ACS OMEGA

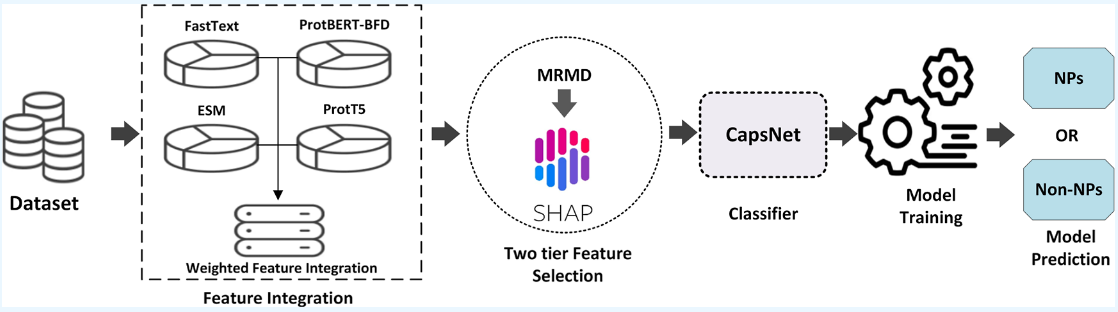
http://pubs.acs.org/journal/acsodf
Article

pNPs-CapsNet: Predicting Neuropeptides Using Protein Language Models and FastText Encoding-Based Weighted Multi-View Feature Integration with Deep Capsule Neural Network

Shahid Akbar, Ali Raza, Hamid Hussain Awan, Quan Zou,* Wajdi Alghamdi, and Aamir Saeed

Cite This: *ACS Omega* 2025, 10, 12403−12416
Read Online

ACCESS | Metrics & More | Article Recommendations



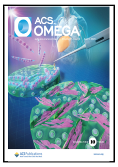
ABSTRACT: Neuropeptides (NPs) are critical signaling molecules that are essential in numerous physiological processes and possess significant therapeutic potential. Computational prediction of NPs has emerged as a promising alternative to traditional experimental methods, often labor-intensive, time-consuming, and expensive. Recent advancements in computational peptide models provide a cost-effective approach to identifying NPs, characterized by high selectivity toward target cells and minimal side effects. In this study, we propose a novel deep capsule neural network-based computational model, namely pNPs-CapsNet, to predict NPs and non-NPs accurately. Input samples are numerically encoded using pretrained protein language models, including ESM, ProtBERT-BFD, and ProtT5, to extract attention mechanism-based contextual and semantic features. A differential evolution-based weighted feature integration method is utilized to construct a multiview vector. Additionally, a two-tier feature selection strategy, comprising MRMD and SHAP analysis, is developed to identify and select optimal features. Finally, the novel capsule neural network (CapsNet) is trained using the selected optimal feature set. The proposed pNPs-CapsNet model achieved a remarkable predictive accuracy of 98.10% and an AUC of 0.98. To validate the generalization capability of the pNPs-CapsNet model, independent samples reported an accuracy of 95.21% and an AUC of 0.96. The pNPs-CapsNet model outperforms existing state-of-the-art models, demonstrating 4% and 2.5% improved predictive accuracy for training and independent data sets, respectively. The demonstrated efficacy and consistency of pNPs-CapsNet underline its potential as a valuable and robust tool for advancing drug discovery and academic research.

1. INTRODUCTION

Neuropeptides (NPs) represent a diverse and complex class of signaling molecules that influence nearly all physiological functions and behaviors in living organisms.[1] Typically composed of fewer than 100 amino acids, neuropeptides are derived from larger precursor molecules through a series of post-translational processing steps.[2] Beyond their role in the nervous system, neuropeptides also exert peripheral effects through the endocrine system, regulating various bodily processes, including metabolism, fluid homeostasis, food intake, cardiovascular function, reproduction, energy balance, stress response, pain perception, memory, learning, and social behaviors.[3] Based on their involvement in numerous physiological functions, NPs are closely associated with various disease processes. The neuropeptide signaling system has emerged as a promising therapeutic target for treating a wide range of conditions, such as epilepsy, hypertension, diabetes, heart failure, obesity, sleep disorders, autism, and depression.[4]

Considering the significance of NPs and the challenges associated with traditional in vitro methods such as laborious, expensive, and difficult processing of large-scale samples. Various computational models have been proposed in the last five years to accurately predict NPs. Kang et al. developed the

Received: December 24, 2024
Revised: February 4, 2025
Accepted: March 7, 2025
Published: March 18, 2025



https://doi.org/10.1021/acsomega.4c11449

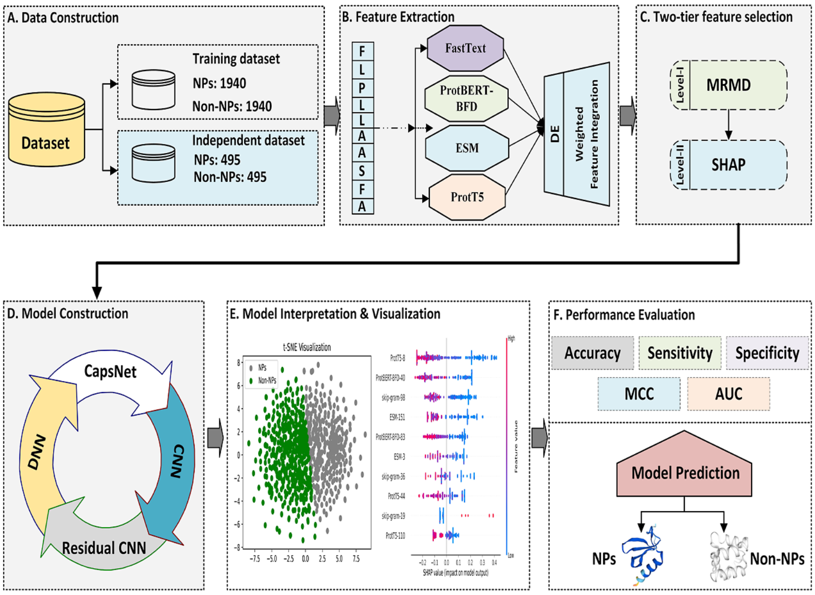


Figure 1. Framework of the proposed model.

NeuroPP predictor, which utilized a combination of residue frequency-based features to represent peptide samples.[5] The optimal feature set was selected from the entire vector and was trained using a support vector machine (SVM), achieving an accuracy of 88.62%. Similarly, in the NeuroPIpred model, the SVM model was trained using positional residue frequency and binary profile-based representation methods.[6] The encoded features were used to train the SVM model. Subsequently, Bin et al. proposed the PredNeuroP model, which trained a stacking ensemble model using sequential features to predict NP samples.[7] Initially, nine encoding methods were employed to extract features from the NP sequence, and then five machine-learning models were used to generate 45 baseline models. Among these, eight optimal baseline models were selected to create a stacking model. Likely, the NeuroPpred-Fuse model employed six different physiochemical and frequency-based sequential encoding methods.[8] The integrated vector from these encoding methods was processed through four different feature selection methods to achieve optimal features, based on which three machine learning models were trained. Finally, a logistic regression-based ensemble model was developed using the predicted probabilistic features of these models. Hasan et al. developed the NeuroPred-FRL model using a heterogeneous vector that incorporated 11 different feature extraction methods, including evolutionary, sequential, and structural amino acid properties.[9] The extracted features were trained using six machine-learning models to produce 66 different baseline models, and a two-step feature selection was employed to generate an optimal feature set of 66D. Ultimately, a random forest classifier was used to train this optimal feature set. Chen et al. proposed the first deep learning predictor, namely NeuroPred-CLQ for predicting NPs.[10] NeuroPred-CLQ explored semantic features from peptide sequences using word2vec-based embedding, and a multiheaded attention mechanism was employed to train temporal convolutional networks. Additionally, Liu et al. Presented the NeuroPpred-SVM model, which formulates NP samples using sequential encoding and a pretrained BERT model to extract semantic features.[11] Several machine learning and deep learning-based training models were investigated. Among these, the SVM model performed satisfactorily. In the NeuroCNN_GNB model, different sequential encoding and word2vec-based word embeddings were utilized to extract features from peptide samples.[12] The extracted numerical vectors were trained using an ensemble model of five machine-learning classifiers. Akbar et al. proposed the ensemble classifier by numerically exploring the peptide samples for evolutionary features using KSB, and bigram-PSSM methods.[13] Additionally, the discrete wavelet transform (DWT) based noise reduction approach was incorporated to produce effective and noiseless features. In the NeuroPred-PLM model, wang et al. used an ESM-based protein language model to generate word embedding features to predict the NP samples.[14] Then, a multiheaded attention-based multiscale CNN training model was employed to enhance the local representation of the ESM embedding. Similarly, Du et al. developed a universal deep learning approach namely UniDL4BioPep for predicting various types of bioactivity Peptides. In the UniDL4BioPep model, the ESM-based word embedding and semantic information are captured from the peptide and then the optimized CNN model was trained for predicting different peptide types.[15] In the case of NP samples, a predictive accuracy of 89.20% was reported using the proposed architecture.

After investigating the existing NPs based computational predictors, it is evident that most models employed sequential residue-based formulation methods without preserving the intrinsic structure of residue ordering within input sequences. Additionally, these predictors lack consideration of contextual relationships and often fail to capture the hidden structural features within peptides. Furthermore, some predictors utilize PSSM matrix-based evolutionary descriptors; however; for large-scale data sets, this approach is time-intensive as it requires searching for similarity matrices for each sequence in online databases. In the case of short peptide sequences, PSSM information may be unavailable due to insufficient data, which can adversely affect model prediction performance. Therefore, further enhancement is essential to effectively determine internal motifs and structural variations in peptides. In the case of feature selection, the existing methods primarily rely on

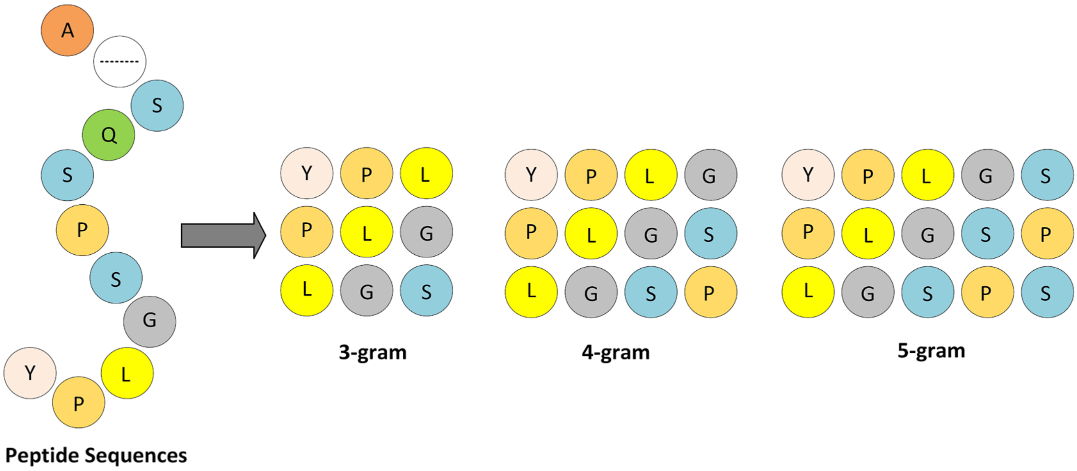


Figure 2. Skip-gram embedding using FastText.

the traditional filter-based methods, which are not effective in choosing optimal discriminative features. During the model training, current models employ conventional learning methods, which come with high execution costs. Consequently, comprehensive improvements and alternative solutions are desirable regarding encoding schemes, training capabilities, model interpretability and generalization, computational efficacy, and prediction results to efficiently discriminate NPs and non-NPs.

In this study, we introduce a novel computational predictor, pNPs-CapsNet designed for the accurate prediction of NP samples and non-NP samples. The framework of the pNPs-CapsNet model is shown in Figure 1. The primary contributions of our model are as follows:

(a) The input samples are formulated using ESM, ProtBERT-BFD, and ProtT5-based pretrained protein language models. Additionally, FastText-based word embedding is employed to capture local contextual relationships and semantic features.

(b) A differential evolution (DE) based weighted feature integration is employed to integrate all extracted single-view features into a weighted multiview vector.

(c) To reduce the computational costs, a two-tier feature selection approach, using MRMD and SHAP analysis is utilized to select optimal features.

(d) Various deep learnings are evaluated, among which our novel capsule neural network (CapsNet) demonstrated superior performance.

(e) To ensure the model's generalization, multiple validation and independent tests on unseen data are conducted, proving the model's effectiveness.

2. MATERIAL AND METHODS

2.1. Data Set. In deep learning-based intelligent predictors, establishing a reliable and comprehensive benchmark data set is a crucial step.[16] A well-constructed data set enables fair evaluation and assessment of a model's predictive ability. In this study, we derived the training data set from the work of Jiang et al.,[17] initially proposed by Bin et al.[7] The data set includes 5948 laboratory-evaluated positive samples (NPs), sourced from diverse taxa in the NeuroPep databanks. Sequences exceeding 50 residues or shorter than 5 residues were excluded to ensure consistency. To further enhance the model's effectiveness, homologous peptides were eliminated using the CD-HIT tool with a similarity threshold of 0.9.[18] Hence, 2425 NPs were collected as positive samples. Similarly, an equal number of negative samples (non-NPs) were derived from Swiss-prot and processed, ensuring redundancy removal. Ultimately, a benchmark training data set of 3880 samples was established containing 1940 NPs and 1940 non-NPs. To evaluate the model's generalization and mitigate overfitting, 20% of the samples were reserved as an independent data set. The independent data set consists of 970 samples, including 495 NPs and 495 non-NPs. It was ensured that no sequences from the training data set were present in the independent data set.

2.2. Feature Encoding Methods. *2.2.1. FastText-Based Skip-Gram Embedding.* Word embedding-based feature encoding has received considerable attention in the past few years, largely due to the impressive results achieved in computational biology.[19] The primary aim of these approaches is to map sequences of amino acids into continuous vectors linearly, without variations in their actual locations or semantic meaning within the input data.[20] Therefore, several word embedding models such as Bag of Words (BOW), ProtVec, word2vec, and GloVe have been introduced to encode features from peptide sequences.[21,22] In comparison with earlier methods, the FastText word embedding addresses certain limitations by tokenizing peptides into subwords (*n*-grams),[22,23] rather than representing each peptide as a single continuous sequence. FastText utilizes morphological features to manage out-of-vocabulary words and enhance performance in downstream tasks by accommodating variations, including spelling errors, and slang.[24] FastText employs two types of embedding techniques: a continuous bag of words (CBOW), and skip-gram for creating word-level embeddings. In this study, we utilized the skip-gram method for n-gram extraction of peptide sequences, as shown in Figure 2.[25]

Let us consider a peptide sequence "*A*" that contains of "*L*" number of amino acids.

$$A = A_1A_2A_3. . . . . A_L \tag{1}$$

where A_i signifies the *i*th amino acid in the peptide sample, and *L* represents the sequence length. The main purpose of the skip-gram model is to increase the average log-likelihood, which can be represented as

$$\delta_{\text{Skip-gram}} = \frac{1}{L}\sum_{q=1}^{L}\sum_{-c\le k\le c,j\ne 0}\log P(A_{q+k}|A_k) \tag{2}$$

where the window size "c" represents the no; of amino acids considered from the left and right side of the target amino acid. In this paper, we employ a context window size of "8". The probability of the A_{q+k} amino acid appearing in the context given the target amino acid A_k can be computed as

$$P(A_{q+k}|A_k) = \frac{\exp(y'_{ni}y_{no})}{\sum_{n=1}^{Q}\exp(y'_{ni}y_{no})} \tag{3}$$

The equation includes two feature sets of the amino acid L, signified as y'_{ni} and y_{no}. The subscripts "i" represent the output and "o" is the input amino acids. In NLP, calculating the log-likelihood for a large vocabulary text corpus can become computationally infeasible. To address this, the log-likelihood can be obtained by transforming each log $\sigma(y'_{ni}y_{no})$ as follows

$$\log\sigma(y'_{ni}y_{no}) + \sum_{i=1}^{J} E_{ni} \sim P_n(w)[\log\sigma(y'_{ni}y_{no})] \tag{4}$$

where

$$\sigma(x) = \frac{1}{1+\exp(-x)} \tag{5}$$

In eq 4, the sign 'J' denotes the negative sequences. The primary purpose is to examine, how effectively the model can predict NPs and non-NPs.

2.2.2. Pretrained Protein Language Models (pLMs). Recently, inspired by the significant successes of transformer-based language models in natural language processing, scientists have successfully adapted these models to predict protein sequences.[26] The primary aim of applying these pLMs on large data sets of protein samples is to enhance protein representation. pLMs have the potential to learn the contextual relationships within protein sequences, which are highly valuable for addressing various challenges related to understanding protein functions. pLMs efficiently capture intrinsic information by considering both individual amino acids (local information) and their interaction across the entire protein (global information).

2.2.2.1. ProtBERT-BFD. In BERT encoding models, amino acid samples are predicted using complicated semantic relationships, and contextual information is extracted from extensive protein databases.[27] In the BERT, peptide samples are treated as sentences to represent contextual relationships through embedded features.[28] Each sentence consists of a comprehensive representation of individual amino acids.[29] Leveraging the outstanding results of BERT models, we applied a self-attention-based ProtBert-BFD model to collect useful features from the amino acid sequences. The ProtBert-BFD model not only preserves the sequence order-based features but also offers residue-based frequency descriptors. In this paper, we applied the pretrained ProtBERT-BFD model to convert the peptide samples into word-embedding vectors. In this paper, we applied the pretrained ProtBERT-BFD model to convert the peptide samples into word-embedding vectors. Each peptide sample is considered as a sentence and divided into 200 different tokens, starting with a special token called "CLS", by aggregating the embedding features of all words in an input sample.[30] Consequently, a fixed 200-dimensional vector is produced by padding the sample with a "PAD" token to handle variations in the length of peptide samples. To separate the information on each peptide sample, a "SEP" token is used.[31] After encoding the amino acid sequences through tokenization, the input is generated for the transformer function, which produces a hidden state at the self-attention layer as the output.[32] Finally, each peptide sentence represented by 200 tokens is embedded to form a feature space of 1024D using global average pooling. The generated vectors are then provided to the input layer of the deep learning model for training.

2.2.2.2. ProtT5 Based Transformer. The ProtT5 is a self-supervised pretrained model that performs transfer learning-based word embedding to encode protein samples.[33] As a type of protein language model, ProtT5 leverages the knowledge from data-rich problems to handle data-limited problems. It enhances downstream prediction by learning summarized representation, which can distill knowledge from large-scale data sets. One of the key characteristics of ProtT5 is to capture global contextual features, which represent protein conformations.[34] Unlike other supervised word embedding models, ProtT5 inputs the full-length samples (i.e., without window size), producing embedding for all residues in a protein sample. In the ProtT5 mechanism, each of the amino acids in the sequence is mapped to a fixed-size vector representation using an embedding layer, and positional encoding is used to capture the positions of the amino acids within the sequence as illustrated in Figure 3. Subsequently, several Transformer encoders are sequentially connected to extract the internal structure of a sequence, and the resulting high-level representations allow for obtaining detailed feature information. ProtT5 does not focus only on the sequence but also on the relative positioning of the elements of the sequence. Ultimately, training sequences are converted into a numerical vector of 1024D. In this study, we used the ProtT5-XL-U50 pretrained model to encode the features of peptide sequences.

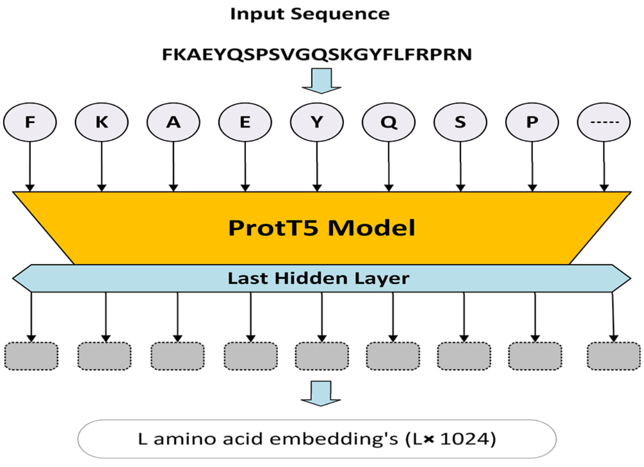


Figure 3. Sequence embedding via ProtT5Model.

2.2.2.3. Evolutionary Scale Modeling. Evolutionary scale modeling (ESM)[35] is a powerful pLM based on the architecture of the BERT transformer.[36] The ESM model is trained to develop a comprehensive understanding of proteins. Numerous researchers have effectively used the ESM model as a feature-encoding method in protein-related tasks.[37] Different variants of the ESM model are available, varying based on factors such as data sets and the number of parameters. In this study, we employed a variant of the ESM model called esm1b_t33_650M_UR50S (referred to as ESM-1b), which is trained on the UniRef50[38] and has approximately 650 million learnable parameters.

First, we created a FASTA file for each data set, which was then provided to ESM-1b to learn and extract the embeddings from each peptide sample. The embeddings of the last hidden layer of the ESM-1b are extracted in this study, with a size of 1280 × L for each peptide sequence, where L is the peptide sequence length. Then, we applied global average pooling (GAP) to generate the final-dimensional feature vector representing each peptide sample. The schematic architecture of the ESM-1b model is illustrated in Figure 4.

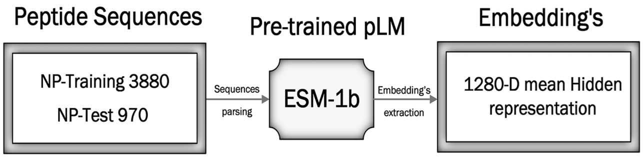


Figure 4. Sequence embedding using ESM-1b model.

2.3. Differential Evolution-Based Feature Integration.

In developing a computational model, feature integration is a necessary step to fuse the features of individual vectors, forming a multiperspective vector.[39] The hybrid vector represents the high discriminative features of each encoding method, leading to accurately discerning the targeted classes. Typically, serial feature integration is widely used to concatenate the extracted features of individual encoding methods. However, serial integration approach treats all features equally without any priority or feature weighting. In contrast, differential evolution-based feature integration assigns weights to the highly informative features, and these weighted features are then fused to produce an optimal hybrid feature vector.[40] Compared to serial feature integration, the differential evolution-based weighted feature vector can significantly improve the predictive performance of the model. Keeping its significance, we utilized differential evolution (DE) to integrate the extracted features of the encoding methods[40,41] due to its high robustness and easy implementation.[42] Comprehensive details on learning the optimal weights of the extracted features are as follows:

Let $f(s)$ represent the optimization problem of identifying the highest MCC value for the k-fold CV test on the training samples. Here, "S" denotes the candidate solution, symbolized by $\boldsymbol{S} = (\boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{s}_3, \boldsymbol{s}_4)^T$. Each element $\boldsymbol{S}$ represents the assigned weight to a specific feature vector such as $\boldsymbol{s}_1$ for FastText, $\boldsymbol{s}_2$ for ProtBERT-BFD, $\boldsymbol{s}_3$ for ProtT5, and $\boldsymbol{s}_4$ represents ESM. All assigned weights are within the range of [−2, 2] i.e., $\boldsymbol{s} = (s_1, s_2, s_3, s_4)^T \in [-2, 2]^4$. The mechanism of DE-based integration can be described as follows:

2.3.1. Mutation. The mutation step is applied to obtain variation in the search space. Within the population P^g, each candidate solution s_k^g, generate a mutant vector s_k^{g+1} as follows

$$\boldsymbol{v}_k^{g+1} = \boldsymbol{s}_{r_1}^g + F\cdot(\boldsymbol{s}_{r_2}^g - \boldsymbol{s}_{r_3}^g) \tag{6}$$

where $\boldsymbol{s}_{r_1}^g$, $\boldsymbol{s}_{r_2}^g$ and $\boldsymbol{s}_{r_3}^g$ represent three different candidate solutions randomly selected in subset $P^g - \{s_k^g\}$.

2.3.2. Crossover. To avoid premature convergence, the crossover function is applied to increase the diversity in the next generation P^{g+1}. By fusing elements from s_k^g in P^g, and $\boldsymbol{v}_k^{g+1}$, produce a trial vector $\boldsymbol{u}_k^{g+1} = (u_{k,1}^{g+1}, u_{k,2}^{g+1}, u_{k,3}^{g+1}, u_{k,4}^{g+1})^T$ as follows

$$\boldsymbol{u}_{k,q}^{g+1} = \begin{cases} \boldsymbol{v}_{k,q}^{g+1}, & \text{if } \mathrm{rand}(0, 1) < CR \text{ or } \boldsymbol{q} = q_{\mathrm{rand}} \\ \boldsymbol{w}_{k,q}^{g}, & \text{otherwise} \end{cases} \tag{7}$$

where q_{rand} is the randomly selected index of {1, 2, 3, 4} i.e. $q_{\mathrm{rand}} \in \{1, 2, 3, 4\}$.

2.3.3. Selection. Based on the fitness function $f(\cdot)$, compare the $f(\boldsymbol{u}_k^{g+1})$ of the trail vector $\boldsymbol{u}_{k,q}^{g+1}$ with the $f(\boldsymbol{s}_k^g)$ of the target solution $\boldsymbol{s}_k^g$. If the $\boldsymbol{u}_k^{g+1}$ has better fitness value than s_k^g, $\boldsymbol{u}_k^{g+1}$ is set to be $\boldsymbol{s}_k^{g+1}$ in the next generation P^{g+1}, otherwise, $\boldsymbol{s}_k^g$ is set to $\boldsymbol{s}_k^{g+1}$

$$\boldsymbol{s}_k^{g+1} = \begin{cases} \boldsymbol{u}_k^{g+1}, & \text{if } f(\boldsymbol{u}_k^{g+1}) > f(\boldsymbol{s}_k^g) \\ s_k^g, & \text{otherwise} \end{cases} \tag{8}$$

2.3.4. Termination. Finally, terminate the process if $g > G_{\max}$ and output the best solution $\boldsymbol{s}_{\mathrm{optimal}}$ in the population P^g, else, repeat steps 2−4.

In this study, we achieved the optimal feature weights $\boldsymbol{s}_{\mathrm{optimal}}$=(−1.1867, 1.1, 1.1649, 0.1875), where the weights of FastText, ProtBERT-BFD, ProtT5, and ESM are −1.1867, 1.1, 1.1649, and 0.1875, respectively. Lastly, we can obtain the integrated feature vector via weights-based integration of FastText, ProtBERT-BFD, ProtT5, and ESM.

2.4. Two-Tier Feature Selection.

In bioinformatics, to develop a computationally efficient model with a low training cost, an effective feature selection method is essential to enhance the model throughput.[43] Various feature selection approaches including wrapper,[44] filters,[45] and Intrinsic techniques[46] are available in the literature. In this study, we proposed a two-layer feature selection approach to develop a computationally effective training model. The relevant details of these feature selection layers are described as follows:

2.4.1. Minimum Redundancy and Maximum Diversity (MRMD). In this study, we used the improved MRMD approach that was previously applied in the CWLy-pred model.[47] The main idea behind MRMD is that a feature's significance is determined by its high correlation with the target vector and low correlation with other feature vectors.[48] To evaluate the relevancy of a feature vector toward its targeted class, the Pearson correlation coefficient is employed and then the Euclidian distance is used to calculate the distance among the extracted feature spaces. Mathematically, these parameters can be represented as follows

$$\mathrm{Rel}_i = \frac{\sum_{j=1}^{K}\left(f_i(j) - \frac{1}{K}\sum_{j=1}^{K} f_i(j)\right)\left(\mathrm{Label}(j) - \frac{1}{K}\sum_{j=1}^{K}\mathrm{Label}(j)\right)}{\sqrt{\sum_{j=1}^{K}(f_i(j) - \frac{1}{K}\sum_{j=1}^{K}(f_i(j))^2}\sqrt{\left(\sum_{j=1}^{K}\left(\mathrm{Label}(j) - \frac{1}{K}\sum_{j=1}^{K}\mathrm{Label}(j)\right)\right)^2}} \tag{9}$$

$$\mathrm{Dist}_i = \frac{1}{K}\sum_{i=1}^{K}\sqrt{\sum_{j=1}^{K}(f_j(j) - f_i(j))^2} \tag{10}$$

where Rel represents the relevance, Dist is the distance among two features, K is the number of total sequences, $f_i(j)$ is the ith feature of the jth sequence, Label(j) is the jth value of the targeted vector.

$$\mathrm{MRMD\_score}_i = \mathrm{Rel}_i + \mathrm{Dist}_i \tag{11}$$

2.4.2. SHAP Feature Selection. In the past decade, explainable-AI-based feature analysis has gained significant attention for its ability to highlight and retain the biological contributions of features toward targeted classes.[49] Shapley Additive Explanation (SHAP) is a globally interpretable tool in explainable AI that utilizes machine learning (ML) models to compute shapely values, which are useful for assessing the magnitude of the impact of each extracted feature.[50] The concept of game theory is applied using the explainability of ML models in predicting peptide sequences based on their characteristics and predicted shapely values, i.e., high and low. However, calculating shapely values for large-scale data sets with high feature dimensions is not always optimal due to their high processing cost.

In this study, we applied the BorutaSHAP approach to select the highly contributory features from the selected mRMD feature set by examining their feature contribution. BorutaSHAP accelerates the training procedure by identifying the global feature importance. In the second tier of feature selection, 385 optimal features are selected from the 954 mRMD-based selected set. An objective function "K" and its corresponding Shapley value δ_D for each extracted descriptor $D \in F$ are calculated. The top-ranked features, signified by "R", where $R < d = |F|$, are selected. The summarized SHAP analysis of the top 10 contributory features is illustrated in Figure 5. In each row, the SHAP values of the selected features are indicated using red and blue colors. The abscissae in Figure 5 represent red points for highly valued features and blue points for low-valued features. Additionally, the instance-based LIME analysis of the independent samples are provided in Figure 6.

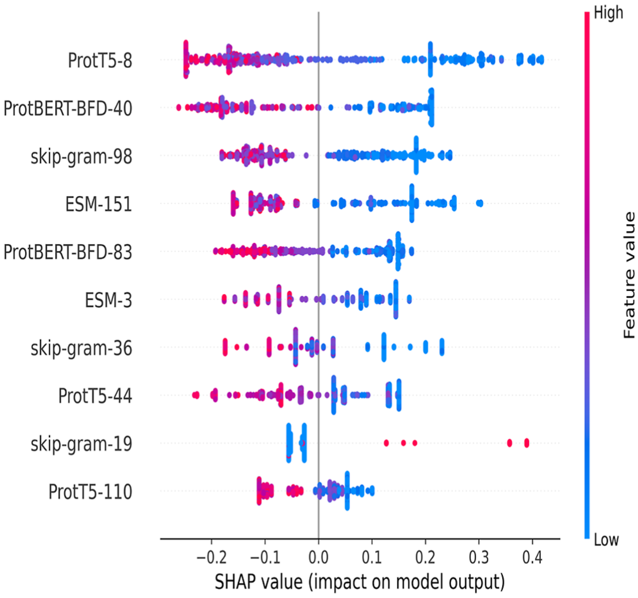


Figure 5. Model Interpretation of the SHAP analysis of top 10 features.

2.5. Capsule Neural Network Architecture and Model Training. Over the past decade, conventional deep learning models, especially convolutional neural networks (CNNs), have achieved substantial advancements across various research areas, such as bioinformatics,[51] image processing, and computer vision.[52] These models have outperformed traditional machine learning models due to their curated feature encoding strategies. Despite their effectiveness, these models have several issues, including an inability to capture the spatial relationships among extracted features and invariance due to their pooling process.[53] To tickle such limitations, the Capsule Network (CapsNet) was proposed by Sabour et al.[54,55] CapsNet is a novel deep-learning model that uses a capsule (collection of neurons) whose activity vector represents the instantiation parameters of a specific entity, such as an object or its components.[54] It means that the length of an activity vector indicates the likelihood of the entity's presence, while its orientation (activity vectors) represents the instantiation parameters. When predictions from low-level capsules are consistently aligned, high-level capsules become activated. Consequently, the low-level capsule aims to send its predicted value to the corresponding high-level capsule. A comprehensive detail of the CNN-based feedforward CapsNet has been discussed by.[56] The architecture of our proposed pNPs-CapsNet model is illustrated in Figure 7. The pNPs-CapsNet model comprises three one-dimensional convolutional layers, namely Conv1, Conv2 & Conv3, followed by the PrimaryCaps layer, and a fully connected layer referred to as NPsCaps. The initial three layers function as standard convolutional layers, transforming the input vector from its original form into intermediate-level features. These features are subsequently processed by the PrimaryCaps and NPsCaps layers for further feature abstraction, enhancing the predictive power of the pNPs-CapsNet model. The optimal hyperparameters were selected using the grid search approach, While Conv1, Conv2, and Conv3 layer hyperparameters were optimized through Bayesian methods.[57] Conv1 employs 64 1D convolutional kernels of size 1, Max-pooling with size 2 with a stride of 1, and ReLU activation.[58] Additionally, a dropout rate of 0.5 was applied to mitigate overfitting.[59] Conv2 uses 64 1D convolutional kernels with size 6, Max-pooling with size 2, and Conv3 similarly uses 32 1D convolutional kernels, with size 1 and Max-pooling with size 2. The details of optimal parameters are provided in Table 1. The output of convolution layers is fed into the PrimaryCaps layer, a 1D convolutional layer with 20 channels of convolutional capsules. Each capsule contains eight convolutional units, generated using a 1D convolution kernel of size 20 and a stride of 1, resulting in 8D vector capsules in PrimaryCaps. Each capsule shares its weight with other capsules, represented based on their probabilities. Capsule layers use the squashing activation function to constrain the capsule lengths within the range (0−1),[54] Notably, the dynamic routing process between capsules is employed between the PrimaryCaps and NPsCaps layers. The final layer, NPsCaps, consists of 8-dimensional capsules for each of the positive and negative classes. The positive capsules represent NPs, while the negative capsules represent the likelihood of non-NPs. Finally, L2 normalization is applied to rescale the output vectors from the NPs Caps layer.

3. PERFORMANCE EVALUATION MEASURES

In computational bioinformatics and deep learning, various parameters are used to evaluate the performance of intelligent models.[60] A model's correct and incorrect predictions are organized in a confusion matrix. Although accuracy is commonly used to assess the model's predictive strength, it is insufficient to measure overall efficiency. Therefore, to evaluate the predictive performance of the proposed pNPs-CapsNet Model, several metrics are applied, including accuracy, sensitivity (Sen), specificity (Spe), Matthews

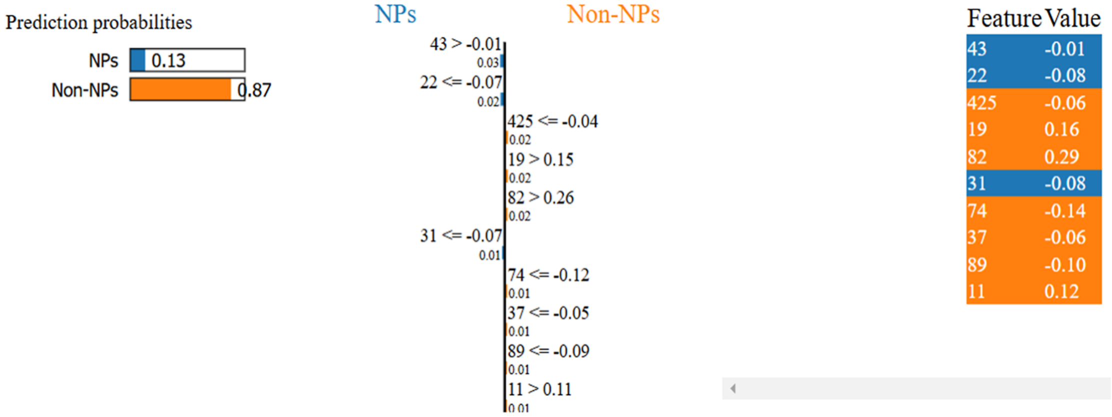


Figure 6. LIME analysis of independent data set.

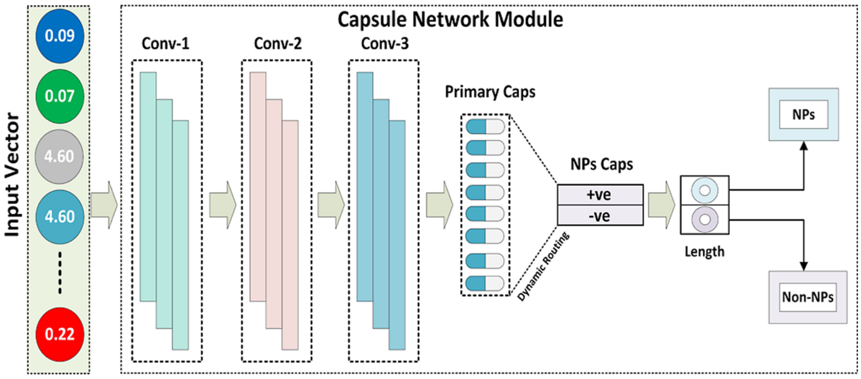


Figure 7. Architecture of the proposed pNPs-CapsNet Model.

Table 1. Hyper Parameters of pNPs-CapsNet Model

hyper-parameters	values
Conv1	number of filters = 64, kernel size = 1
Max pooling	size = 2
Conv2	number of filters = 64, kernel size = 6
Max pooling	2
Conv3	Number of Filters = 32, Kernel Size = 1
Max pooling	2
L2 regularization	0.001
primary capsule	capsule dimension = 8, number of channels = 20
DigitCaps	capsule dimension = 10, number of neurons = 128
sStride	1
dropout rate	0.5
optimizer	Adam, SGD
epochs	20−40
learning rate	0.001
dense layer unit	2
batch size	16, 32, 64
loss function	binary cross entropy
activation function	ReLU, squash, sigmoid

correlation coefficient (MCC), and area under the curve (AUC), as described below

$$\mathrm{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}} \qquad (12)$$

$$\mathrm{Sn} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \qquad (13)$$

$$\mathrm{Sp} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \qquad (14)$$

$$\mathrm{MCC} = \frac{(\mathrm{TN} \times \mathrm{TP}) - (\mathrm{FN} \times \mathrm{FP})}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}} \qquad (15)$$

Here, TP denotes the true positive samples (neuropeptides) while TN represents the true negative samples (non-neuropeptides). Similarly, FN indicates the false negative rates, which is an error as the model inaccurately predicted the samples to be true. Lastly, FP represents another type of error in which the model is incorrectly predicted as false, where the sample is true in reality. The MCC values are computed in the range (−1 to 1).

4. RESULT AND DISCUSSIONS

In this model, we employed 5-fold cross-validations (CV) and independent testing to evaluate the efficacy of our proposed pNPs-CapsNet Model. To ensure an unbiased and random distribution of the training features, we applied a stratified looping strategy for 100 iterations, obtaining the mean prediction across these iterations to assess the model's results.[61] In the following subsections, we briefly discuss the predicted results of the training and test samples in terms of various encoding methods and classification models.

4.1. Predictive Performance of the Classifiers Using Individual and Weighted Hybrid Vector.

The predictive performance of the applied encoding methods, including

Table 2. Prediction Analysis of Individual and Hybrid Feature Vectors Using Training Samples

encoding method	classifier	accuracy (%)	sensitivity (%)	specificity (%)	MCC	AUC (%)
ProtT5	CNN	83.14	84.79	81.61	0.66	0.90
	DNN	82.21	82.59	81.79	0.64	0.90
	RCNN	89.92	87.56	92.28	0.79	0.93
	CapsNet	92.78	91.17	94.56	0.85	0.95
ProtBERT-BFD	CNN	89.07	88.68	89.46	0.78	0.92
	DNN	86.58	87.20	85.94	0.73	0.91
	RCNN	85.02	84.51	85.50	0.70	0.91
	CapsNet	90.48	90.05	90.92	0.80	0.93
ESM	CNN	85.97	87.26	84.54	0.71	0.90
	DNN	84.04	83.08	85.07	0.68	0.89
	RCNN	86.18	88.29	85.00	0.73	0.90
	CapsNet	90.97	89.46	92.66	0.82	0.92
skip-gram5	CNN	73.47	73.35	74.53	0.47	0.80
	DNN	72.03	74.26	69.56	0.43	0.78
	RCNN	61.85	57.52	66.18	0.23	0.62
	CapsNet	77.16	78.40	76.39	0.54	0.82
weighted hybrid vector	CNN	91.21	90.23	92.11	0.82	0.93
	DNN	88.70	86.28	91.17	0.77	0.92
	RCNN	92.21	97.10	87.31	0.84	0.94
	CapsNet	94.86	94.31	95.47	0.89	0.96

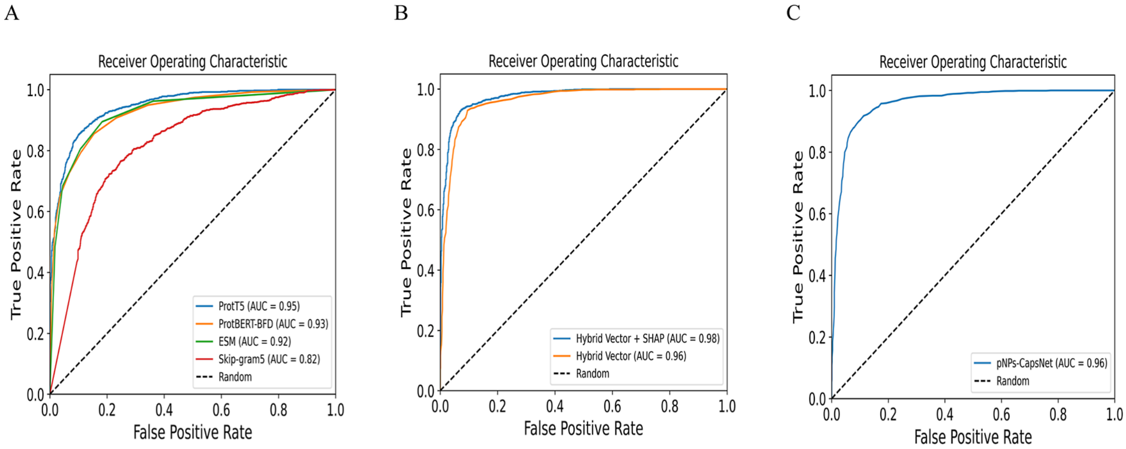


Figure 8. ROC analysis of training data set (A) individual features (B) hybrid and selected hybrid features (C) independent.

Table 3. Performance of the Classifiers after Two-Tier Feature Selection Using Training Samples

classifier	Acc (%)	Sen (%)	Spe (%)	MCC	AUC
CNN	96.91	98.33	95.42	0.93	0.98
DNN	94.97	95.34	94.56	0.89	0.95
RCNN	95.36	95.58	95.10	0.90	0.96
CapsNet	98.10	99.34	96.75	0.96	0.98

fastText-based Skip-gram5, ProtT5, ESM, and ProtBERT-BFD, was evaluated using various learning models such as convolutional neural network (CNN), deep neural network (DNN), residual convolutional neural network (RCNN), and capsule neural network (CapsNet), as provided in Table 2. Among the individual features, ProtT5 combined with the proposed CapsNet model achieved the highest predictive performance, with an accuracy (ACC) of 92.78%, Spe of 94.56%, Sen of 91.17%, MCC of 0.85, and AUC of 0.95. ESM features, also using the CapsNet model, demonstrated the second highest predictive results, with an ACC, Spe, Sen, MCC, and AUC of 90.97%, 92.66%, 89.46%, 00.82, and 0.92, respectively. Similarly, ProtBERT-BFD and Skip-gram5 features achieved accuracies of 90.48% and 77.16%, respectively, with corresponding AUC values of 0.93, and 0.82. Furthermore, other training models, namely CNN, DNN, and RCNN also exhibited favorable performance across all four encoding vectors. Based on a comprehensive comparison of the predictive performance of all the classifiers, the CapsNet model emerged as the most effective, outperforming the other classification models. To further examine the impact of individual encoding vectors, we calculated the ROC curve for each feature vector using the CapsNet model, as illustrated in Figure 8A. The results indicate that the individual feature vectors (ProtT5, ESM, and ProtBERT-BFD), exhibited consistent performance, thereby validating the significance of protein language model-based semantic features for predicting NPs and non-NPs.

To enhance the predictive capabilities of the encoded features, a weighted feature integration was performed using the DE approach to develop a multiperspective feature vector. The predictive performance of the DE-based integrated vector is provided in Table 2. The weighted hybrid vector (sfastText + sProtBERT-BFD + sESM + sProtT5) significantly improved

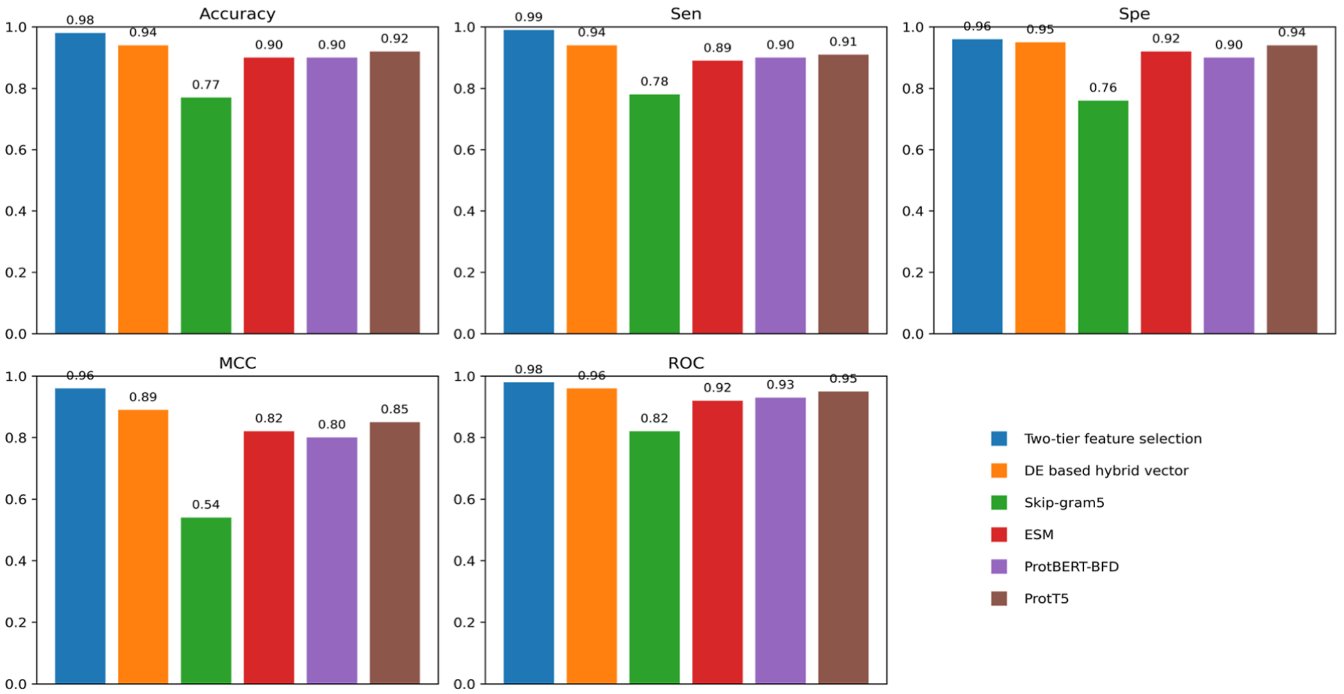


Figure 9. Comparative analysis of CapsNet training model using different feature vectors.

the predictive performance when used with the CapsNet model, achieving an ACC of 94.86%, Spe of 95.47%, Sen of 94.31%, MCC of 0.89 and AUC of 0.90, respectively. These findings highlight the potential benefits of incorporating weighted feature integration to enhance model predictive performance.

4.2. Effects on the Model Performance by Optimal Two-Tier Feature Selection.

Our predicted results in Section 4.1, clearly indicate that the weighted hybrid integration of the individual features, including fastText, ProtBERT-BFD, ESM, and ProtT5, enhances the predictive performance of NPs training samples, as shown in Table 2. To mitigate the model overfitting and reduce computational costs, we applied a two-tier feature selection approach, which played a significant role in improving the overall model performance. Initially, the mRMD-based filter approach was applied on the weighted hybrid vector of *s*fastText + *s*ProtBERT-BFD + *s*ESM + *s*ProtT5, resulting in a feature set of 954D. Subsequently, SHAP interpolation-based feature selection was performed in the second phase, yielding an optimal selected set of 385 dimensions. The predictive results of the proposed classifiers, using a stratified 5-fold CV looping strategy, are provided in Table 3. After applying the novel two-tier FS approach, the proposed CapsNet model achieved the highest prediction results, including an ACC, Sen, Spe, MCC, and AUC of 98.10%, 99.34%, 96.75%, 0.96, and 0.98, respectively. These results demonstrate that all classifiers consistently improved their predictive performance after applying the proposed FS approach. Specifically, the CapsNet model showed notable improvements, including a 3.24% increase in ACC, and 5.03%, 1.28%, 5%, and 2%, in Sen, Spe, MCC, and AUC, respectively as provided in Table 2. The comparative analysis of the CapsNet model, using all encoded vectors (individual vectors, weighted hybrid set, and selected optimal set) is illustrated in Figure 9. The t-SNE visualization of these encoded vectors is also presented in Figure 10. To validate the model generalization, all the classification models using the proposed feature encoding scheme were evaluated on unseen independent samples, as shown in Table 4. The proposed CapsNet model achieved higher predictive rates, with an Acc of 95.21%, Sen of 93.45%, Spe of 95.75%, MCC of 0.88, and AUC of 0.96. Additionally, to further analyze the proposed capsule training network, detailed ROC curves for training samples (individual, weighted hybrid vector, selected feature set), and independent samples are shown in Figure 8.

4.3. Model Visualization of the Encoded Vectors.

To further explore the effects of extracted features, we applied t-SNE to convert the high-dimensional extracted vectors in a 2D space.[62] t-SNE is a data mapping approach that is used for highlighting the effectiveness of each encoding method leading to developing an efficient and reliable predictor. The t-SNE analysis provided in Figure 10 illustrates the distribution of extracted features across different feature vectors, where each panel (A–F) of Figure 10 represents the contribution of the features toward the targeted classes (NPs, non-NPs). Where the data points for the NPs class and non-NPs class are represented by gray color and green color, respectively. In the case of individual vectors, the Skip-gram5 features demonstrate moderate discrimination between the samples of both classes; however, considerable overlap remains. The ProtBERT-BFD features shown in Figure 10B appear densely mixed, suggesting limited efficacy in distinguishing between the two classes. In contrast, the ProtT5 features in Figure 10C reveal a more complex structure and class-specific features for each targeted class, forming s distinct, curved pattern; nonetheless, overlap persists. The ESM plot in Figure 10D also shows the intermixing of samples between NPs and non-NPs. While the ESM features exhibit enriched class-specific information than protBERT-BFD, they are less discriminative than skip-gram5. The hybrid vector in Figure 9E displays a more dispersed grouping of features with some small clustered regions, indicating that the hybrid features provide a broader representation of both classes but still lack entire distinctness. Figure 10F illustrates a clearer separation between the two

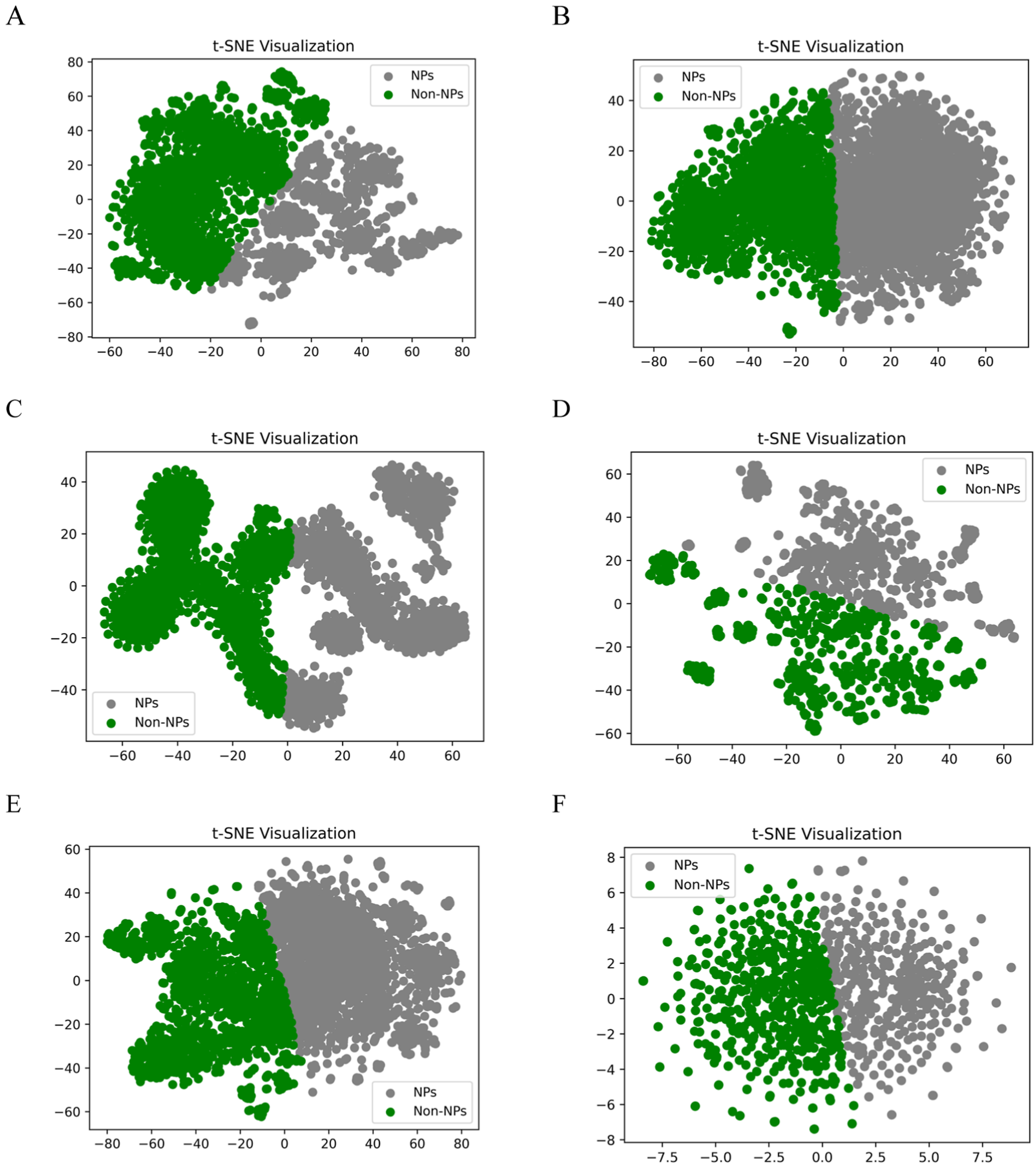


Figure 10. t-SNE analysis of training features using (A) skip-gram5 (B) ProtBERT-BFD (C) ProtT5 (D) ESM (E) hybrid features (F) selected features.

Table 4. Performance of the Classifiers Using an Independent Dataset

classifier	Acc (%)	Sen (%)	Spe (%)	MCC	AUC
CNN	87.29	83.51	91.26	0.75	0.91
DNN	78.35	76.02	80.68	0.56	0.85
RCNN	89.29	89.46	89.05	0.78	0.92
CapsNet	95.21	93.45	95.75	0.88	0.96

classes after applying a two-tier feature selection. The selected vector optimally predicts class distinctions with minor overlap, suggesting that the proposed feature selection strategy has significantly enhanced the discrimination between NPs and non-NPs.

4.4. Comparison of pNPs-CapsNet Model Work with Existing Predictors. To evaluate the efficacy and superiority of our proposed pNPs-CapsNet model in predicting neuropeptides, we assessed using a cross-validation test on the training samples and an independent test on unseen data. Our

Table 5. Comparison of the Proposed pNPs-CapsNet with Existing Models Using Training and Independent Dataset

data set	predictor	Acc (%)	Sen (%)	Spe (%)	MCC
training data set	NeuroCNN-GNB	88.70	89.10	88.30	0.77
	UniDL4BioPep	89.2	87.5	90.9	0.78
	NeuroPpred-SVM	89.50	88.60	90.40	0.79
	NeuroPred-FRL	91.90	89.50	94.30	0.84
	Akbar et al	94.47	97.32	93.81	0.91
	NeuroPred-CLQ	94.70	94	95.30	0.89
	pNPs-CapsNet	98.10	99.34	96.75	0.96
independent data set	NeuroPIpred	53.60	33.10	73.60	0.74
	PredNeuroP	89.70	88.60	90.70	0.79
	NeuroPpred-Fuse	90.60	88.20	93.00	0.81
	NeuroPred-FRL	91.10	92.70	89.50	0.82
	NeuroPpred-SVM	91.50	89.10	94	0.83
	NeuroCNN-GNB	91.80	91.90	90.70	0.79
	Akbar et al	92.55	93.84	91.23	0.87
	NeuroPred-CLQ	93.60	89.70	97.50	0.87
	pNPs-CapsNet	95.21	93.45	95.75	0.88

cross-validation was based on four different deep-learning models, as presented in Table 3, while the independent test results are provided in Table 4. The proposed CapsNet model consistently outperformed the other training models, achieving predictive accuracies of 98.10% and 95.21%, with corresponding AUC values of 0.98, and 0.96 for the training and independent samples, respectively.

To further validate the prediction of our pNPs-CapsNet model, we performed a comparative analysis with existing state-of-the-art NPs models. For the training data set, we selected several prominent predictors, including NeuroCNN-GNB,[63] UniDL4BioPep,[15] NeuroPpred-SVM,[11] NeuroPred-FRL,[9] Akbar et al.,[13] and NeuroPred-CLQ,[10] as shown in Table 5. The predictive accuracies of these models ranged from 88.70% to 94.70%, with sen values ranging from 87.10% to 94.32%, spe ranging from 88.30%−95.30%, and MCC values between 0.77 and 0.91. In contrast, our pNPs-CapsNet model significantly outperformed these methods, achieving a higher acc of 98.10%, sen of 99.34%, Spe of 96.75, and MCC value of 0.96. For the independent data set, we compared our model with the performance of several previous models, NeuroPred-FRL,[9] NeuroPpred-Fuse,[17] NeuroPIpred,[6] PredNeuroP,[7] NeuroPpred-SVM,[11] NeuroCNN-GNB,[63] Akbar et al.,[13] and NeuroPred-CLQ.[10] These models reported predictive accuracies ranging from 53.60% to 93.60%, with Sen values from 33.10 to 93.84%, and Spe values from 73.60% to 97.50%. In contrast, our pNPs-CapsNet model again outperformed these models, achieving a higher accuracy of 95.21%, with Sen, Spe, and MCC of 93.45%, 95.75, and 0.88, respectively. A comparison of our pNPs-CapsNet model with existing methods using a training data set and independent data set is illustrated in Figure 11.

5. CONCLUSION

In this study, we propose a novel deep learning-based protein language model to accurately predict neuropeptides. Addressing limitations of existing sequential encoding methods, such as the loss of residue-specific information, we introduce protein language models, including ProtBERT-BFD, ESM, and ProtT5, combined with fastText-based Skip-gram encoding to capture semantic and contextual features from peptide samples. The encoded feature sets were integrated using a Differential Evolution (DE)-based weighted feature ranking approach. A two-tier feature selection approach involving mRMD and SHAP analysis was employed to select the optimal subset from the integrated hybrid feature set. Comprehensive evaluation of various deep learning models using the selected feature set revealed that the proposed CapsNet model achieved

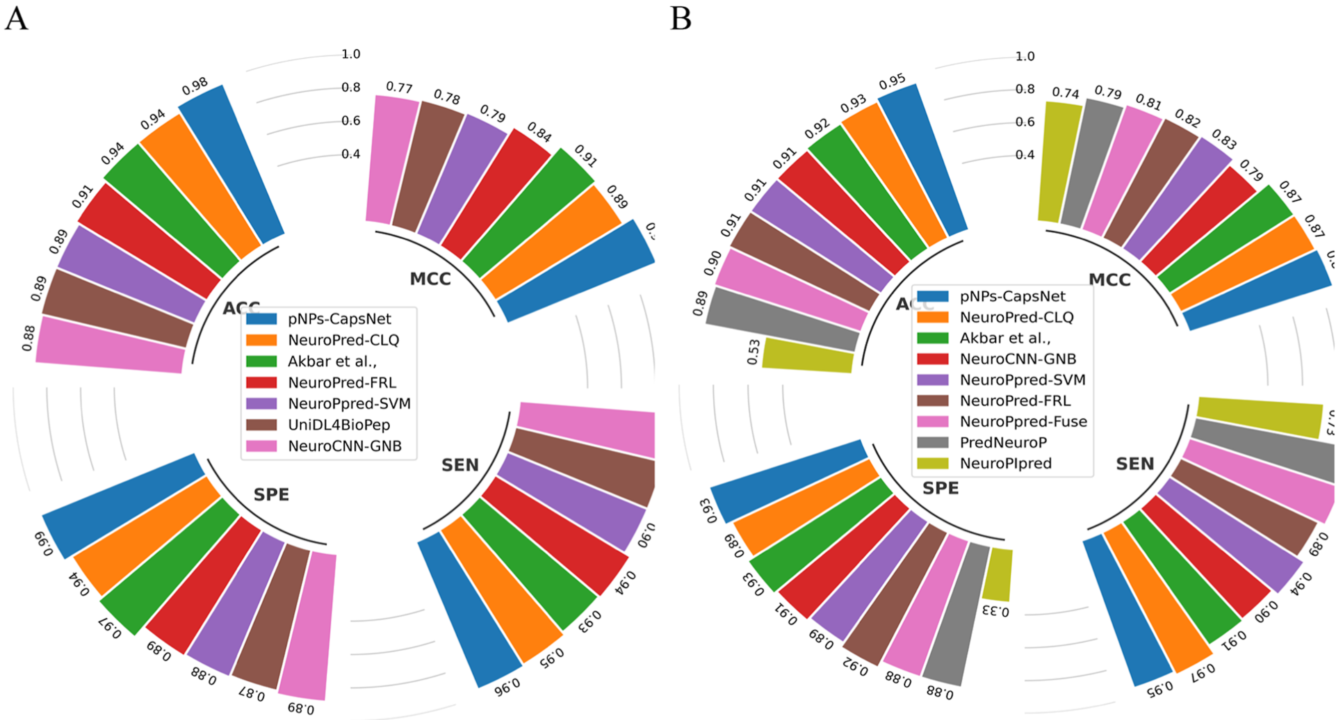


Figure 11. Comparison of our pNPs-CapsNet model with existing methods using (A) training data set and (B) independent data set.

superior performance, with an accuracy of 98.10% and an AUC value of 0.98 on the training data set. Compared to state-of-the-art models, our proposed pNPs-CapsNet model demonstrated an accuracy improvement of approximately 3.4% and an MCC enhancement of 7%. To validate the model's generalization, the pNPs-CapsNet model was tested on an independent data set, achieving an accuracy of 95.21% and an AUC value of 0.96. These findings suggest that the pNPs-CapsNet model represents a significant advancement in the predictions of NPs and has strong potential for widespread application in drug discovery and research academia.

5.1. Future Work. In our future directions, we will further investigate the prediction of neuropeptides by incorporating alternative feature encoding techniques, including novel image processing methods based on local structural and evolutionary approaches for large-scale data sets. We will also develop new optimal feature selection methods and novel generative deep learning models to enhance predictive performance.

■ ASSOCIATED CONTENT

Data Availability Statement

Data sets and codes are publicly available at https://github.com/shahidawkum/pNPs-CapsNet

■ AUTHOR INFORMATION

Corresponding Author

Quan Zou – *Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu 610054, China; Yangtze Delta Region Institute (Quzhou), University of Electronic Science and Technology of China, Quzhou 324000, PR China;* orcid.org/0000-0001-6406-1142; Email: zouquan@nclab.net

Authors

Shahid Akbar – *Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu 610054, China; Department of Computer Science, Abdul Wali Khan University Mardan, Mardan 23200 Khyber Pakhtunkhwa, Pakistan;* orcid.org/0000-0002-1106-0012

Ali Raza – *Department of Computer Science, Bahria University, Islamabad 44220, Pakistan*

Hamid Hussain Awan – *Department of Computer Science, Rawalpindi Women University, Rawalpindi 46300 Punjab, Pakistan*

Wajdi Alghamdi – *Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia*

Aamir Saeed – *Department of Computer Science and IT, University of Engineering and Technology, Jalozai Campus, Peshawar 25000, Pakistan*

Complete contact information is available at:
https://pubs.acs.org/10.1021/acsomega.4c11449

Author Contributions

SA performed Implementation, Model Creation, Methodology, and writing. AR performed Data Curation, Validation, Model Creation, and Writing. HA Performed Writing Model Validation, implementation, and writing. QZ Performed Supervision, Idea, Funding, Proof-Reading. WA Performed Formal analysis, Model visualization, and Model Interpretation. AS performed implementation, model validation, writing, and proofreading.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The authors are thankful for the support of the National Natural Science Foundation of China (no. 62131004), and the National Key R&D Program of China (2022ZD0117700).

■ REFERENCES

(1) Mendel, H. C.; Kaas, Q.; Muttenthaler, M. Neuropeptide signalling systems – An underexplored target for venom drug discovery. *Biochem. Pharmacol.* **2020**, *181*, 114129.

(2) Wang, Y.; Wang, M.; Yin, S.; Jang, R.; Wang, J.; Xue, Z.; Xu, T. NeuroPep: a comprehensive resource of neuropeptides. *Database* **2015**, *2015*, bav038.

(3) (a) Kormos, V.; Gaszner, B. Role of neuropeptides in anxiety, stress, and depression: From animals to humans. *Neuropeptides* **2013**, *47* (6), 401–419. (b) Hökfelt, T.; Broberger, C.; Xu, Z.-Q. D.; Sergeyev, V.; Ubink, R.; Diez, M. Neuropeptides — an overview. *Neuropharmacology* **2000**, *39* (8), 1337–1356.

(4) (a) Nässel, D. R.; Zandawala, M. Recent advances in neuropeptide signaling in Drosophila, from genes to physiology and behavior. *Prog. Neurobiol.* **2019**, *179*, 101607. (b) Nässel, D. R. Neuropeptides in the nervous system of Drosophila and other insects: multiple roles as neuromodulators and neurohormones. *Prog. Neurobiol.* **2002**, *68* (1), 1–84.

(5) Kang, J.; Fang, Y.; Yao, P.; Li, N.; Tang, Q.; Huang, J. NeuroPP: A Tool for the Prediction of Neuropeptide Precursors Based on Optimal Sequence Composition. *Interdiscip. Sci.:Comput. Life Sci.* **2019**, *11* (1), 108–114.

(6) Agrawal, P.; Kumar, S.; Singh, A.; Raghava, G. P. S.; Singh, I. K. NeuroPIpred: a tool to predict, design and scan insect neuropeptides. *Sci. Rep.* **2019**, *9* (1), 5129.

(7) Bin, Y.; Zhang, W.; Tang, W.; Dai, R.; Li, M.; Zhu, Q.; Xia, J. Prediction of Neuropeptides from Sequence Information Using Ensemble Classifier and Hybrid Features. *J. Proteome Res.* **2020**, *19* (9), 3732–3740.

(8) Jiang, M.; Zhao, B.; Luo, S.; Wang, Q.; Chu, Y.; Chen, T.; Mao, X.; Liu, Y.; Wang, Y.; Jiang, X.; et al. NeuroPpred-Fuse: an interpretable stacking model for prediction of neuropeptides by fusing sequence information and feature selection methods. *methods* **2021**, *22* (6), bbab310.

(9) Hasan, M. M.; Alam, M. A.; Shoombuatong, W.; Deng, H.-W.; Manavalan, B.; Kurata, H. NeuroPred-FRL: an interpretable prediction model for identifying neuropeptide using feature representation learning. *Briefings Bioinf.* **2021**, *22* (6), bbab167.

(10) Chen, S.; Li, Q.; Zhao, J.; Bin, Y.; Zheng, C. NeuroPred-CLQ: incorporating deep temporal convolutional networks and multi-head attention mechanism to predict neuropeptides. *Briefings Bioinf.* **2022**, *23* (5), bbac319.

(11) Liu, Y.; Wang, S.; Li, X.; Liu, Y.; Zhu, X. NeuroPpred-SVM: A New Model for Predicting Neuropeptides Based on Embeddings of BERT. *J. Proteome Res.* **2023**, *22* (3), 718–728.

(12) Liu, D.; Lin, Z.; Jia, C. NeuroCNN_GNB: an ensemble model to predict neuropeptides based on a convolution neural network and Gaussian naive Bayes. *Front. Genet.* **2023**, *14*, 1226905.

(13) Akbar, S.; Mohamed, H. G.; Ali, H.; Saeed, A.; Khan, A. A.; Gul, S.; Ahmad, A.; Ali, F.; Ghadi, Y. Y.; Assam, M. Identifying Neuropeptides via Evolutionary and Sequential Based Multi-Perspective Descriptors by Incorporation With Ensemble Classification Strategy. *IEEE Access* **2023**, *11*, 49024–49034.

(14) Wang, L.; Huang, C.; Wang, M.; Xue, Z.; Wang, Y. NeuroPred-PLM: an interpretable and robust model for neuropeptide prediction by protein language model. *Briefings Bioinf.* **2023**, *24* (2), bbad077.

(15) Du, Z.; Ding, X.; Xu, Y.; Li, Y. UniDL4BioPep: a universal deep learning architecture for binary classification in peptide bioactivity. *Briefings Bioinf.* **2023**, *24* (3), bbad135.

(16) Uddin, I.; Awan, H. H.; Khalid, M.; Khan, S.; Akbar, S.; Sarker, M. R.; Abdolrasol, M. G. M.; Alghamdi, T. A. H. A hybrid residue based sequential encoding mechanism with XGBoost improved ensemble model for identifying 5-hydroxymethylcytosine modifications. *Sci. Rep.* **2024**, *14* (1), 20819.

(17) Jiang, M.; Zhao, B.; Luo, S.; Wang, Q.; Chu, Y.; Chen, T.; Mao, X.; Liu, Y.; Wang, Y.; Jiang, X.; et al. NeuroPpred-Fuse: an interpretable stacking model for prediction of neuropeptides by fusing sequence information and feature selection methods. *Briefings Bioinf.* **2021**, *22* (6), bbab310.

(18) Huang, Y.; Niu, B.; Gao, Y.; Fu, L.; Li, W. CD-HIT Suite: a web server for clustering and comparing biological sequences. *Bioinformatics* **2010**, *26* (5), 680−682.

(19) Khan, Z. U.; Pi, D.; Yao, S.; Nawaz, A.; Ali, F.; Ali, S. piEnPred: a bi-layered discriminative model for enhancers and their subtypes via novel cascade multi-level subset feature selection algorithm. *Front. Comput. Sci.* **2021**, *15*, 156904.

(20) (a) Barukab, O.; Ali, F.; Khan, S. A. DBP-GAPred: an intelligent method for prediction of DNA-binding proteins types by enhanced evolutionary profile features with ensemble learning. *J. Bioinf. Comput. Biol.* **2021**, *19* (04), 2150018. (b) Raza, A.; Uddin, J.; Almuhaimeed, A.; Akbar, S.; Zou, Q.; Ahmad, A. AIPs-SnTCN: Predicting Anti-Inflammatory Peptides Using fastText and Transformer Encoder-Based Hybrid Word Embedding with Self-Normalized Temporal Convolutional Networks. *J. Complementary Integr. Med.* **2023**, *63* (21), 6537−6554.

(21) (a) Wu, C.; Gao, R.; Zhang, Y.; De Marinis, Y. PTPD: predicting therapeutic peptides by deep learning and word2vec. *BMC Bioinf.* **2019**, *20*, 456−458. (b) Zhang, Y.-F.; Wang, X.; Kaushik, A. C.; Chu, Y.; Shan, X.; Zhao, M.-Z.; Xu, Q.; Wei, D.-Q. SPVec: a Word2vec-inspired feature representation method for drug-target interaction prediction. *Front. Chem.* **2020**, *7*, 895. (c) Yao, Y.; Du, X.; Diao, Y.; Zhu, H. An integration of deep learning with feature embedding for protein−protein interaction prediction. *PeerJ* **2019**, *7*, No. e7126. (d) Sharma, R.; Shrivastava, S.; Kumar Singh, S.; Kumar, A.; Saxena, S.; Kumar Singh, R. Deep-ABPpred: identifying antibacterial peptides in protein sequences using bidirectional LSTM with word2vec. *Briefings Bioinf.* **2021**, *22* (5), bbab065. (e) Heinzinger, M.; Elnaggar, A.; Wang, Y.; Dallago, C.; Nechaev, D.; Matthes, F.; Rost, B. Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinf.* **2019**, *20*, 723.

(22) Le, N. Q. K.; Huynh, T.-T. Identifying SNAREs by incorporating deep learning architecture and amino acid embedding representation. *Front. Physiol.* **2019**, *10*, 1501.

(23) Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135−146.

(24) Nguyen, T.-T.-D.; Le, N.-Q.-K.; Ho, Q.-T.; Phan, D.-V.; Ou, Y.-Y. TNFPred Identifying tumor necrosis factors using hybrid features based on word embeddings. *BMC Med. Genomics* **2020**, *13*, 155.

(25) Le, N. Q. K.; Yapp, E. K. Y.; Ho, Q.-T.; Nagasundaram, N.; Ou, Y.-Y.; Yeh, H.-Y. iEnhancer-5Step: identifying enhancers using hidden information of DNA sequences via Chou's 5-step rule and word embedding. *Anal. Biochem.* **2019**, *571*, 53−61.

(26) Zhang, S.; Fan, R.; Liu, Y.; Chen, S.; Liu, Q.; Zeng, W. Applications of transformer-based language models in bioinformatics: a survey. *Bioinform. adv.* **2023**, *3* (1), vbad001.

(27) (a) Lee, H.; Lee, S.; Lee, I.; Nam, H. AMP-BERT: Prediction of antimicrobial peptide function based on a BERT model. *Protein Sci.* **2023**, *32* (1), No. e4529. (b) Shahid; Hayat, M.; Alghamdi, W.; Akbar, S.; Raza, A.; Kadir, R. A.; Sarker, M. R. pACP-HybDeep: predicting anticancer peptides using binary tree growth based transformer and structural feature encoding with deep-hybrid learning. *Sci. Rep.* **2025**, *15* (1), 565.

(28) Elnaggar, A.; Heinzinger, M.; Dallago, C.; Rihawi, G.; Wang, Y.; Jones, L.; Gibbs, T.; Feher, T.; Angerer, C.; Steinegger, M. ProtTrans.: Towards cracking the language of Life's code through self-supervised deep learning and high performance computing. *arXiv* **2007**, 2007.06225.

(29) Brandes, N.; Ofer, D.; Peleg, Y.; Rappoport, N.; Linial, M. ProteinBERT: a universal deep-learning model of protein sequence and function. *Bioinformatics* **2022**, *38* (8), 2102−2110.

(30) Pei, H.; Li, J.; Ma, S.; Jiang, J.; Li, M.; Zou, Q.; Lv, Z. Identification of Thermophilic Proteins Based on Sequence-Based Bidirectional Representations from Transformer-Embedding Features. *Appl. Sci.* **2023**, *13* (5), 2858.

(31) Akbar, S.; Ullah, M.; Raza, A.; Zou, Q.; Alghamdi, W. DeepAIPs-Pred: Predicting Anti-Inflammatory Peptides Using Local Evolutionary Transformation Images and Structural Embedding-Based Optimal Descriptors with Self-Normalized BiTCNs. *J. Complementary Integr. Med.* **2024**, *64* (24), 9609−9625.

(32) Vig, J.; Madani, A.; Varshney, L. R.; Xiong, C.; Socher, R.; Rajani, N. F. Bertology meets biology: Interpreting attention in protein language models. *arXiv* **2020**, 2006.15222.

(33) Pokharel, S.; Pratyush, P.; Heinzinger, M.; Newman, R. H.; Kc, D. B. Improving protein succinylation sites prediction using embeddings from protein language model. *Sci. Rep.* **2022**, *12* (1), 16933.

(34) Elnaggar, A.; Heinzinger, M.; Dallago, C.; Rehawi, G.; Wang, Y.; Jones, L.; Gibbs, T.; Feher, T.; Angerer, C.; Steinegger, M.; et al. ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44* (10), 7112−7127.

(35) (a) Rives, A.; Meier, J.; Sercu, T.; Goyal, S.; Lin, Z.; Liu, J.; Guo, D.; Ott, M.; Zitnick, C. L.; Ma, J.; et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci. U.S.A.* **2021**, *118* (15), No. e2016239118. (b) Meier, J.; Rao, R.; Verkuil, R.; Liu, J.; Sercu, T.; Rives, A. Language models enable zero-shot prediction of the effects of mutations on protein function. In *Adv. Neural Inf. Process. Syst.*, 2021; Vol. *34*, pp 29287−29303.

(36) (a) Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, 1810.04805. (b) Ullah, M.; Akbar, S.; Raza, A.; Khan, K. A.; Zou, Q. TargetCLP: clathrin proteins prediction combining transformed and evolutionary scale modeling-based multi-view features via weighted feature integration approach. *Briefings Bioinf.* **2024**, *26* (1), bbaf026.

(37) (a) Kucera, T.; Togninalli, M.; Meng-Papaxanthos, L. Conditional generative modeling for de novo protein design with hierarchical functions. *Bioinformatics* **2022**, *38* (13), 3454−3461. (b) Yadav, S.; Vora, D. S.; Sundar, D.; Dhanjal, J. K. TCR-ESM: Employing protein language embeddings to predict TCR-peptide-MHC binding. *Comput. Struct. Biotechnol. J.* **2024**, *23*, 165−173. (c) Hu, J.; Li, Z.; Rao, B.; Thafar, M. A.; Arif, M. Improving protein-protein interaction prediction using protein language model and protein network features. *Anal. Biochem.* **2024**, *693*, 115550.

(38) Suzek, B. E.; Wang, Y.; Huang, H.; McGarvey, P. B.; Wu, C. H.; the UniProt Consortium. UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics* **2014**, *31* (6), 926−932.

(39) Raza, A.; Uddin, J.; Zou, Q.; Akbar, S.; Alghamdi, W.; Liu, R. AIPs-DeepEnC-GA: Predicting Anti-inflammatory Peptides using Embedded Evolutionary and Sequential Feature Integration with Genetic Algorithm based Deep Ensemble Model. *Chemom. Intell. Lab. Syst.* **2024**, *254*, 105239.

(40) Zhou, X. G.; Zhang, G. J. Abstract Convex Underestimation Assisted Multistage Differential Evolution. *IEEE Trans. Cybern.* **2017**, *47* (9), 2730−2741.

(41) Storn, R.; Price, K. Differential Evolution − A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *J. Global Optim.* **1997**, *11* (4), 341−359.

(42) Hu, J.; Zhou, X. G.; Zhu, Y. H.; Yu, D. J.; Zhang, G. J. TargetDBP: Accurate DNA-Binding Protein Prediction Via Sequence-Based Multi-View Feature Learning. *TCBB* **2020**, *17* (4), 1419−1429.

(43) Sikander, R.; Ghulam, A.; Ali, F. XGB-DrugPred: computational prediction of druggable proteins using eXtreme gradient boosting and optimized features set. *Sci. Rep.* **2022**, *12* (1), 5505.

(44) (a) Akbar, S.; Ali, H.; Ahmad, A.; Sarker, M. R.; Saeed, A.; Salwana, E.; Gul, S.; Khan, A.; Ali, F. Prediction of Amyloid Proteins using Embedded Evolutionary & Ensemble Feature Selection based Descriptors with eXtreme Gradient Boosting Model. *IEEE Access* **2023**, *11*, 39024. (b) Ullah, M.; Akbar, S.; Raza, A.; Zou, Q. DeepAVP-TPPred: identification of antiviral peptides using transformed image-based localized descriptors and binary tree growth algorithm. *Bioinformatics* **2024**, *40* (5), btae305.

(45) Akbar, S.; Rahman, A. U.; Hayat, M.; Sohail, M. cACP: Classifying anticancer peptides using discriminative intelligent model via Chou's 5-step rules and general pseudo components. *Chemom. Intell. Lab. Syst.* **2020**, *196*, 103912.

(46) Sheikhpour, R.; Sarram, M. A.; Gharaghani, S. Constraint score for semi-supervised feature selection in ligand-and receptor-based QSAR on serine/threonine-protein kinase PLK3 inhibitors. *Chemom. Intell. Lab. Syst.* **2017**, *163*, 31−40.

(47) Meng, C.; Wu, J.; Guo, F.; Dong, B.; Xu, L. CWLy-pred: A novel cell wall lytic enzyme identifier based on an improved MRMD feature selection method. *Genomics* **2020**, *112* (6), 4715−4721.

(48) (a) Gong, Y.; Dong, B.; Zhang, Z.; Zhai, Y.; Gao, B.; Zhang, T.; Zhang, J. VTP-Identifier: Vesicular Transport Proteins Identification Based on PSSM Profiles and XGBoost. *Front. Genet.* **2022**, *12*, 808856. (b) Rukh, G.; Akbar, S.; Rehman, G.; Alarfaj, F. K.; Zou, Q. StackedEnC-AOP: prediction of antioxidant proteins using transform evolutionary and sequential features based multi-scale vector with stacked ensemble learning. *BMC Bioinf.* **2024**, *25*, 256.

(49) Chen, T.; Wang, X.; Chu, Y.; Wang, Y.; Jiang, M.; Wei, D.-Q.; Xiong, Y. T4SE-XGB: interpretable sequence-based prediction of type IV secreted effectors using eXtreme gradient boosting algorithm. *Front. Microbiol.* **2020**, *11*, 580382.

(50) (a) Kumar, C. S.; Choudary, M. N. S.; Bommineni, V. B.; Tarun, G.; Anjali, T. Dimensionality reduction based on shap analysis: a simple and trustworthy approach. In *2020 international conference on communication and signal processing (ICCSP)*; IEEE, 2020, pp 558−560. (b) Charoenkwan, P.; Schaduangrat, N.; Moni, M. A.; Shoombuatong, W. iMRSA-Fuse: A fast and accurate computational approach for predicting anti-MRSA peptides by fusing multi-view information. *IEEE ACM Trans. Comput. Biol. Bioinf* **2025**, *22*, 2−12.

(51) (a) Cao, X.; He, W.; Chen, Z.; Li, Y.; Wang, K.; Zhang, H.; Wei, L.; Cui, L.; Su, R.; Wei, L. PSSP-MVIRT: peptide secondary structure prediction based on a multi-view deep learning architecture. *Briefings Bioinf.* **2021**, *22* (6), bbab203. (b) Khanal, J.; Nazari, I.; Tayara, H.; Chong, K. T. 4mCCNN: Identification of N4-methylcytosine sites in prokaryotes using convolutional neural network. *IEEE Access* **2019**, *7*, 145455−145461. (c) Khanal, J.; Tayara, H.; Chong, K. T. Identifying enhancers and their strength by the integration of word embedding and convolution neural network. *IEEE Access* **2020**, *8*, 58369−58376. (d) Charoenkwan, P.; Chumnanpuen, P.; Schaduangrat, N.; Shoombuatong, W. Deepstack-ACE: A deep stacking-based ensemble learning framework for the accelerated discovery of ACE inhibitory peptides. *Methods* **2025**, *234*, 131−140.

(52) Lu, L.; Yi, Y.; Huang, F.; Wang, K.; Wang, Q. Integrating local CNN and global CNN for script identification in natural scene images. *IEEE Access* **2019**, *7*, 52669−52679.

(53) Dong, Z.; Lin, S. Research on image classification based on capsnet. In *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*; IEEE, 2019, pp 1023−1026.

(54) Sabour, S.; Frosst, N.; Hinton, G. E. Dynamic routing between capsules. In *Advances in neural information processing systems*, 2017; *30*

(55) Hinton, G. E.; Sabour, S.; Frosst, N. Matrix capsules with EM routing. In *International conference on learning representations*, 2018.

(56) Khanal, J.; Tayara, H.; Zou, Q.; To Chong, K. DeepCap-Kcr: accurate identification and investigation of protein lysine crotonylation sites based on capsule network. *Briefings Bioinf.* **2022**, *23* (1), bbab492.

(57) Snoek, J.; Larochelle, H.; Adams, R. P. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, 2012; *25*

(58) Nair, V.; Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning*; ICML-10, 2010, pp 807−814.

(59) Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15* (1), 1929−1958.

(60) (a) Dwivedi, A. K. Performance evaluation of different machine learning techniques for prediction of heart disease. *Neural Comput. Appl.* **2018**, *29*, 685−693. (b) Raza, A.; Uddin, J.; Akbar, S.; Alarfaj, F. K.; Zou, Q.; Ahmad, A. Comprehensive Analysis of Computational Methods for Predicting Anti-inflammatory Peptides. *Biochem. Pharmacol.* **2024**, *31* (6), 3211−3229.

(61) Akbar, S.; Zou, Q.; Raza, A.; Alarfaj, F. K. iAFPs-Mv-BiTCN: Predicting antifungal peptides using self-attention transformer embedding and transform evolutionary based multi-view features with bidirectional temporal convolutional networks. *J. Med. Artif. Intell.* **2024**, *151*, 102860.

(62) (a) Yu, S.; Liao, B.; Zhu, W.; Peng, D.; Wu, F. Accurate prediction and key protein sequence feature identification of cyclins. *Briefings Funct. Genomics* **2023**, *22*, 411. (b) Charoenkwan, P.; Chumnanpuen, P.; Schaduangrat, N.; Shoombuatong, W. Stack-AVP: A Stacked Ensemble Predictor Based on Multi-view Information for Fast and Accurate Discovery of Antiviral Peptides. *J. Mol. Biol.* **2025**, *437*, 168853.

(63) Liu, D.; Lin, Z.; Jia, C. NeuroCNN_GNB: an ensemble model to predict neuropeptides based on a convolution neural network and Gaussian naive Bayes. *Front. Genet.* **2023**, *14*, 1226905.