TABLE 1. ACCURACY OF THE MODELS ON 1988 STUDY DATASET

| Model | Overall | Beta-sheet | Alpha-helix | Coil |
|---|---|---|---|---|
| Initial Model | 59.0% | 30% | 44% | 85% |
| Conv1D Model | 61.7% | 35% | 49% | 84% |
| SVM Model | 65.9% | 33% | 51% | 91% |
| Random Forest | 64.9% | 31% | 46% | 96% |

TABLE 2. ACCURACY RESULTS FOR THE RS126 DATASET.

The analysis of model performances as summarized in Table 1 indicates the prediction ability for each secondary structure. Each model is highly accurate at predicting coil structures, with the SVM and Random Forest models exhibiting the highest accuracies in this category. However, the Random Forest model performs notably worse in predicting other structural types, particularly beta-sheets and alpha-helices, which results in the lowest average accuracy for the 1988 dataset.

Following the performance overview on the 1988 dataset, Table 2 presents the accuracy metrics for the RS126 dataset. The results show a marked improvement in performance by the SVM and Random Forest models, particularly in their ability to predict more complex structures such as alpha-helices and coils, with the SVM model achieving the highest overall accuracy at 65.9%. This suggests that while the Random Forest model exhibited some limitations on the 1988 dataset, it adapts well to the RS126 dataset. This enhancement indicates the potential adaptability of the Random Forest model.

| Qian's Model | **64.3%** |
|---|---|
| **Model** | **Difference in Overall Accuracy** |
| Initial Model | -1.9% |
| Conv1D Model | +0.2% |
| SVM Model | -1.6% |
| Random Forest | -3.6% |

TABLE 3. DIFFERENCE IN OVERALL ACCURACY COMPARED TO ORIGINAL

## 5. Discussion

The initial reimplementation of Qian and Sejnowski's cascading network model resulted in a slight decrease in accuracy compared to the original, which could be attributed to variations in the optimization functions or other unspecified parameters in their study. In the paper, it is stated that they believe it would be difficult to improve from their 64.3% score, however,by introducing a modern Conv1D layer, the improved model was able to surpass their overall accuracy. The additional layer improvement, alongside the changes to accommodate it, indicate that a network previously devised can benefit from modern architectures. While the accuracy score from the improvement is small, it can be seen from the MCNN-GRNN network [5], that modern advances can significantly improve predictive capabilities beyond the existing frameworks.

Moreover, the superior performance of the SVM and Random Forest models on the RS126 dataset highlights the potential of these methods to adapt to different datasets, unlike the cascaded models. The improved cascading model has a larger gap between the training and test data than the original, indicating overfitting might be the reason for the lower accuracy on the RS126 dataset.

With the improved cascading model, it can be viewed from Figures 1 and 2, that while the introduction of the Conv1D layer did enhance performance, it also led to a larger gap between training and testing accuracies compared to the original model. This disparity points to potential overfitting, which could be attributed to the limited size of the training dataset used in the 1988 study.

In conclusion, this analysis has not only demonstrated the potential of integrating modern neural network architectures to improve traditional models but also highlighted the importance of dataset diversity. While the improvements with the modernized cascading model were evident, the problem of overfitting is still apparent, particularly with smaller amounts of training data.

## 6. Conclusion

This module served as my introduction to machine learning, and through this project, I began to see the practical applications of the theories covered in lectures. The project allowed me to investigate how neural networks function, with a specific focus on feedforward networks, backpropagation, and convolutional neural networks (CNNs). This

exploration required not only an understanding of how these networks operate but also my first use of the TensorFlow library to implement them effectively.

Additionally, I explored specific machine learning methods such as Support Vector Machines (SVM) and Random Forests. The results of the project clearly showed that both methods have their distinct benefits as well as limitations.

Becoming familiar with these models and techniques has greatly enhanced my understanding and marked a significant first step into the world of machine learning.

## 7. Future Works

Testing the current models on a wide variety of datasets, which vary not only in size but also in sequence complexity, would provide a stronger analysis of their performance across different biological contexts. This would help in understanding the generalizability and limitations of each model more accurately.

Due to previous courseworks requiring computational resources, there was limited opportunity to optimise hyperparameters through extensive methods like grid search. Future work should include a thorough exploration of hyperparameter optimization, which could potentially yield significantly better results.

Instead of making one improvement to the original cascading network, considering a complete overhaul of the network architecture could be beneficial. Introducing additional layers or experimenting with different types of layers might lead to a fundamentally different network, but it could achieve significant improvements in predictive performance.

Another method worth exploring is Large Language Models (LLMs) for protein structure prediction. LLMs have shown ongoing success in various fields of study, and their potential in computational biology is worth investigating.

## References


[1] N. Qian and T. J. Sejnowski, "Predicting the secondary structure of globular proteins using neural network models," 1988, received 25 September 1987, and in revised form 14 March 1988.

[2] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[3] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.

[4] J. Zhou and O. G. Troyanskaya, "Predicting effects of noncoding variants with deep learning-based sequence model," *Nature Methods*, vol. 12, no. 10, pp. 931–934, 2015.

[5] V. Bongirwar and A. S. Mokhade, "An improved multi-scale convolutional neural network with gated recurrent neural network model for protein secondary structure prediction," *Neural Computing and Applications*, 2024, received 24 August 2023; Accepted 15 April 2024; Published 13 May 2024. [Online]. Available: https://doi.org/10.1007/s00521-024-09822-8

[6] W. S. Noble, "What is a support vector machine?" *Nature Biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2006.

[7] C. H. Q. Ding and I. Dubchak, "Multi-class protein fold recognition using support vector machines and neural networks," *Bioinformatics*, vol. 17, no. 4, pp. 349–358, 2001.

[8] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[9] H.-B. Shen and K.-C. Chou, "Predicting protein subcellular localization by the ensemble of multiple predictors via protein-protein interaction network with edge clustering coefficients," *PLoS ONE*, vol. 3, no. 3, p. e1656, 2008.

[10] T. Hasan, "RS126Data - Protein Secondary Structure Prediction Dataset," https://www.kaggle.com/datasets/tamzidhasan/rs126data, 2022, accessed: 14/05/2024.