Karaelmas Science and Engineering Journal
Journal home page: https://dergipark.org.tr/tr/pub/karaelmasfen
DOI: 10.7212/karaelmasfen.1415248

Research Article
Received / Geliş tarihi : 05.01.2024
Accepted / Kabul tarihi : 01.05.2024

Siamese Neural Networks Based Ensemble Model for the Prediction of Protein-Protein Interactions

Protein-Protein Etkileşimlerinin Tahmini İçin Siyam Sinir Ağı Tabanlı Topluluk Modeli

Duygu Geçkin[1], Güleser Kalaycı Demir[2*]

[1]Dokuz Eylül University, The Graduate School of Natural and Applied Sciences, İzmir, Türkiye

[2]Dokuz Eylül University, Faculty of Engineering, Department of Electrical and Electronics Engineering, İzmir, Türkiye

Abstract

A wide range of biological processes, including signal transmission, immunological responses, and metabolic cycles, are impacted by protein-protein interactions. These interactions have enormous implications for figuring out the origins of diseases and creating treatments. However, experimental methods for identifying PPIs are resource-intensive, time-consuming, and have limited coverage. Thus, computational techniques are essential to help and enhance activities related to protein identification. This study aims to build a deep learning network for predicting protein-protein interactions using only sequence information. Three different encoding methods are used to encode protein sequences: Binary Encoding, Autocovariance, and Position Specific Scoring Matrix. In order to predict protein-protein interactions, a convolutional Siamese neural network is employed to find complex patterns between protein sequence pairs. This network consists of two identical subnetworks with matched parameters. When applied to the human dataset, the suggested technique shows strong prediction performance with an accuracy of 84.07%, sensitivity of 92.45%, and precision of 91.45% for the model using the PSSM protein representation approach. An ensemble approach is suggested to combine the outputs from these three encoders because it is known that different encoding techniques capture various aspects of the same protein sequence. The accuracy obtained increased to 86.27% for the ensemble approach on the test set, with a sensitivity of 93.07% and a precision of 92.15%. The outcome highlights the importance of integrating several encoding methods to benefit from their complementary features and raise the accuracy of protein-protein interaction prediction.

Keywords: Deep learning, one-hot encoding, position-specific scoring matrices, autocovariance

Öz

Sinyal iletimi, immünolojik yanıtlar ve metabolik döngüler dahil olmak üzere çok çeşitli biyolojik süreçler, protein-protein etkileşimlerinden etkilenir. Bu etkileşimlerin, hastalıkların kökeninin anlaşılması ve tedavilerin oluşturulması açısından çok büyük etkileri vardır. Ancak protein-protein etkileşimlerini belirlemeye yönelik deneysel yöntemler yoğun kaynak gerektirir, zaman alıcıdır ve kapsamı sınırlıdır. Bu nedenle, protein tanımlamayla ilgili faaliyetlere yardımcı olmak ve bunları geliştirmek için hesaplamalı teknikler önemlidir. Bu çalışma, yalnızca dizi bilgisini kullanarak protein-protein etkileşimlerini tahmin etmek için derin öğrenme ağı oluşturmayı amaçlamaktadır. Protein dizilerini kodlamak için üç farklı kodlama yöntemi kullanılmıştır: İkili Kodlama, Otokovaryans ve Konuma Özel Puanlama Matrisi. Protein-protein etkileşimlerini tahmin etmek amacıyla, protein dizi çiftleri arasındaki karmaşık modelleri bulmak için evrişimli bir Siyam sinir ağı kullanılmıştır. Bu ağ, eşleşen parametrelere sahip iki özdeş alt ağdan oluşmaktadır. Önerilen teknik, insan veri kümesine uygulandığında, PSSM protein temsili yaklaşımını kullanan model için %84.07 doğruluk, %92.45 hassasiyet ve %91.45 kesinlik ile güçlü tahmin performansı göstermektedir. Farklı kodlama tekniklerinin aynı protein dizisinin farklı yönlerini yakaladığı bilindiğinden bu üç kodlayıcıdan gelen çıktıları birleştirmek için bir topluluk yaklaşımı önerilmektedir. Test setinde topluluk yaklaşımı için elde edilen doğruluk %86.27'ye hassasiyet ve %93.07'ye kesinlik ise %92.15'e artırılmıştır. Sonuç, tamamlayıcı özelliklerinden yararlanmak ve protein-protein etkileşimi tahmininin doğruluğunu artırmak için çeşitli kodlama yöntemlerinin entegre edilmesinin önemini vurgulamaktadır.

Anahtar Kelimeler: Derin öğrenme, ikili kodlama, konuma özel puanlama matrisi, otokovaryans

*Corresponding author: guleser.kalayci@deu.edu.tr

Duygu Geçkin orcid.org/0000-0002-2257-5484
Güleser Kalaycı Demir orcid.org/0000-0003-3808-5305

1. Introduction

The prediction of protein interactions is crucial for studying diseases, cellular systems, and forming the foundation for therapeutic strategies (Browne et al. 2010). Protein-protein interactions (PPIs) play a pivotal role in many cellular biological processes, e.g., cellular organization, transmission of signals, recognition of foreign molecules, and acceleration of chemical reactions (Sun et al. 2017). Various experimental techniques have facilitated the exploration of conserved protein interaction sites and the screening of numerous protein interaction partners. Tandem affinity purification (TAP), nuclear magnetic resonance (NMR), atomic force microscopy (AFM), X-ray crystallography, and chemical crosslinking are among the techniques used (Zhu et al. 2019). While these biological experimental approaches have greatly helped to the identification of PPIs, it is critical to recognize that due to their labor-intensive, expensive and time consuming nature, they only cover a portion of the vast areas of PPIs.

As a result, there is a rising need for computational tools to complement and improve our understanding of protein interactions (Yang et al. 2021). Many researchers have actively pursued the creation of sequence-based approaches for finding novel PPIs. According to experimental findings, it is possible to predict PPIs by only using information from amino acid sequences (You et al. 2013). Shen et al. (2007) used protein information mining to compute the frequencies of conjoint triads inside protein sequences by treating three consecutive amino acids as a unit. Their studies demonstrated that PPIs could be predicted based on sequence information (Shen et al. 2007).

Deep neural networks, a major development in machine learning in recent years, have the capacity to learn efficient representations of raw data automatically. They are excellent at identifying high-level features, improving performance beyond what can be accomplished by conventional models, and also providing increased interpretability. So, in recent times, deep learning has demonstrated significant interest in domains such as computer vision, machine translation, and bioinformatics. Deep neural networks also help us better understand the information contained in biological data by giving us valuable insights into its underlying structure (Angermueller et al. 2016). For instance, it has been utilized for tasks like calling Single Nucleotide Polymorphisms (SNPs) and detecting small insertions and deletions (indels) (Poplin et al. 2018). Additionally, deep learning techniques have been employed to assess the impact of non-coding sequence variants on 3D chromatin structure (Trieu et al. 2020). Furthermore, deep learning plays a crucial role in predicting various aspects of proteins, including their function (Gligorijević et al. 2021), structural attributes (Jumper et al. 2021), and interactions with other proteins (Hashemifar et al. 2018).

A fundamental computational problem in predicting PPIs based on sequences is efficiently encoding the critical information inherent in PPIs. Shen et al. (2007), addressed this problem by using the conjoint triad technique, which allows for extracting characteristics from protein sequences based on the unique properties of amino acids. Scientists divided the 20 amino acids into seven groups to simplify the representation based on parameters such as dipoles and side chain volumes. This approach to categorization allows for a more efficient and informative description of the protein sequences (Shen et al. 2007).

Guo et al. (2008) generated feature vectors from protein sequences using the auto covariance (AC) approach. This approach takes into account surrounding effects, allowing it to reveal patterns that span whole sequences (Guo et al. 2008). Sun et al. (2017) developed a PPI prediction model that relies on sequence information, utilizing a stacked autoencoder. This deep learning approach is built upon the encoding-decoding process (Sun et al. 2017). Wang et al. (2019) proposed a deep neural network (DNN) model for predicting PPIs that included AC and conjoint triad (CT) descriptors (Wang et al. 2019). The feature vector space of an amino acid consists of AC and CT features. Thus, each pair of proteins is encoded with a vector. Gao et al. (2023) used an approach for feature extraction method that combined several techniques. The vectors obtained via pseudo amino acid composition (PseAAC), auto covariance descriptor (AC), pseudo position-specific scoring matrix (PsePSSM), encoding based on grouped weight (EBGW), multivariate mutual information (MMI), and conjoint triad (CT) are concatenated to create the fused feature representation. Convolution and pooling of the residual convolutional neural network can then be used to get high-level information. At last, to construct the EResCNN model, an ensemble of RCNN, XGBoost, random forest, LightGBM, and extremely randomized trees is used (Gao et al. 2023). Zhang et al. (2019) presented a deep learning-based strategy called EnsDNN inspired by Deep Neural Networks (DNNs) characteristics. Three descriptors the auto covariance descriptor (Wold et al. 1993), local descriptor (LD) (Tong and Tammi 2008), and multi-scale continuous and discontinuous local descriptor (MCD) (You et al. 2015) are originally used in the EnsDNN algorithm (Zhang et al. 2019).

Siamese neural networks (NNs) are effective in tasks requiring them to understand the dynamic interaction between two distinct variables properly. In order to handle the input protein pair for PPI prediction, several current PPI deep learning architectures have been implemented. Madan et al. (2022) developed a deep learning model that uses a Siamese neural network and the ProtBERT19 (Elnaggar et al. 2021) deep sequence embedding technique to predict PPIs using the primary sequences of protein pairs (Madan et al. 2022). Nourani et al. (2022) created a functional protein association network by integrating protein sequences during the embedding process. They presented TripletProt, a novel method for protein representation learning based on Siamese neural networks (Nourani et al. 2022). Özger and Çakabay (2023) used a Siamese neural network and Resnet50 to predict protein-protein interactions for SARS-CoV-2 for PSSM image datasets of different sizes. Their approach is similar to ours in that they consider PSSM matrices of proteins as grayscale images. Their findings showed that protein-protein interaction network prediction could potentially be successfully achieved by utilizing pictures produced from PSSM matrices (Özger and Çakabay 2023).

In this study, we aim to utilize Siamese Convolutional Neural Network to predict PPI solely based on amino acid sequences of proteins. These proteins are encoded using three distinct methods: Binary Encoding, position-specific scoring matrix (PSSM), and autocovariance. Furthermore, we have introduced an ensemble approach that combines the prediction results of these three encoding approaches. What sets our approach apart from existing ensemble encoding methods is that we do not concatenate different encodings into a vector. Instead, the network individually processes each encoded amino acid, after which a combination matrix is constructed to include the prediction results. The combination enhances the accuracy performance of the network due to its capacity to extract various feature information from interacting protein sequences using a range of descriptors. These descriptors work together to provide complementary feature information. One-hot encoding is a simple technique for encoding protein sequences that requires no prior knowledge and represents each amino acid separately (Richoux et al. 2019). PSSM, on the other hand, dives into evolutionary links by combining information from homologous sequences. Meanwhile, AC is concerned with capturing the physicochemical properties of amino acids, which are critical for understanding protein features. (ElAbd et al. 2020).

Also, we have integrated both a Siamese neural network and a multilayer feed-forward neural network (MLF-NN) to enhance the prediction performance of PPIs. Siamese Neural Networks are created specifically to learn a similarity score between pairs of data points. MLP is used to extract and abstract multi-level features from learned representations. When paired with Siamese networks, they help represent more sophisticated and nonlinear similarity patterns between input pairs.

The contributions of this study are as follows:

- Introduction of a convolutional Siamese neural network for predicting protein-protein interactions by employing three distinct protein sequence encoding methods.
- Development of an ensemble strategy aimed at enhancing the performance of a single predictor.

This paper is organized as follows: Section 2 presents our dataset and protein representation techniques and introduces the constructed Siamese convolutional neural network. The obtained results are given in Section 3. Finally, the conclusion and suggestions are presented in Section 4.

2. Material and Methods

This section elaborates on the proposed ensemble approach for predicting PPIs based on amino acid sequences. Our model consists of three steps: (1) Encoding the protein sequences into numeric values via binary encoding, PSSM, and AC descriptors, respectively. (2) Training these sequences individually using Convolutional Siamese Neural Networks. (3) Ensembling the prediction results and inputting them into MLF-NN. The flowchart of the proposed study is shown in Figure 1.

2.1. Dataset

We utilized the dataset offered by Richoux et al. (Richoux et al. 2019). A list of protein pairs known to interact was accessed via UniProt website. This query was conducted on June 18th, 2018, to gather all human protein sequences with evidence of interactions with other proteins. Employing Biopython, an internal Python script was developed to generate a negative dataset. This dataset consisted of randomly selected proteins, ensuring they did not exhibit any known interactions. Furthermore, sequences with more than 1,166 amino acids were not included in this set. Since the study by Nevers et al. (Nevers et al. 2023) indicates that only a small fraction of proteins exceed 1200 amino acids, 1,166 as the length of proteins appears reasonable and sufficient. Subsequently, this dataset was randomly divided into three distinct groups, each containing an equal number of positive and negative samples, making up the hold-out test set, the hold-out validation set, and the training set. For training purposes, the network was trained using Richoux medium train dataset, encompassing 26,303 protein pairs known to interact and an equivalent number of non-interacting protein pairs (Richoux et al. 2019).

We observed that Richoux test protein pairs were absent from the training set, but their mirrored counterparts were present. To avoid potential overfitting and misleading evaluation results, we excluded the protein pairs with mirror counterparts from the test dataset. In this study, a total of 56,674 human protein interactions were utilized, with 52,606 interactions used for training our network model. The testing phase incorporated a dataset comprising 797 interactions classified as interacted and 3,271 as non-interacted, resulting in a total of 4,068 interactions used for testing the model's performance.

2.2. Protein sequence Representation

Effective computational identification of PPIs depends on a carefully thought-out strategy based on protein sequences. This usually involves two main steps.

First, a feature extraction technique needs to be developed. The key characteristics of the protein sequence's inherent important properties as well as the substance of data on protein-protein interactions must be captured by this method.

Second, choosing and creating an effective prediction classifier assumes critical significance. This decision needs to be carefully thought out because how well it fits the particulars of the work will have a big impact on predicting PPIs.

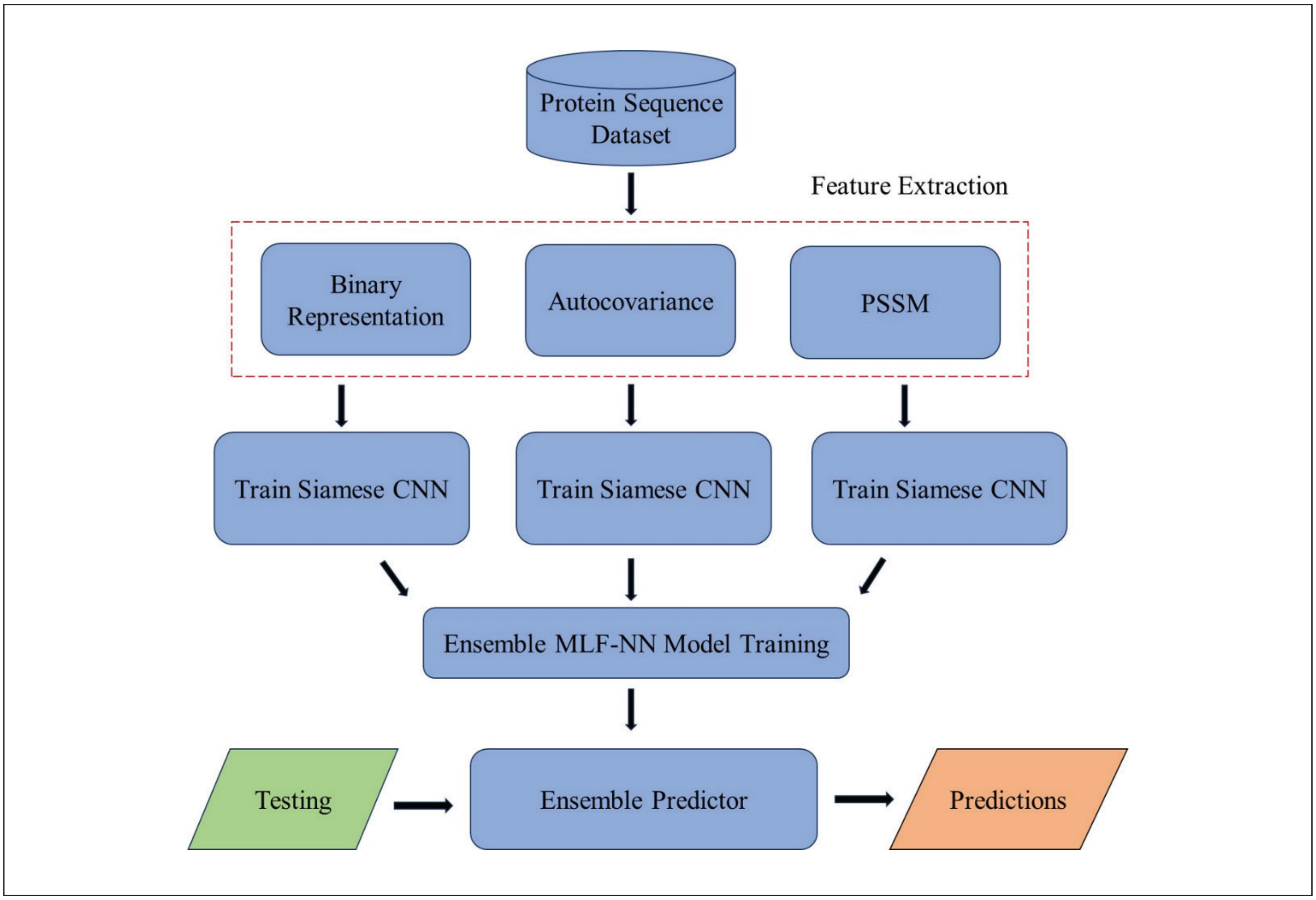

Figure 1. The flowchart of the proposed method**.**

2.2.1. Binary Representation

In this method, individual amino acids are represented by binary vectors. The one-hot encoding, often known as orthogonal encoding, is the most well-known type of binary encoding. Each location in the vector corresponds to a different amino acid using this encoding technique (Richoux et al. 2019). Notably, the location that corresponds to the amino acid found in the sequence is given a value of 1, while every other position is given a value of 0. So, one-hot encoding creates sparse matrices with a large percentage of zero values. When working with huge datasets, this sparsity is favorable for memory use and processing performance.

The datasets we work with encompass additional elements beyond the standard set of 20 proteinogenic amino acids. These additional elements include selenocysteine (U), a placeholder for either asparagine or aspartic acid (B), another placeholder for either glutamic acid or glutamine (Z), and a placeholder for unidentified amino acids (X). Therefore, in the context of a protein sequence, each individual amino acid is depicted using a binary vector that is 24 units in length. In order to ensure uniformity and match the required sequence matrix length, zeros are introduced to the one-hot encoded representation. As a consequence, we used a technique of padding sequences with zeros in order to deal with the problem of different length sequences. This step is crucial because the supported maximum sequence length is 1,166. The representation of amino acids in a protein sequence using the one-hot encoding technique is shown in Figure 2.

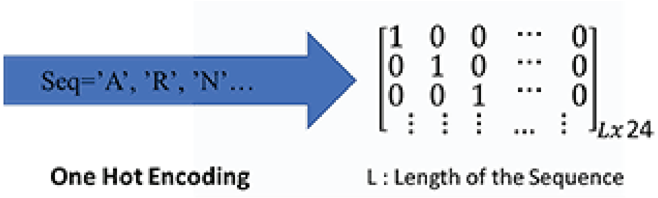

Figure 2: Example of binary representation of amino acids in a protein sequence.

2.2.2. Autocovariance (AC)

AC, a statistical tool, is used to transform amino acid sequences into uniform matrices and takes into account the interactions between amino acids at specific points in a protein sequence (Li and Chen 2013).

Firstly, amino acid sequences are captured by seven different physicochemical attributes. These characteristics include the following for amino acids: solvent-accessible surface area (SASA), polarity (P1), polarizability (P2), hydrophobicity (H), side chain volumes (VSC), and the net charge index of side chains (NCISC) (Fauchère et al.1988).

The prediction of PPIs is based on these characteristics. This entails converting the physical qualities of amino acid residues into numerical values and then normalizing these values to have a mean of zero and a standard deviation according to Equation (1):

$$P_{ij}' = \frac{Pi,j - Pj}{Sj} \quad (i = 1, 2, \ldots 20;\ j = 1, 2, \ldots, 7) \qquad (1)$$

where $P_{i,j}$ is the *j-th* physicochemical property value for the *i-th* amino acid, P_j is the mean of the *j-th* physicochemical property over 20 amino acids, and S_j is the corresponding standard deviation of the *j-th* physicochemical property. Each protein sequence is transformed into seven vectors, representing each amino acid by normalized values. To represent a protein sequence X with length L, the AC variables are computed as follows:

$$AC_{lag,j=\frac{1}{n-lag}} \sum_{i=1}^{n-lag} \vdots\vdots\vdots (X_{i,j} - \frac{1}{n}\sum_{i=1}^{n} \vdots\vdots\vdots X_{i,j}) \\ (X_{(i+lag)j} - \frac{1}{n}\sum_{i=1}^{n} \vdots\vdots\vdots X_{i,j}) \qquad (2)$$

The distance or gap between two successive amino acid residues—where one amino acid is regarded as the neighbor of the other—is indicated by the lag value. In this instance, "j" stands for a particular descriptor or characteristic, "i" stands for the location of the amino acid residue in the sequence designated as "X," and "n" stands for the overall length or number of amino acids in sequence "X".

2.2.3. Position Specific Scoring Matrix (PSSM)

The conservation of amino acid residues at particular places within protein chains is reflected by the PSSM (Wang et al. 2017). It is built utilizing evolutionary data and based on feature extraction methods that have proven successful in several bioinformatics applications. These fields involve predicting the secondary structure of proteins, detecting proteins that bind DNA, and forecasting PPIs (Zahiri et al. 2013).

The structure of a PSSM is an Lx20 matrix, where L represents the length of the protein sequence, while natural amino acids are represented by 20. Significantly conserved locations are given higher ratings, while places with little or no conservation are given scores that are close to zero (Gao et al. 2016). PSSM elements are calculated as in Equation (3):

$$PSSM(i,j) = \sum_{k=1}^{20} ::: w(i,k) \times Y(j,k) \quad (i = 1,\ldots,L; 1,2,\ldots,20) \qquad (3)$$

The expression w(i,k) represents the frequency of the i-th amino acid (out of a set of 20 amino acids) occurring at position i within a collection of functionally related, already aligned protein sequences. Meanwhile, Y(j,k) is a substitution matrix that reflects the values found in Dayhoff's mutation matrix for transitions between the j-th and k-th amino acids. In simpler terms, Y(j,k) signifies how quickly one character in a protein sequence changes to another character over time.

In this study, we calculate PSSM by running three rounds of PSI-BLAST with an E-value of 0.001 against the SwissProt database at the NCBI for a specific protein (Altschul et al. 1997). Finally, encoded sequences are fixed at 1,166 by padding with zeros to ensure uniformity in length.

2.2.4. Model Construction

Siamese CNN (Convolutional Neural Network) architecture, which comprises of two identical subnetworks that share the same structure, variables, and weights, is the architecture used by the implemented framework.

Siamese neural network, sometimes referred to as the twin neural network, is used for determining how similar or dissimilar two inputs are. Each input is processed by a separate subnetwork in a normal Siamese neural network, which is typically made up of one or more layers of neurons. The weights of these subnetworks are shared, distinguishing it from past systems and allowing for simultaneous changes of the parameters in both networks. The network can efficiently learn and gauge the similarity between pairs of inputs because of this shared-weight structure (Chen et al. 2022). To ensure uniformity and consistency, a Siamese design was used to grasp the complex interactions between two proteins. Therefore, each pair of proteins was handled simultaneously through a single network with weight sharing as opposed to using two separate models with different parameters. Our neural network structure delivered a unified representation, enabling a more accurate analysis of the protein connection.

As shown in Figure 3, our Siamese CNN architecture is made up of three main parts: a profile module that uses encoded protein sequences, a convolutional module, and a prediction module. Each encoded sequence is fed into two shared-weight subnetworks in order to determine how similar or dissimilar the two encoded sequences are. A thorough representation of the differences between the sequences is then produced by subtracting and combining the output feature vectors from each subnetwork. The resulting information is then placed through a fully connected operation, producing a single output that captures the distinct qualities and differences of the sequences.

The differences between the sequences are then thoroughly represented by subtracting and combining the feature vectors generated by each subnetwork. A fully-connected operation is then used to handle the collected data, producing a single output that contains the distinctive qualities and variations between the sequences. The input is multiplied by a weight matrix and a bias vector in the fully-connected stage. The output is then converted into a probability that ranges from 0 to 1 by means of the application of a sigmoid function. This

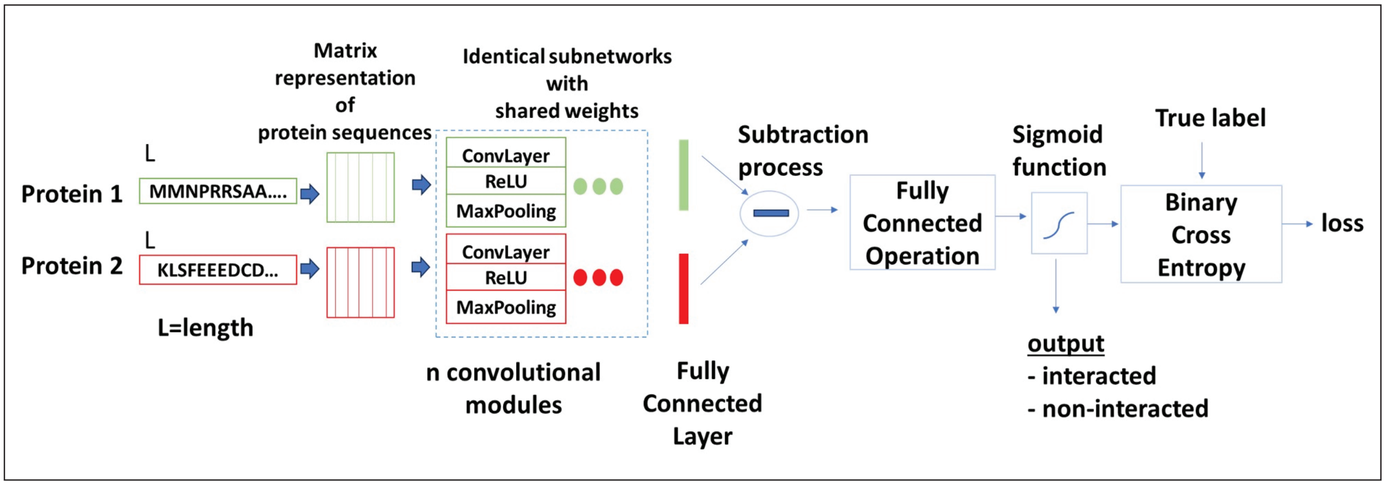

Figure 3. Convolutional Siamese neural network framework.

probability indicates the degree of similarity or dissimilarity between the two protein sequences.

The network updates itself throughout the training phase by minimizing the binary cross-entropy loss. This loss metric calculates the variance between the predicted labels and the actual labels, which helps the network perform better over time.

3. Results and Discussion

3.1. Evaluation Metrics

The network's performance and its predictive capabilities are determined by taking into account the calculation of the Matthews Correlation Coefficient (MCC), F-Score (F1), Specificity (SPE), Precision (PRE), Sensitivity (SE) or Recall, and Overall Prediction Accuracy (ACC). SPE assesses the true negative (TN) rate, whereas ACC indicates the percentage of events that were accurately predicted. While SE (or recall) evaluates a true positive rate (TP), PRE measures the accuracy of positive predictions. Additionally, although F1 indicates the harmonic mean of precision and recall, MCC offers a balanced metric that takes into account both false positives (FP) and false negatives (FN). Their definitions are outlined as follows:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

$$SPE = \frac{TN}{TN + FP} \quad (5)$$

$$PRE = \frac{TP}{TP + FP} \quad (6)$$

$$SE = \frac{TP}{TP + FN} \quad (7)$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN) \times (TN + FP) \times (TP + FP) \times (TN + FN)}} \quad (8)$$

$$F - Score\,(F1) = \frac{2 \times (PRE \times SE)}{(PRE + SE)} \quad (9)$$

3.2. Performance of the Proposed Method

Three types of feature vectors extracted by binary encoding, AC or PSSM, are separately used as the inputs for Siamese CNN with different configurations. These neural networks have varying learning rates, batch sizes, number of the convolutional filters, and filter sizes. Because the Siamese network demonstrates higher accuracy when utilizing the Adam optimization, we employ this algorithm for updating the network's parameters (Alkhalid 2022). To evaluate the performance of features generated through the application of the Binary Encoding method for protein representation, we conduct training and testing using 52,606 and 4,068 pairs of proteins, respectively. We combine different convolutional layer configurations and hyperparameters for the constructed layers to obtain maximum accuracies. Each convolutional module consists of three layers: 2D convolution, rectified linear unit (ReLU), and 2D pooling. After the input layer, we define a sequence of convolutional layers. Upon defining the final convolutional layer, we introduce a fully-connected layer and flatten it into a single vector.

In our experimental configuration, we tested various learning rates, including 1e-03, 5e-03, 1e-04, 5e-04, 1e-05, and 6e-05. Furthermore, we explored different training batch sizes, specifically 10, 16, 20, 32, 50, and 64. The number of neurons in the fully-connected layer ranged from 100 to 1500, and the iteration count varied from 10,000 to 50,000.

3.2.1. Experiments on Binary Encoding

The initial network includes four layers, combining 2D convolution and ReLU activation, along with three max-pooling layers. The first convolutional layer utilizes a 3x3 filter with 32 filters, maintaining spatial dimensions through the 'same' padding parameter. The layer's weights and biases are initialized using the 'narrow-normal' method, and a ReLU layer follows to introduce non-linear characteristics. Next, we utilize max pooling with a 2x2 pooling window and a stride of 2 to reduce the spatial dimensions of the feature maps while preserving essential features. After adding convolutional filters in a similar structure, a fully connected layer with 100 neurons is employed to capture high-level representations. We initialized the weights and biases of the fully connected layer using the 'narrow-normal' method. We also adjusted the learning rate, which ranged from 1e-03 to 6e-05.

In the first and second series of experiments, we kept the filter size constant at (3x3, 5x5) while varying the number of filters in each convolutional layer from 16 to 128. The best results achieved were 76.60% and 78.66%, respectively. The maximum values for precision (PRE), specificity (SPE), sensitivity (SE), F-Score, and MCC can be found in Table 1. In the third set of experiments, we used non-square filters (5x3, 7x3, and 9x3) for the four 2D convolutional layers, with filter sizes of 32, 64, and 128. The results showed that non-square filter sizes outperformed square ones, achieving accuracy as high as 80.24%. In the final series of experiments, we aimed to assess accuracy by using a combination of square and non-square filters. Unfortunately, the results showed a decrease in the maximum accuracy, which reached 77.67%.

Based on the experimental results, we identified the optimal hyperparameter set, as indicated in Table 2.

3.2.2. Experiments on AC

We created the AC dataset by removing uncommon amino acids Z, U, B, and X, resulting in 26,289 interacted and 26,206 non-interacted protein pairs for training. The test set already excludes uncommon amino acids, allowing it to be used for testing. The lag value was established at 15, and we employed seven physicochemical properties. As a result, the image input size was set to 15x7. The convolutional layer included two 2D-convolutional layers with filter sizes of 3x3, featuring 128 and 64 filters, respectively. These layers were then followed by ReLU activations and 2D-max-pooling with a pooling size and stride set to 2.

We conducted a set of experiments to fine-tune the size of the fully-connected layer. During each epoch, the network was trained with 1,000 inputs, and the total number of iterations was fixed at 10,000. Initially, the number of neurons in the fully-connected layer was adjusted, ranging from 100 to 2,000. Starting with 100 neurons, we achieved an accuracy of 62.2%. Then, increasing the number to 500 led to a substantial improvement, reaching 71.46% accuracy. Subsequently, we experimented with 1,000, 1,200, 1,400, 1,500, and 2,000 neurons to attain the highest accuracy. The obtained accuracy values, as shown in Figure 4, were 77.63%, 77.44%, 78.13%, 77.04%, and 77.41%.

It is concluded that the prediction accuracy significantly improves with the number of neurons, reaching its peak accuracy at a specific value.

With the fully-connected layer set at 1400, we aimed to investigate the impact of layers while keeping the iteration number, learning rate, and training size constant at 15,000, 6e-04, and 10,000, respectively. Initially, the convolutional layer consisted of two 2D-convolutional layers (3x3) with 128 and 64 filters, followed by ReLU and 2D-max-pooling with a size and stride of 2. The average prediction accuracy reached 76.33%. Next, we increased the number of 2D-convolutional layers by doubling the second layer to 128 filters, resulting in an accuracy increase to 77.46%.

In the third experiment, additional max-pooling and convolution layers were introduced into the network. The added max-pooling layer had a size and stride of 2, while the

Table 1. Assessing the performance of the proposed network under various hyperparameter sets for binary encoding.

Hyper-parameters Set Number	ACC (%)	PRE (%)	SPE (%)	SE (%)	FScore (%)	MCC (%)
Set 1	76.6	85.07	87.82	65.93	74.29	55
Set 2	78.67	84.47	88.27	68.37	75.57	58
Set 3	**80.2**	84.89	86.5	**74.04**	**79.1**	**61**
Set 4	77.67	**87.44**	**90**	65.97	75.2	57

Table 2. Siamese CNN model optimal hyper-parameters and activations for binary encoding.

Layer	Hyper-parameters	Activations
Input	Input Size=1,166x24	1,166×24
Convolution 2D	Filters=64, Kernel size=[7 3] Stride=1, padding=same Activation=ReLU	1,166×24×64
MaxPooling 2D	Pool size=[2 2], Stride=2, padding=[0 0 0 0	583×12×64
Convolution 2D	Filters=64, Kernel size=[7 3] Stride=1, padding=same Activation=ReLU	583×12×64
MaxPooling 2D	Pool size=[2 2], Stride=2, padding=[0 0 0 0]	291×6×64
Convolution 2D	Filters=64, Kernel size=[7 3] Stride=1, padding=same Activation=ReLU	291×6×64
Fully Connected	Activation=sigmoid	250

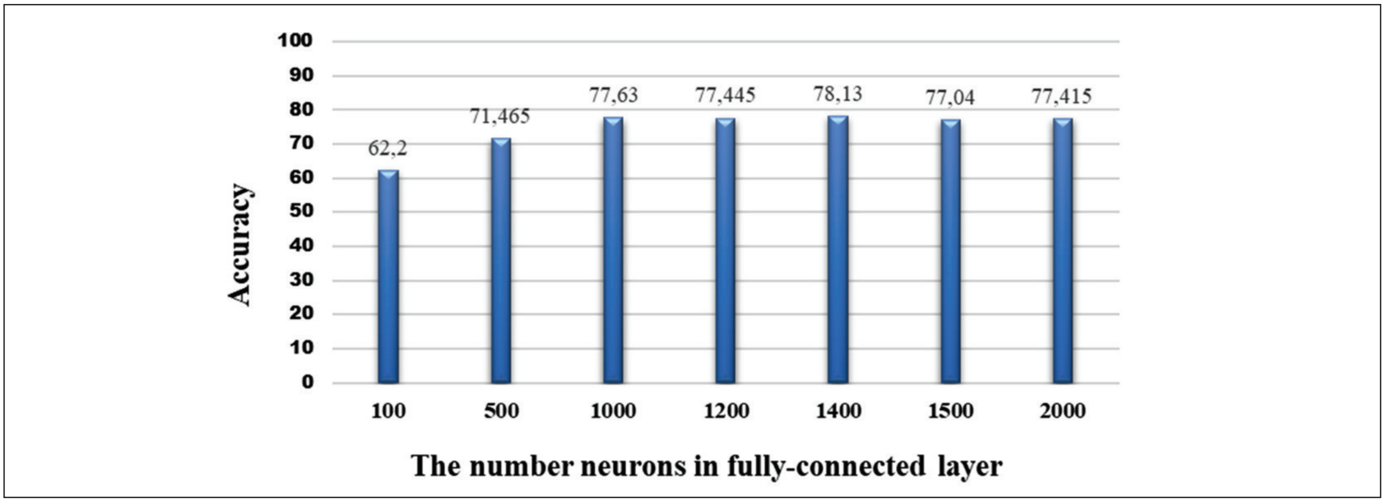

Figure 4. The effect of the neuron count in the fully-connected layer on accuracy.

Table 3. The impact of different layer configurations on the accuracy of AC.

Hyper-parameters Set Number	ACC (%)	PRE (%)	SPE (%)	SE (%)	FScore (%)	MCC (%)
Set 1	76.33	**87.52**	**90.76**	62.43	72.88	55
Set 2	77.47	85.05	88.46	66.35	74.55	56
Set 3	76.46	82.2	84.93	69.6	75.38	55
Set 4	**78.6**	85.47	86.94	**70.9**	**77.51**	**58**
Set 5	77.33	84.48	88.95	64.82	73.33	56

convolution layer consisted of 64 kernels (3x3). However, the use of the second max-pooling layer resulted in information loss, causing a decrease in prediction accuracy to 76.46%. Subsequently, the second max-pooling layer was removed, and the third convolution layer was increased to 128 kernels, leading to an accuracy increase to 78.60%.

Finally, when the number of filters in the convolutional layers was increased to 256, there was a slight decrease in accuracy, reaching 77.33%. Table 3 summarizes the network's performance across different layer configurations. The highest accuracy was achieved with a network featuring three convolution layers, each with a size of 3x3 and 128 neurons. In this architecture, the first convolution layer is followed by a max pooling layer for spatial downsampling. However, the second and third convolution layers skip max pooling to preserve detailed spatial information, enabling the capture of finer features and patterns. This configuration resulted in an accuracy of 78.60%.

3.2.3. Experiments on PSSM

Protein sequences were encoded into 1,166x20 matrices using the PSSM method with the dataset that does not contain uncommon amino acids. We tested various learning rates, including 1e-03, 5e-03, 1e-04, 5e-04, 1e-05, and 6e-05. Additionally, we explored different training batch sizes: 10, 16, 20, 32, 50, and 64. The fully-connected layer had a range of neurons from 100 to 1500, and the iteration number varied between 10,000 and 50,000.

The impact of four layer configurations on the accuracy of PSSM is shown in Table 4. In the first group, the architecture featured four convolutional layers. The first and second convolutions had a filter size of 7x3 with 128 and 64 filters, while the third and fourth convolutions used a filter size of 5x3 with 64 filters each. The design included two max-pooling layers and four ReLU layers. With a learning rate of 6e-05 and 1,400 neurons in the fully connected layer, an accuracy of 83.00% was achieved. This configuration allowed the network to capture both large-scale and smallscale features in the input data. The use of different filter sizes in the convolutional layers enabled the extraction of various levels of visual information, contributing to the improvement of accuracy. Furthermore, by fine-tuning the number of neurons to 32 in the fully connected layer, the model's performance was further optimized, leading to an increase in accuracy.

In the second group of convolutional layers, we examined the impact of using smaller 3x3 filter sizes on accuracy. The first and second convolutions used 3x3 filters, with 128 and 64 filters, respectively. These layers were designed to capture higher-level features by building upon the low-level features learned in the previous group of convolutional layers. The larger number of filters in the first convolutional layer suggested its role in learning complex, high-level features.

The third convolution also used a 3x3 filter size, but with 64 filters, aiming to capture higher level features while mitigating overfitting risks and reducing dimensionality by reducing the number of output feature map.

In the third experimental group, we studied the influence of mixing square and non-square filters on network performance. We experimented with combinations like 3x3 and 5x3, 3x3 and 7x3, and 5x5 and 7x3 filters using optimal parameters. The highest accuracy achieved in this experiment reached 81.93%. After systematic adjustment of hyperparameters and evaluation, we identified the optimal settings that led to a significant accuracy boost, reaching 84.07%.

We train the network using three different protein representation techniques within the range of 5,000 to 50,000 iterations. With the validation set, prediction error was calculated for each different iteration number. The training and validation losses dropped synchronously. The number of iterations was set at 10,000 for binary, 15,000 for autocovariance, and 50,000 for PSSM to reduce computing costs, as there was no substantial drop in loss after a certain number of iterations, or it remained constant. Based on the figures, the loss decreases over iterations and becomes stable, showing that the network has successfully learned the underlying patterns in the data for AC, as shown in Figure 5, Binary encoding, as shown in Figure 6, and the PSSM as shown in Figure 7 protein representation methods.

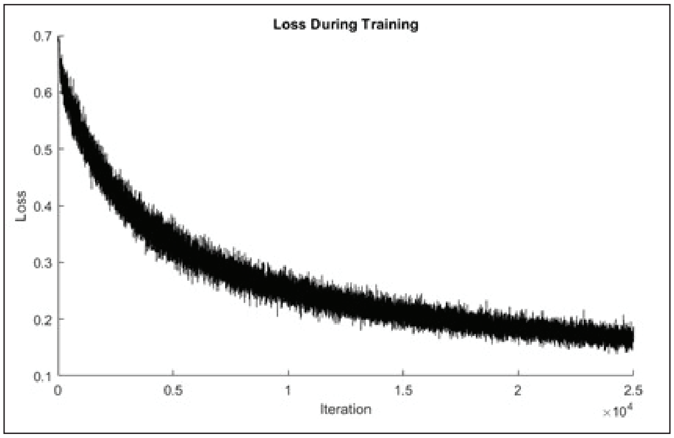

Figure 5. Network loss convergence for AC.

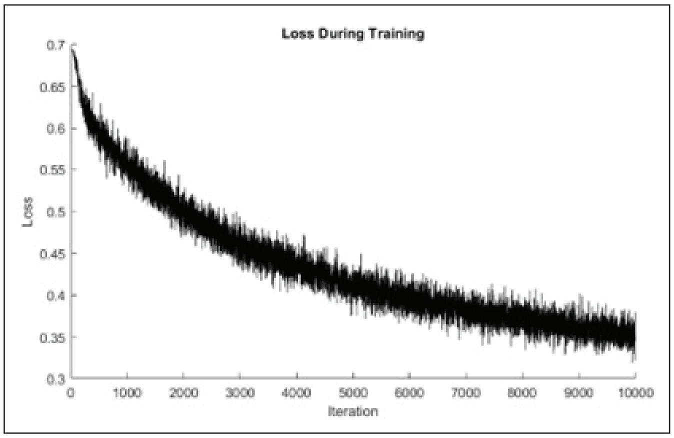

Figure 6. Network loss convergence for binary encoding.

Table 4. The impact of different layer configurations on the accuracy of PSSM.

Hyper-parameters Set Number	ACC (%)	PRE (%)	SPE (%)	SE (%)	FScore (%)	MCC (%)
Set 1	83	89.11	90.34	75.95	82	67
Set 2	83.73	90.39	92.08	73.51	81.08	67
Set 3	**84.07**	**91.45**	**92.45**	**76.17**	**83.11**	**69**
Set 4	81.93	87.05	89.21	74.46	80.26	64

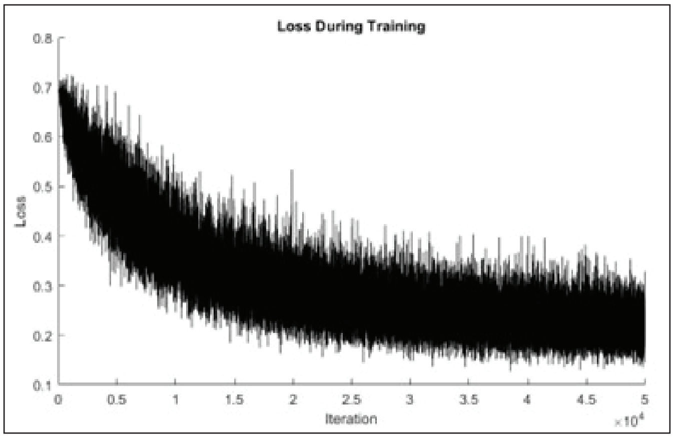

Figure 7. Network loss convergence for PSSM.

3.2.4. Performance Comparison of Three Encoding Methods

Table 5 compares the performance of three encoding methods with the mirror copies extracted from a dataset of Richoux et al. (2019). The graph shows how different encoding approaches affect the performance and efficacy of our deep learning model. Precision and sensitivity levels for the PSSM approach remain constant at 91.45% and 92.45%, respectively. As a result, protein representation approaches enhance network performance on both positive and negative datasets.

The Binary Encoding method displayed promising outcomes and yielded an average prediction accuracy (ACC) of 80.2%, precision (PRE) of 84.86%, specificity (SPE) of 86.5%, sensitivity (SE) of 74.04%, F-Score of 79.1%, and Matthews Correlation Coefficient (MCC) of 61%. This approach offers a notable advantage in terms of fast protein encoding when compared to the AC and PSSM methods. It achieves this through the use of one-hot encoding, which generates multi-dimensional and sparse vector representations. This efficient encoding allows us to process and analyze protein sequences faster, making it particularly advantageous in terms of computational speed.

In contrast, the AC method achieved an average ACC of 78.6%, PRE of 85.47%, SPE of 86.94%, SE of 70.09%, F-Score of 77.51%, and MCC of 58%. When encoding protein sequences, the AC approach integrates the neighboring effect. However, the results indicate that the AC approach has slightly lower accuracy than both the PSSM and Binary Encoding methods. However, when compared to the other two protein sequence encoding methods, the AC method excels in training speed because of its smaller input size. This advantage in training speed reduces the computing time required for model training, ultimately speeding up the PPI prediction process (Jia et al. 2020).

The network became enhanced outcomes while using the PSSM encoding method with an average prediction accuracy (ACC) of 84.07%, precision (PRE) of 91.45%, specificity (SPE) of 92.45%, sensitivity (SE) of 76.17%, F-Score of 83.11%, and Matthews Correlation Coefficient (MCC) of 69%.

This significant performance boost could be due to the fact that PSSMs contain information gained from the evolutionary history of proteins. This evolutionary information has been shown to have more predictive value for PPIs than the other two sequence-based protein encoding approaches.

The reason position-dependent approaches perform well is their effectiveness in gathering homologous data, providing crucial insights into protein evolution. In contrast, position-independent techniques excel at revealing the intrinsic properties of amino acids, allowing us to gain a better understanding of their fundamental characteristics. Our experimental results prove that the PSSM encoding captures the evolutionary relationships between proteins, and a Siamese Neural Network is able to detect this homology between two proteins through their PSSM matrices.

Table 5. Performance comparison of three encoding methods.

Protein Encoding Method	Binary Encoding	AC	PSSM
ACC (%)	80.2	78.6	**84.07**
PRE (%)	84.89	85.47	**91.45**
SPE (%)	86.5	86.94	**92.45**
SE (%)	74.04	70.9	**92.45**
F-Score (%)	79.1	77.51	**83.11**
MCC (%)	61	58	**69**

3.3. Performance of the Ensemble Encoding Siamese Model

Many sequence-based feature extraction techniques used in early studies mostly focused on a single-feature strategy. A protein sequence has a multitude of information about critical features, but this approach had drawbacks because it couldn't effectively integrate that data. The interrelationships between various elements within the sequence were also not fully taken into account. As a result, there has been a lot of interest in creating a revolutionary multi-feature fusion encoding technique. Building on this foundation, we present a deep ensemble learning strategy for protein interaction prediction. By combining the strengths of many methodologies, this method provides an appropriate means of complete learning. Protein sequences were encoded using three alternative representations in this approach, capturing the specific characteristics of each protein within the protein interaction network.

To extract significant feature data from amino acid sequences, we used three different encoding methods in our study: Binary Encoding, AC, and PSSM. Furthermore, we used convolutional Siamese neural networks to extract protein predictions from individual encoders, which were then merged as features into a multilayer feed-forward neural network. The network's top-performing weights, biases, and parameter values have been preserved. Following that, we acquired model outputs as floating-point values ranging from 0 to 1. The first model output corresponds to binary prediction values, the second output to AC prediction values, and the third output to PSSM prediction values for 50,000 protein pairs. As a result, all our model outputs form a matrix of size 50,000×3, which serves as input to the feed-forward neural network structure. In addition, we kept the corresponding labels, which are linked to the 50,000 protein pairs and stored in a 50,000×1 matrix. We configured the transfer functions for the appropriate levels to define the desired activation functions inside the network. We used a 'hardlim' transfer function in the first layer, which represents the hidden layer. Following that, in the second layer, which includes the output layer, we used a 'purelin' transfer function, which corresponds to a linear activation function. To improve the prediction performance of PPIs based on primary sequences, we combined a Siamese neural network with a multilayer feed-forward neural network (MLF-NN). The conceptual model framework is visually represented in Figure 8.

The MLF-NN structure used consists of a single hidden layer with neurons ranging in number from 2 to 16. Various backpropagation methods were tested during the network training phase. Three specific training algorithms produced the most accurate results: Levenberg-Marquardt, Gradient Descent with Momentum, and Fletcher-Powell Conjugate Gradient. These algorithms consistently outperformed in terms of reaching the highest levels of accuracy throughout training. Figure 9 depicts the network's performance with these transfer functions and various number of hidden layers.

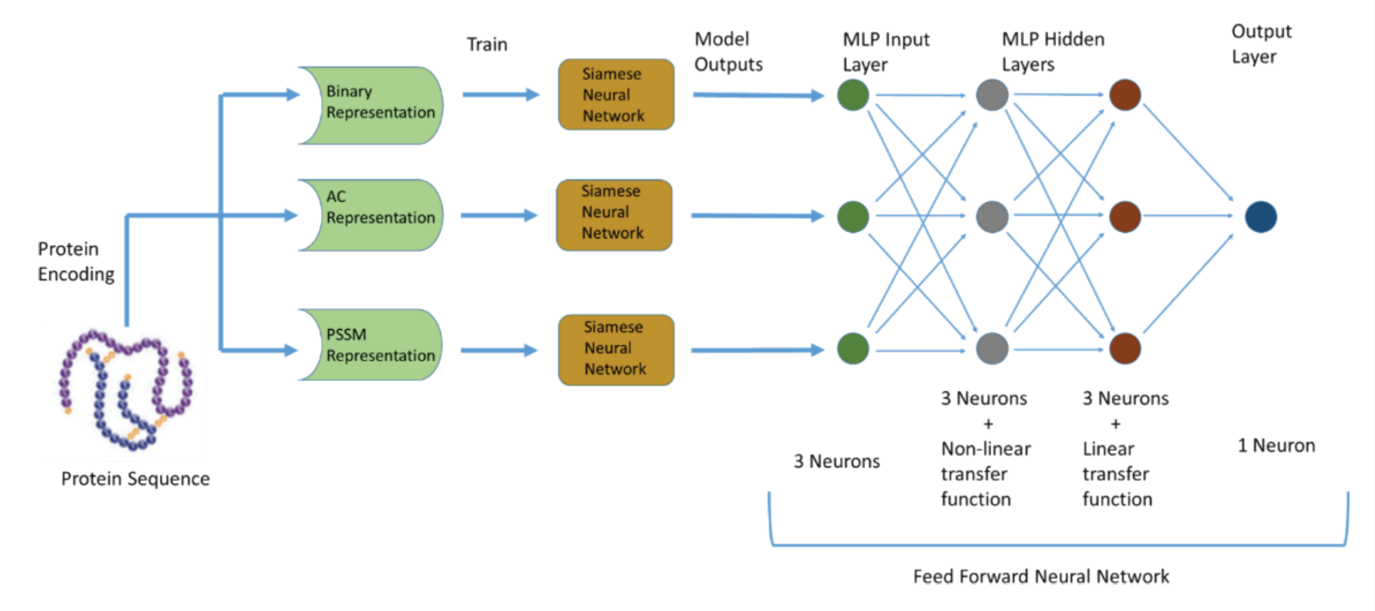

Figure 8. The framework for the proposed ensemble model.

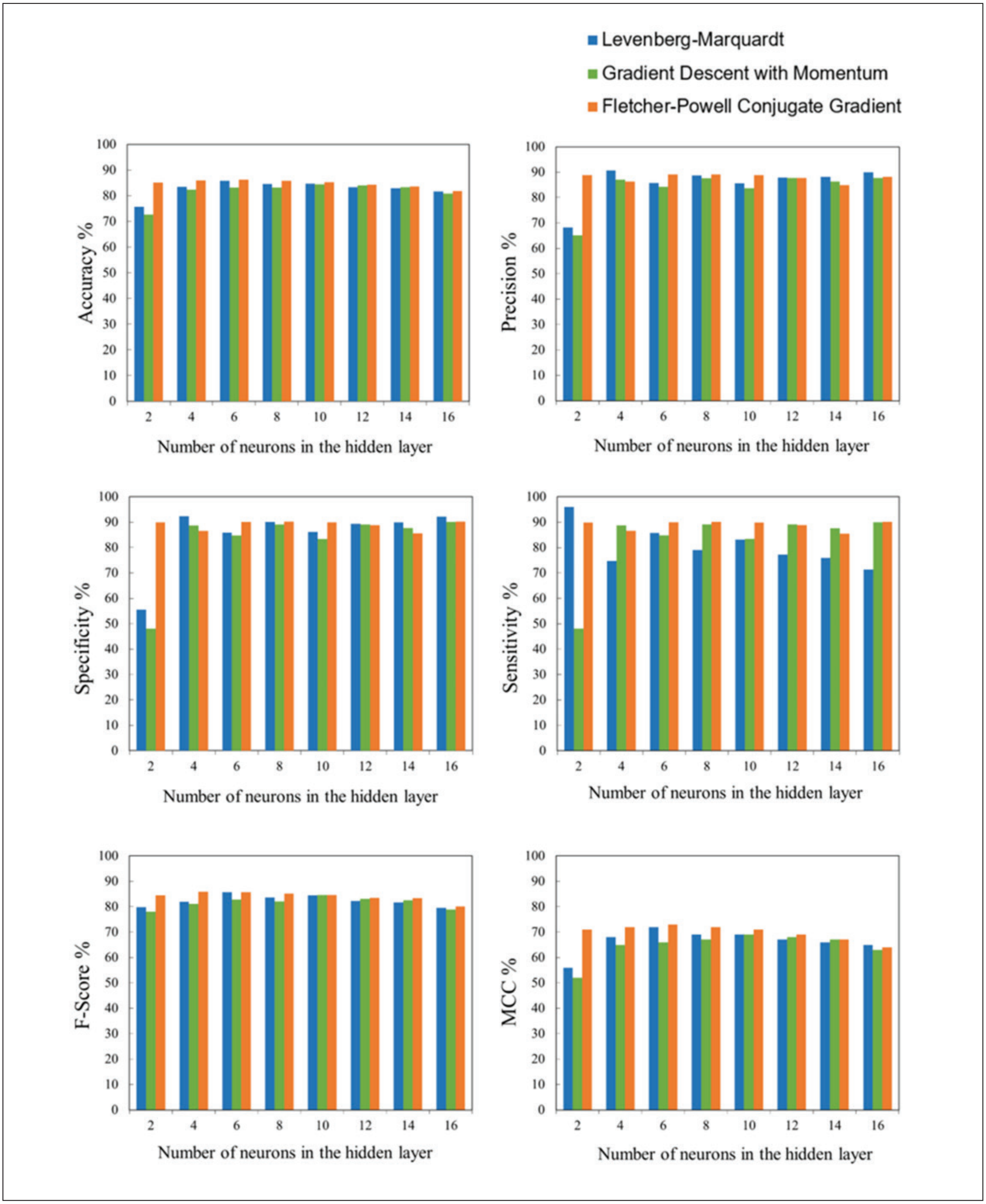

Figure 9. Ensemble network results with varying numbers of hidden neurons.

When the hidden layer was set to 6 neurons, the Fletcher-Powell Conjugate Gradient backpropagation algorithm attained its highest accuracy. This result emphasizes the significance of selecting the proper network configuration for improving performance. Furthermore, the algorithm's efficiency in terms of faster convergence contributes to its attraction for training neural networks. The algorithm's capacity to explore conjugate directions contributes to its efficiency, which greatly cut training time.

Among the individual encoding strategies, PSSM achieved the highest accuracy at 84.07%, which was further improved to 86.27% with an increase of 2.62% using the ensemble strategy.

3.3.1. Comparative Analysis with Richoux's Dataset

We conducted a thorough evaluation of our proposed method through a comparative study. This involved contrasting our results with those of Richoux and colleagues, who employed the same encoding methodology and made their data accessible Richoux et al. employed a fully connected deep learning model that used binary encoding for protein pairs to PPIs and these results confirmed the model's efficacy.

Notably, we utilized the same feature extraction method, namely Binary Encoding, to assess our network's performance on the identical dataset.

We conducted a comparative analysis between our convolutional Siamese neural network model and Richoux's fully connected model on the strict dataset, and we observed significant increases in prediction performance metrics. The strict dataset was designed to include protein pairs in which each protein appeared at most twice in the entire dataset. By imposing this constraint, the strict dataset ensures a more balanced representation of proteins, reducing the risk of the model becoming overly reliant on specific individual proteins.

Table 6 provides a comparison between our convolutional Siamese neural network model and Richoux's model on the strict dataset. Richoux's model achieved an accuracy (ACC) of 78.33%, precision (PRE) of 55.76%, recall (SE) of 77.95%, and an F-score of 65.02%. In contrast, our convolutional Siamese neural network model exhibited enhanced performance, achieving an ACC of 83.6%, PRE of 98.26%, SE of 67.87%, and an F-score of 80.28%. We also calculated the specificity value as 98.77%. It is important to highlight that Richoux's strict dataset includes mirror copies of proteins, mostly from the positive dataset. The inclusion of mirror copies has a considerable effect on the precision value, which was evaluated at 98.26% utilizing Siamese neural network structure. Various metrics highlight the importance of removing these mirror copies from the test set. Here, we would like to note that the recall (SE) value significantly increases to 74.04% when we remove mirror copies from the dataset, as we already indicated in Table 5.

These findings highlight that our Siamese-CNN model surpassed Richoux's model in terms of accuracy (ACC), precision (PRE), and the F-score. While Richoux's model exhibited higher sensitivity, our model displayed superior overall performance across various metrics.

The strict dataset effectively addressed the issue of overfitting, and our network demonstrated strong performance when evaluated with this dataset. Our model's improved accuracy (ACC), precision (PRE), sensitivity (SE), and F-score demonstrate its usefulness in predicting PPIs as well as its capacity to generalize effectively to information that was previously unknown.

4. Conclusion and Suggestions

In the present study, we developed and applied a convolutional Siamese neural network model for predicting PPIs using only protein sequencing data. We developed

Table 6. Comparison between our network and Richoux et al.'s fully connected model on strict dataset.

Study	Richoux et. al.	Our Method
Feature	Binary Encoding	Binary Encoding
Classifier	recurrent neural model	Convolutional Siamese NN
ACC (%)	78.33	**83.36**
PRE (%)	55.76	**98.26**
SE (%)	**77.95**	67.87
F-Score (%)	**83.36**	80.28

an ensemble encoding strategy to construct an optimum feature representation capable of encapsulating the critical information about protein interactions. This method was developed by combining three unique encoding strategies, each of which was aimed to capture distinctive characteristics inherent in individual protein sequences. The combination of Siamese networks and PSSMs can enable the model to discriminate between interacting and non-interacting protein pairs more effectively. This integration ultimately enhances the prediction accuracy and overall performance of the model. The inclusion of PSSMs, which allows the network to leverage the informative evolutionary patterns inherent within the protein sequences, is responsible for the improvement. Furthermore, we used the collaborative power of a Siamese neural network and a multilayer feed-forward neural network (MLF-NN). When compared to using various encoding approaches in isolation, the ensemble model demonstrated significant performance improvements. Our model signifies a substantial advancement in the field of PPI prediction, with the potential to greatly enhance the accuracy and reliability of these predictions.

Author contribution: All authors contributed equally to the study.

Ethics committee approval: The author declares that this study complies with Research and Publication Ethics.

5. References

Alkhalid, FF. 2022. The effect of optimizers on siamese neural network performance. Proceedings of the International Conference on Industrial Engineering and Operations Management. Doi:10.46254/an12.20221019

Altschul, SF., Madden, TL., Schäffer, AA., Zhang, J., Zhang, Z., Miller, W., Lipman, DJ. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Research, 25(17):3389–3402. Doi:10.1093/nar/25.17.3389

Angermueller, C., Pärnamaa, T., Parts, L. , Stegle, O. 2016. Deep learning for computational biology: Molecular Systems Biology, 12:878. Doi:10.15252/msb.20156651

Browne, F., Zheng, H., Wang, H., Azuaje, F. 2010. From Experimental Approaches to Computational Techniques: A Review on the Prediction of Protein-Protein Interactions. Advances in Artificial Intelligence, 2010:924529. Doi:10.1155/2010/924529

Chen, W., Wang, S., Song, T., Li, X., Han, P., Gao, C. 2022. DCSE:Double-Channel-Siamese-Ensemble model for protein protein interaction prediction. BMC Genomics, 23(1):555. Doi:10.1186/s12864-022-08772-6

ElAbd, H., Bromberg, Y., Hoarfrost, A., Lenz, T., Franke, A., Wendorff, M. 2020. Amino acid encoding for deep learning applications. BMC Bioinformatics, 21(1):235. Doi:10.1186/s12859-020-03546-x

Elnaggar, A., Heinzinger, M., Dallago, C., Rihawi, G., Wang, Y., Jones, L., … Rost, B. 2021. ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Learning. Doi:10.48550/arXiv.2007.06225

Fauchère, J., Charton, M., Kier, L., Verloop, A. , Pliska, V. 1988. Amino acid side chain parameters for correlation studies in biology and pharmacology. International journal of peptide and protein research, 32:269–278. Doi:10.1111/j.1399-3011.1988.tb01261.x

Gao, H., Chen, C., Li, S., Wang, C., Zhou, W., Yu, B. 2023. Prediction of protein-protein interactions based on ensemble residual convolutional neural network. Computers in Biology and Medicine. 152:106471. Doi:10.1016/j.compbiomed.2022.106471

Gao, ZG., Wang, L., Xia, SX., You, ZH., Yan, X., Zhou, Y. 2016. Ens-PPI: A Novel Ensemble Classifier for Predicting the Interactions of Proteins Using Autocovariance Transformation from PSSM. BioMed Research International, 2016:4563524. Doi:10.1155/2016/4563524

Gligorijević, V., Renfrew, PD., Kosciolek, T., Leman, JK., Berenberg, D., Vatanen, T., … Bonneau, R. 2021. Structure-based protein function prediction using graph convolutional networks. Nature Communications. 12(1):3168. Doi:10.1038/s41467-021-23303-9

Guo, Y., Yu, L., Wen, Z., Li, M. 2008. Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences. Nucleic acids research, 36:3025–3030. Doi:10.1093/nar/gkn159

Hashemifar, S., Neyshabur, B., Khan, A.A., Xu, J. 2018. Predicting protein–protein interactions through sequence-based deep learning. Bioinformatics, 34(17):i802–i810. Doi:10.1093/bioinformatics/bty573

Jia, LN., Yan, X., You, ZH., Zhou, X., Li, LP., Wang, L., Song, KJ. 2020. NLPEI: A Novel Self-Interacting Protein Prediction Model Based on Natural Language Processing and Evolutionary Information. Evolutionary Bioinformatics, 16:1176934320984171. Doi:10.1177/1176934320984171

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., … Hassabis, D. 2021. Highly accurate protein structure prediction with AlphaFold. Nature. 596(7873):583–589. Doi:10.1038/s41586-021-03819-2

Li, J., Chen, Y. 2013. Auto Covariance Combined with Artificial Neural Network for Predicting Protein-Protein Interactions, V. 765–767. Doi:10.2991/icsem.2013.153

Madan, S., Demina, V., Stapf, M., Ernst, O., Fröhlich, H. 2022. Accurate prediction of virus-host protein-protein interactions via a Siamese neural network using deep protein sequence embeddings, Patterns. 3(9):100551. Doi:10.1016/j.patter.2022.100551

Nevers, Y., Glover, NM., Dessimoz, C., Lecompte, O. 2023. Protein length distribution is remarkably uniform across the tree of life. Genome Biology, 24(1). Doi:10.1186/s13059-023-02973-2

Nourani, E., Asgari, E., McHardy, AC., Mofrad, MRK. 2022. TripletProt: Deep Representation Learning of Proteins Based On Siamese Networks. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 19(6):3744–3753. Doi:10.1109/TCBB.2021.3108718

Özger, ZB., Çakabay, Z. 2023. Computational Prediction of Interactions Between SARS-CoV-2 and Human Protein Pairs by PSSM-Based Images. Bitlis Eren *Üniversitesi* Fen Bilimleri Dergisi, 12(1):166–179. Doi:10.17798/bitlisfen.1220301

Poplin, R., Chang, PC., Alexander, D., Schwartz, S., Colthurst, T., Ku, A., … DePristo, MA. 2018. A universal SNP and small-indel variant caller using deep neural networks. Nature Biotechnology, 36(10):983–987. Doi:10.1038/nbt.4235

Richoux, F., Servantie, C., Borès, C., Téletchéa, S. 2019. Comparing two deep learning sequence-based models for protein-protein interaction prediction. Doi:10.48550/arXiv.1901.06268

Shen, J., Zhang, J., Luo, X., Zhu, W., Yu, K., Chen, K., Li, Y., Jiang, H. 2007. Predicting protein–protein interactions based only on sequences information. Proceedings of the National Academy of Sciences, 104(11):4337–4341. Doi:10.1073/pnas.0607879104

Sun, T., Zhou, B., Lai, L., Pei, J. 2017. Sequence-based prediction of protein protein interaction using a deep-learning algorithm. BMC Bioinformatics, 18(1):277. Doi:10.1186/s12859-017-1700-2

Tong, J., Tammi, M. 2008. Prediction of protein allergenicity using local description of amino acid sequence. Frontiers in bioscience : a journal and virtual library, 13:6072–6078. Doi:10.2741/3138

Trieu, T., Martinez-Fundichely, A., Khurana, E. 2020. DeepMILO: a deep learning approach to predict the impact of non-coding sequence variants on 3D chromatin structure. Genome Biology, 21(1):79. Doi:10.1186/s13059-020-01987-4

Wang, L., Yo, ZH., Xia, SX., Liu, F., Chen, X., Yan, X., Zhou, Y. 2017. Advancing the prediction accuracy of protein-protein interactions by utilizing evolutionary information from position-specific scoring matrix and ensemble classifier. Journal of Theoretical Biology, 418:105–110. Doi:10.1016/j.jtbi.2017.01.003

Wang, X., Wang, R., Wei, Y., Gui, Y. 2019. A novel conjoint triad auto covariance (CTAC) coding method for predicting protein-protein interaction based on amino acid sequence. Mathematical Biosciences, 313:41–47. Doi:10.1016/j.mbs.2019.04.002

Wold, S., Jonsson, J., Sjörström, M., Sandberg, M., Rännar, S. 1993. DNA and peptide sequences and chemical processes multivariately modelled by principal component analysis and partial least-squares projections to latent structures. Analytica Chimica Acta, 277:239–253. Doi:10.1016/0003-2670(93)80437-P

Yang, X., Zhang, Z., Wuchty, S. 2021. Multi-scale convolutional neural networks for the prediction of human-virus protein interactions. In: ICAART 2021 - Proceedings of the 13th International Conference on Agents and Artificial Intelligence,. V. 2. SciTePress, 41–48. Doi:10.5220/0010185300410048

You, ZH., Lei, YK., Zhu, L., Xia, J., Wang, B. 2013. Prediction of protein-protein interactions from amino acid sequences with ensemble extreme learning machines and principal component analysis. BMC Bioinformatics, 14(8):S10. Doi:10.1186/1471-2105-14-S8-S10

You, ZH., Chan, K., Hu, P. 2015. Predicting Protein-Protein Interactions from Primary Protein Sequences Using a Novel Multi-Scale Local Feature Representation Scheme and the Random Forest. PloS one, 10:e0125811. Doi:10.1371/journal.pone.0125811

Zahiri, J., Yaghoubi, O., Mohammad-Noori, M., Ebrahimpour, R., Masoudi-Nejad, A. 2013. PPIevo: Protein–protein interaction prediction from PSSM based evolutionary information. Genomics, 102(4):237–242. Doi:10.1016/j.ygeno.2013.05.006

Zhang, L., Yu, G., Xia, D., Wang, J. 2019. Protein–protein interactions prediction based on ensemble deep neural networks. Neurocomputing, 324:10–19. Doi:10.1016/j.neucom.2018.02.097

Zhu, HJ., You, Z-H., Shi, WL., Xu, SK., Jiang, TH., Zhuang, LH. 2019. Improved Prediction of Protein-Protein Interactions Using Descriptors Derived From PSSM via Gray Level Co-Occurrence Matrix. IEEE Access, 7:49456–49465. Doi:10.1109/ACCESS.2019.2907132