BMC Genomics

RESEARCH

Open Access

# Accurate prediction of virulence factors using pre-train protein language model and ensemble learning

Guanghui Li[1*], Jian Zhou[1], Jiawei Luo[2] and Cheng Liang[3*]

## Abstract

**Background** As bacterial pathogens develop increasing resistance to antibiotics, strategies targeting virulence factors (VFs) have emerged as a promising and effective approach for treating bacterial infections. Existing methods mainly relied on sequence similarity, and remote homology relationships cannot be discovered by sequence analysis alone.

**Results** To address this limitation, we developed a **p**rotein **l**anguage **m**odel and ensemble learning approach for **VF** identification (PLMVF). Specifically, we extracted features from protein sequences using ESM-2 and their three-dimensional (3D) structures using ESMFold. We calculated the true TM-score of the proteins based on their 3D structures and trained a TM-predictor model to predict structural similarity, thereby capturing hidden remote homology information within the sequences. Subsequently, we concatenated the sequence-level features extracted by ESM-2 with the predicted TM-score features to form a comprehensive feature set for prediction. Extensive experimental validation demonstrated that PLMVF achieved an accuracy (ACC) of 86.1%, significantly outperforming existing models across multiple evaluation metrics. This study provided an ideal tool for identifying novel targets in the development of anti-virulence therapies, offering promise for the effective prevention and control of pathogenic bacterial infections.

**Conclusions** The proposed PLMVF model offers an efficient computational approach for VF identification.

**Keywords** Virulence factor prediction, Protein language model, Ensemble learning, TM-score, Remote homology

## Introduction

With the rising prevalence of antibiotic resistance, bacterial infections have emerged as a major challenge in modern healthcare [1]. In this context, VFs, which are key molecules mediating pathogenic bacterial infections, enable pathogens to establish infections and cause host damage. These VFs constitute the core of pathogenicity mechanisms, determining their ability to overcome host defenses, adapt to host environments, and induce tissue damage or pathological changes [2]. Although antibiotics have long served as effective tools against bacterial infections, their overuse and misuse have exacerbated the global crisis of antimicrobial resistance [3]. Given these challenges, elucidating pathogen VFs is critical for

*Correspondence:
Guanghui Li
ghli16@hnu.edu.cn
Cheng Liang
alcs417@sdnu.edu.cn
[1] School of Information and Software Engineering, East China Jiaotong University, Nanchang 330013, China
[2] College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China
[3] School of Information Science and Engineering, Shandong Normal University, Jinan 250358, China

uncovering pathogenic mechanisms, identifying therapeutic targets, developing novel drugs, and designing vaccines, ultimately contributing to infectious disease control and treatment.

Given the critical nature of this issue, scientists have invested considerable research effort in this field. Thanks to significant advancements in DNA sequencing technology, the acquisition of bacterial genome data has become not only more efficient but also much more extensive, providing rich resources for in-depth research. As a result, several extensive and comprehensive VF databases have been developed by researchers, including VFDB [4], VICTORS [5], and MVIRDB [6]. These resources not only compile a large amount of information on VFs but also provide valuable data support for further research into the pathogenic mechanisms of bacteria. In earlier years, sequence similarity search techniques and machine learning became the main methods for identifying bacterial VFs. For sequence similarity search techniques, for example, VRprofile used HMMer [7] and BLASTp [8] to search for homologs of conserved gene clusters, aiming to identify homologous sequences of virulence-related gene clusters within query genome sequences, thereby facilitating the functional interpretation and co-localization analysis of these genes [9]. Liu et al. developed an online platform called VFanalyzer [10], which was specifically designed to identify potential VFs. When the evolutionary distance between the query sequence and its homolog was too large, sequence similarity searches might fail to accurately identify true homologous relationships. As the species divergence increases, the alignment quality decreases, and there are limitations in handling remote homologs.

To overcome this limitation, several approaches have been proposed in recent years to improve the accuracy of identifying remote homologs. In addition to traditional sequence alignment methods, techniques based on evolutionary models, such as RAxML [11], and structural alignment methods, such as TM-align [12, 13], have also demonstrated advantages when dealing with remote homologs. Specifically, the use of phylogenetic tree construction and homology inference methods can assist in the identification of VFs by capturing the conservation and structural information of sequences at long evolutionary distances.

For machine learning techniques, for instance, Sachdeva et al. designed a software program called SPAAN to classify a specific virulence factor (VF) known as adhesin, achieving high accuracy [14]. SPAAN utilized a neural network based on five distinct features for classification. To enhance the accuracy and practicality of VF prediction, the research team developed VirulentPred [15], an online platform powered by a two-level cascading Support Vector Machine (SVM) architecture. This tool integrates comprehensive VF datasets with sequence- and position-specific scoring matrix-based feature extraction methods. Model performance was further optimized through stacking strategies, while an intuitive user interface ensures accessibility for diverse research applications. Subsequently, the researchers developed an independent tool and web server called MP3 [16], which integrated SVM and HMM for large-scale genomic or metagenomic dataset predictions. Singh et al. developed a new framework (VF-Pred) for detecting VFs by analyzing genomic data [17]. This framework combined various feature engineering techniques and input them into different machine learning classification models. Among these features was a novel Seq-Alignment feature that significantly improved the model's accuracy. With the development of artificial intelligence (AI), researchers applied deep learning to identify VFs. Xie et al. proposed a deep learning (DL)-based hybrid framework called DeepVF, which used a stacking strategy to achieve more accurate VF recognition. This model utilized four traditional machine learning methods along with three DL approaches, trained 62 baseline models with features that included sequence characteristics, physicochemical properties, and evolutionary information [18]. Sun et al. proposed a novel model, DTVF [19], which integrated Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNN), incorporating an attention mechanism to significantly enhance the accuracy of VF detection. However, existing prediction models primarily rely on the contextual features of VF sequences and fail to fully consider their spatial structural information. Adhering to the principle that"sequence determines structure, and structure determines function,"understanding the spatial structure of VFs is crucial for accurately analyzing their functional types. Therefore, integrating spatial structural information can significantly enhance our understanding of VF functions and improve prediction accuracy.

For nearly half a century, researchers [20–22] have studied how to predict the 3D structure of proteins from their one-dimensional amino acid sequences. With the advancement of computational power and algorithm development, computer-based prediction methods have gradually emerged. Notably, the application of AI and machine learning technologies has brought revolutionary changes to protein structure prediction [23–26]. For example, AlphaFold2 [24] demonstrated remarkable predictive accuracy by generating high-quality 3D structural models in a short amount of time. More recently, Meta's ESMFold [26] innovatively replaced traditional multiple sequence alignment (MSA) with large language models, achieving more efficient and accurate structure

predictions, thereby significantly improving both speed and precision. This has provided a new approach to predicting VFs using structural methods. For instance, GTAE-VF [27] introduced a new model that first constructs a contact map for each protein, then extracts sequence representations from the ESMFold and ESM-2 [26] as graph inputs. The model employs an encoder-decoder architecture based on a graph transformer autoencoder, incorporating graph convolutional networks (GCN) [28] and transformers [29]. This architecture enables adaptive learning of node representations, captures long-range dependencies and latent relationships, and ultimately achieves higher accuracy than the VF-Pred model [17].

With the application of ESMFold and additional advanced language models in the field of protein structure prediction, researchers can not only rapidly obtain high-quality 3D structural models, but also perform various tasks by calculating the structural similarities of proteins, achieving promising results [30, 31]. Although traditional sequence alignment methods can identify conserved regions between homologous proteins, their effectiveness is limited for proteins with low sequence similarity but similar functions. In contrast, using high-precision 3D structural models allows for direct or indirect comparisons of protein spatial configurations, thereby helping to uncover distant relationships that are difficult to detect through sequence analysis alone [32, 33].

Building on this foundation, we design a novel binary classification model for VFs. This model first obtains the features of protein sequences and their 3D structures through ESM-2 and ESMFold [26], respectively. Then, it calculates the real TM-score based on the 3D structure of the proteins. Next, by training a structural similarity prediction model (TM-predictor) with a large dataset of known true structural similarities (TM-scores), it can effectively identify remote homology relationships hidden within the sequence. We combine the features obtained from ESM-2 [26] and the predicted TM-score. The ESM-2 features provide sequence-level contextual information, while the predicted TM-score supplements structural-level information. These combined features are then input into an ensemble model for initial prediction. Finally, a Knowledge-Augmented Network (KAN) [34] is applied to predict the VFs. After extensive experimentation, PLMVF demonstrates superior performance compared to existing models.

In summary, our key contributions are as follows:

- Existing VF prediction models solely rely on sequence similarity while neglecting structural similarity. To address this limitation, we develop a structural similarity prediction model capable of capturing remote homology information hidden behind sequences.
- We design the PLMVF framework, integrating two complementary features: ESM-based features to encode sequence-level contextual information and predicted TM-score features to incorporate structural-level insights.
- A two-stage hierarchical architecture is proposed: Feature fusion of ESM-2 sequence embeddings and TM-predictor structural features; Replacement of traditional MLP with KAN, leveraging its interpretable sparse network structure to optimize feature interactions and enhance model generalization.

## Methods

The structure of the proposed model is illustrated in Fig. 1, and the complete workflow is outlined below: 1. The features of protein sequences and their 3D structures are obtained using the ESM-2 protein language model and ESMFold, respectively. Subsequently, TM-scores are calculated from these protein structures. 2. The TM-score is predicted based on TM-predictor; 3. The integrated features are then input into an ensemble model for training; 4. Finally, KAN is used to predict VFs.

### Data collection

The study utilized a dataset established in previous research [18], containing 9,749 bacterial pathology-related VFs. The data originate from three publicly available repositories: VICTOR [5], VFDB [4], and PATRIC [35, 36]. As negative samples 66,982 non-VF samples were extracted from PBVF [37]. Clustering of both positive and negative datasets is performed using the CD-HIT program [38] with a sequence similarity threshold of 0.3, grouping similar sequences. To remove redundancy, representative sequences are chosen from each cluster, yielding a final non-redundant dataset of 3,576 VF and 4,910 non-VF sequences. For dataset balancing, 3,000 VFs and 3,000 non-VFs are selected as the training set, while 576 VFs and 576 non-VFs are designated as the test set. Figure 2 depicts the length distributions of protein sequences in the training and test sets.

### Protein sequence feature extraction

To harness the representational power of protein language models for feature extraction, the ESM-2 model was selected to derive sequence embeddings. ESM-2 employs a 33-layer transformer architecture, where each layer integrates multi-head self-attention mechanisms, feedforward neural networks, layer normalization, and residual connections. This deep architecture enables comprehensive modeling of long-range dependencies in protein sequences, capturing critical residue-residue

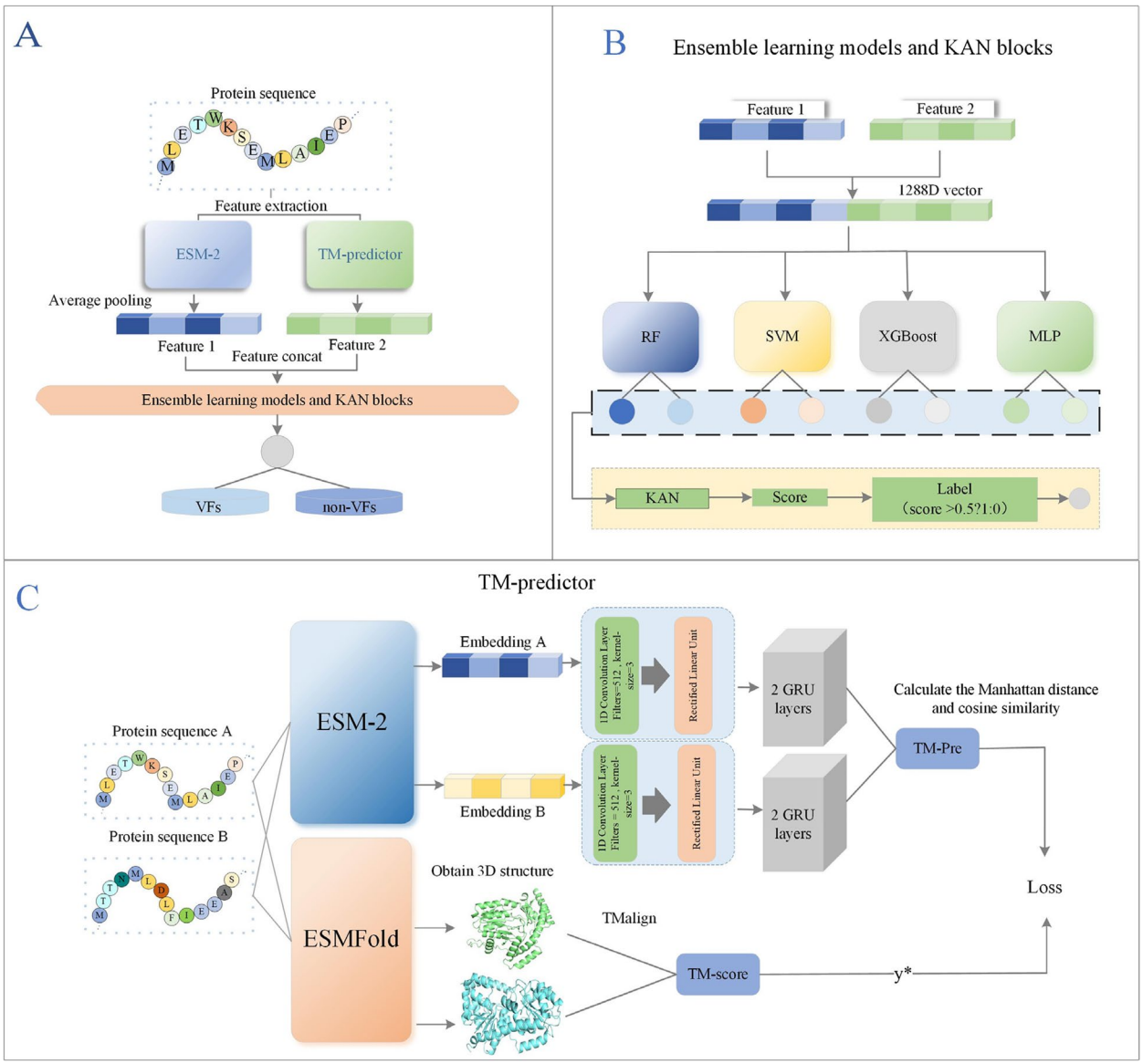


**Fig. 1** The architecture of the PLMVF

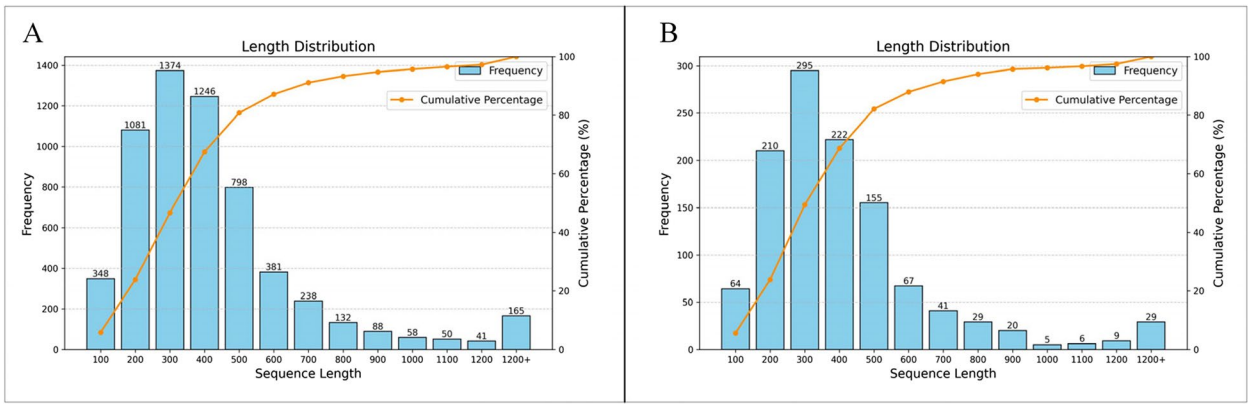


**Fig. 2** The length distribution of the training set (**A**) and test set (**B**)

interactions essential for predicting structural elements, functional domains, and catalytic sites [39]. Input amino acid sequences are processed by the pre-trained model to generate per-residue 1280-dimensional feature vectors. Global protein representations are subsequently obtained via average pooling, yielding a consolidated $1\times 1280$ feature vector per protein ($X_{ESM}$).

#### Obtaining protein 3D Structures and TM-score

ESMFold enables rapid and relatively accurate protein structure prediction. Therefore, we use ESMFold to predict the 3D structures of proteins. After obtaining these 3D structures, we use TM-align to compute the TM-scores between every pair of proteins to assess the structural similarity.

TM-align is a structural alignment algorithm used to compare the 3D structures of two proteins and assess their similarity through the TM-score [13]. The TM-score is a length-independent metric that, compared to the traditional Root-Mean-Square Deviation, better reflects global structural similarity of proteins. Formula used by TM-align to compute the TM-score is:

$$TM-score = max\left(\frac{1}{L_{target}}\sum_{i=1}^{L_{aligned}}\frac{1}{1+\left(\frac{d_i}{d_0(L_{target})}\right)^2}\right), \tag{1}$$

where $L_{target}$ represents the length of the target protein, $L_{aligned}$ denotes the count of aligned residues, $d_i$ is the distance between the $i$ pair of corresponding residues, and $d_0(L_{target})$ serves as a scaling factor dependent on length. TM-align optimizes the rotation and translation of the two protein structures to minimize the distance between corresponding residue pairs, yielding the highest possible TM-score. This algorithm is widely used in structural biology to assess protein fold similarity and evolutionary relationships.

#### TM-score predictor

To rapidly predict remote homology between proteins, we propose a TM-predictor model that relies solely on protein embeddings to directly estimate the TM-score, which serves as a measure of structural similarity. This approach avoids the complexity associated with traditional structure-based computations. The model first obtains feature embeddings for each protein through ESM-2, which are then input into the TM-predictor. The model uses CNN [40] to process the features and input them into the Gated Recurrent Unit (GRU) [41] for the next processing. Finally, the Manhattan distance and cosine similarity for each protein pair are computed and combined to approximate the true TM-score, the structure of the model is as follows:

First, the protein features obtained from ESM-2 are fed into the model. Then, a 1D CNN [40] layer is used to capture local features. The CNN is defined as follows:

$$X^{(1)} = Conv1D\left(X^{(0)}, W_0\right), \tag{2}$$

where $W_0$ is a learnable parameter and $X^{(0)}$ is the feature obtained by ESM-2. After the 1D convolution, the output is processed using the ReLU activation function. The activation function is defined as follows:

$$X^{(2)} = ReLU\left(X^{(1)}\right), \tag{3}$$

The processed features are then fed into a two-layer GRU [41] to capture the global dependencies of the sequence. The GRU is defined as follows:

$$X^{(3)} = GRU\left(X^{(2)}\right), \tag{4}$$

The GRU controls the update and forgetting of information through the reset gate and the update gate, after processing, the model computes the Manhattan distance and cosine similarity between two protein sequences to measure their structural similarity. The formulas for Manhattan distance and cosine similarity are defined as follows:

$$d_{Manhattan}(A,B) = \sum_{i=1}^{h}\left|X_{A,i}^{(3)} - X_{B,i}^{(3)}\right|, \tag{5}$$

$$cosine(A,B) = \frac{X_A^{(3)}.X_B^{(3)}}{\left\|X_A^{(3)}\right\|\left\|X_B^{(3)}\right\|}, \tag{6}$$

where $X_A^{(3)}$ and $X_B^{(3)}$ represent the features of two protein sequences. By concatenating these two distance measures, a linear weighting function is applied to approximate the true TM-score. The predicted TM-score is defined as follows:

$$TM_{pre}(A,B) = w_1(d_{manhattan}(A,B) + cosine(A,B)) + b, \tag{7}$$

where $w_1$ is a learnable weight and $b$ is a bias term. We use the true TM-score as the supervision signal and adopt the MSE loss function to minimize the difference between the predicted values and the true values. The MSE loss function is defined as follows:

$$MSE_{(TM_{pre},y)} = \frac{1}{N}\sum_{i=1}^{N}\left(TM_{pre}^{(i)} - y^i\right)^2, \tag{8}$$

Here, $TM_{pre}^{(i)}$ and $y^i$ represent the predicted value and the true value for $i$ sample, respectively.

Ultimately, for each protein, we predict its TM-score with both positive and negative samples separately, and then select the top $k$ highest structure similarity scores from each group. These scores are combined to form the final feature ($X_{TM}$) for that protein.

### Ensemble model

In this study, we concatenate the obtained $X_{ESM}$ and $X_{TM}$ features and input them into our ensemble model. To evaluate model performance, we employ a stacking approach, which enhances overall effectiveness by integrating predictions from multiple base learners. Compared to traditional single classifiers, stacking models are more effective at capturing complex patterns and features in the data. In our stacking model, multiple base learners (RF [42], SVM [43], XGBoost, and MLP) first independently train and make predictions on the input samples. Each base learner generates a probability distribution, representing the relative likelihood of each class. When generating the final prediction, we combine the probability outputs from all base learners. By stacking the probabilities of different base learners, we are able to more comprehensively utilize the unique features of each model, improving the robustness and accuracy of the model across different data samples. After stacking the features from all the base learners, we train them with a KAN, and the resulting probabilities are used as the final prediction. This method harnesses the advantages of multiple models to enhance prediction accuracy on complex datasets.

### KAN

By employing various machine learning models, we extract a set of two-dimensional features from each model. These features are concatenated to form an input feature matrix $X \in R^{n\times d}$, where $n$ represents the number of sample types and $d$ represents the total dimension of the combined features. The resulting feature matrix encapsulates the collective knowledge of multiple models, serving as the input for the KAN. The KAN is responsible for learning the complex mapping between the stacked features and the target outputs. According to the Kolmogorov representation theorem, any multivariate continuous function $f$ can be expressed as a sum of univariate functions. For the stacked feature matrix $X$, it can be expressed as:

$$f(x_1, x_2..., x_d) = \sum_{i=1}^{2d+1} \Psi_i \left( \sum_{j=1}^{d} \Phi_{ij}(X_j) \right), \tag{9}$$

where $\Psi_i, \Phi_{ij}$ are continuous univariate functions, and $x_1, x_2..., x_d$ represents the input features. The stacked features $X$ are transformed through a series of univariate functions $\Psi_i$, and then combined nonlinearly through $\Phi_{ij}$, Enabling the model to capture complex interactions among the input features. The output expression of the KAN is given by:

$$y = \sum_{i=1}^{k} \Psi_i \left( \sum_{j=1}^{d} w_{ij}.\Phi_{ij}(X_j + b_i) \right), \tag{10}$$

where $y$ is the final predicted output, $w_{ij}$ represents the learned weights, $\Phi_{ij}$ is the univariate activation function applied to each feature $X_j$. $\Psi_i$ is the nonlinear activation function in the output layer, and $b$ is the bias term.

The KAN module was trained using standard backpropagation and optimized with the Adam algorithm. Overall, integrating KAN into our framework effectively modeled the stacked ensemble features, enabling better generalization in complex feature spaces and enhancing accuracy. We provide a succinct and systematic exposition of our proposed model, as shown in Algorithm 1.

Algorithm 1: The pseudocode of PLMVF

```
Input: Protein sequences of positive and negative samples;
Output: Predicted sequence label ŷ;
Extract sequence features using ESM-2 → X_seq
Construct 3D structure using ESMFold → Structure_db
Calculate real TM-score using Eq. (1) → real_TM
Train TM-predictor:
  Input: (protein_i, protein_j) pairs with real_TM and sequence features
  Train model to predict TM-score from (X_seq_i, X_seq_j)
for n = 1 → N do
    for each protein in Positive_Set do
        Predict TM-score using TM-predictor → TM_pos
    end for
    for each protein in Negative_Set do
        Predict TM-score using TM-predictor → TM_neg
    end for
    Select top k structure similarity scores from TM_pos and TM_neg
end for
Concatenate X_seq and X_TM → Combined_Feature
for i = 1 → Epoch do
    Obtain ensemble output: EnsembleModel (Combined_Feature) → Z
    Calculate loss and update Z by Adam optimizer
end for
Output: Predicted label ŷ using Eq. (10)
```

## Results

### Model evaluation metrics

In this study, we used seven metrics to evaluate the model performance: AUC, AUPR, F1- score, Accuracy, Recall, Specificity, and Precision. The calculation was performed as follows:

$$AUC = 0.5 \times \sum_{i=1}^{n} \left(x_{ROC,i} - x_{AOC,i-1}\right) \times \left(y_{AOC,i-1} + y_{AOC,i}\right), \tag{11}$$

$$AUPR = 0.5 \times \sum_{i=1}^{n} (x_{PR,i} - x_{PR,i-1}) \times (y_{PR,i-1} + y_{PR,i}), \quad (12)$$

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}, \quad (13)$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}, \quad (14)$$

$$Recall = \frac{TP}{TP + FN}, \quad (15)$$

$$Specificity = \frac{TN}{TN + FP}, \quad (16)$$

$$Precision = \frac{TP}{TP + FP}, \quad (17)$$

where $x_{Roc}$ and $y_{Roc}$ are the sequences of *FPR* and *TPR*, $x_{PR}$ and $y_{PR}$ are the sequences of *Recall* and *Precision*, *TP* denotes true positives, *TN* denotes true negatives, *FP* denotes false positives, and *FN* denotes false negatives.

#### Experimental settings

In our experiments, the dataset was divided into a training set and an independent test set. The training process employed tenfold cross-validation, where model performance was evaluated on the validation set after each fold. Detailed results are reported in Table 1. Our model achieved an average accuracy of 0.889 on the validation sets. Notably, as shown in Fig. 3, the average AUC and AUPR reached 0.948 and 0.942, respectively. These results demonstrate that the proposed model exhibits excellent performance in the VF prediction task.

The hyperparameters were set as follows: for the TM-predictor module, the batch size was set to 100, the learning rate was set to 1e-6, and the number of training epochs was set to 50. For KAN module, the grid size was set to 3, and the noise scale was set to 0.1. The TM-score feature dimension was 8. Table 2 presented the detailed configuration of all hyperparameters, where the optimal settings were retained. The selection criterion for choosing all optimal hyperparameters was based on accuracy during tenfold cross-validation.

#### Impact of TM-score dimensions on model performance

TM-score was a key metric for measuring the structural similarity of proteins, and its dimensionality significantly affected the model's performance. To systematically assess how different TM-score dimensions influenced model performance, we conducted experiments on the same independent test set, testing the model's performance as the TM-score dimension $k$ varied from 1 to 10. For each protein, we obtained the predicted TM-scores from both positive and negative samples, and then selected the top $k$ highest structural similarity scores. As shown in Table 3, when $k$= 4, the model achieved the highest accuracy, indicating the optimal performance.

#### Comparison of performance across different methods

To evaluate the effectiveness of PLMVF, we compared it with existing sequence-based and structure-based models. All models were evaluated on the same dataset under consistent experimental settings. The sequence-based models included CNN, GRU, LSTM, and Transformer, while the structure-based models included GCN and GAT. Table 4 listed the parameter configurations of each model, and Table 5 presented the performance comparison on the final independent test set.

Among all models, PLMVF consistently achieved the best results across all evaluation metrics. In particular, in terms of accuracy, PLMVF outperformed CNN, GRU, LSTM, Transformer, GCN, and GAT by 3.91%, 2.52%, 3.07%, 3.56%, 2.49%, and 4.03%, respectively. Compared

**Table 1** Results of PLMVF on tenfold cross-validation

| Fold | AUC | AUPR | Accuracy | F1-socre | Recall | Specificity | Precision |
|---|---|---|---|---|---|---|---|
| 1 | 0.943 | 0.943 | 0.888 | 0.889 | 0.89 | 0.887 | 0.887 |
| 2 | 0.942 | 0.930 | 0.885 | 0.891 | 0.937 | 0.833 | 0.849 |
| 3 | 0.952 | 0.945 | 0.898 | 0.898 | 0.903 | 0.893 | 0.894 |
| 4 | 0.938 | 0.943 | 0.873 | 0.878 | 0.91 | 0.837 | 0.848 |
| 5 | 0.952 | 0.95 | 0.895 | 0.90 | 0.943 | 0.847 | 0.860 |
| 6 | 0.945 | 0.941 | 0.87 | 0.875 | 0.907 | 0.833 | 0.845 |
| 7 | 0.949 | 0.928 | 0.888 | 0.892 | 0.923 | 0.853 | 0.863 |
| 8 | 0.951 | 0.951 | 0.898 | 0.903 | 0.943 | 0.853 | 0.865 |
| 9 | 0.943 | 0.923 | 0.883 | 0.884 | 0.887 | 0.88 | 0.881 |
| 10 | 0.961 | 0.965 | 0.912 | 0.913 | 0.927 | 0.897 | 0.90 |
| Average | 0.948 | 0.942 | 0.889 | 0.892 | 0.917 | 0.861 | 0.869 |

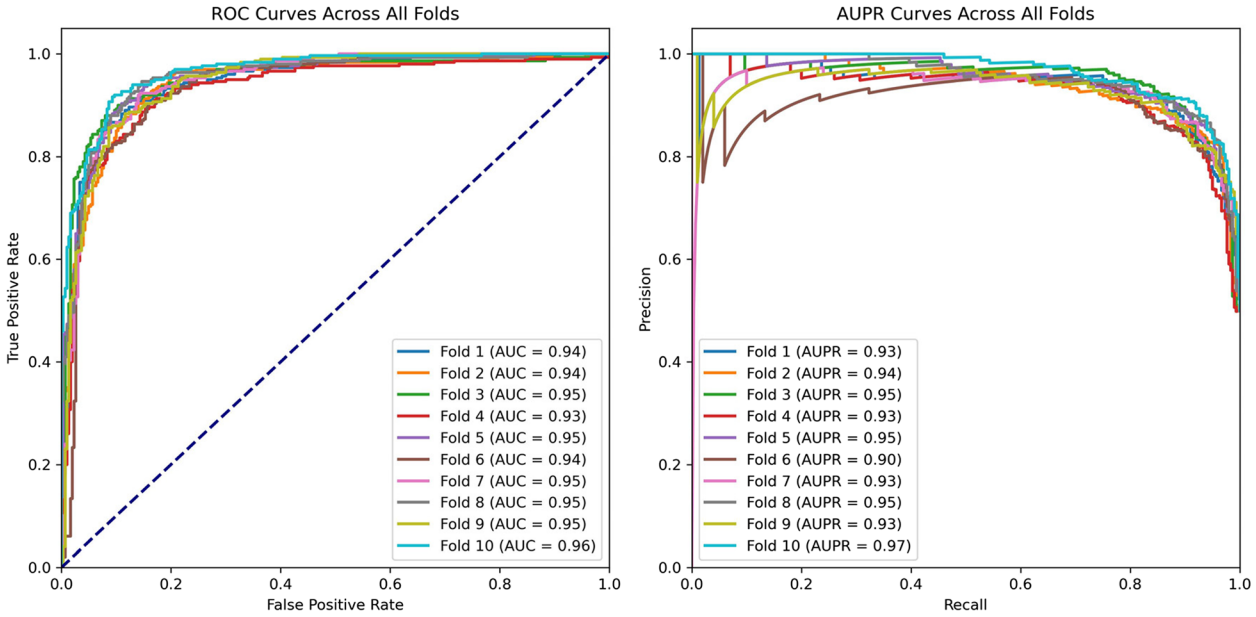


**Fig. 3** ROC curves and PR curves of PLMVF on tenfold cross-validation

**Table 2** Hyperparameter settings of PLMVF

| | Hyperparameter | Setting |
|---|---|---|
| TM-predictor | bitch size | 100 |
| | epoch | 50 |
| | learning_rate | 1e-6 |
| | hidden_dim | 512 |
| | Seed | 42 |
| KAN | width | [8, 1],[8, 2, 1], [8, 4, 1], [8, 4, 2, 1] |
| | grid | [1, 2, 3, 4, 5, 6, 7] |
| | noise_scale | [0.1, 0.2, 0.3, 0.4, 0.5, 0.6 |
| | seed | 42 |
| | steps | [10, 20, 30, 40, 50, 60, 70] |
| $X_{TM}$ | dimension | [2, 4, 6, **8**, 10, 12, 14, 16, 18, 20] |

to traditional sequence-based models, PLMVF made better use of structural information and was able to capture remote homology features hidden within sequences. Among the structure-based models, GCN achieved the second-highest accuracy on the test set, surpassing all sequence-based methods, which further confirmed the importance of structural information in protein function prediction. However, GCN relied solely on structural features and lacked the ability to comprehensively represent sequence-level characteristics. In comparison to GCN, PLMVF achieved higher accuracy, highlighting the effectiveness of integrating multiple modalities.

### Comparison with existing models

To assess the performance of the PLMVF model, we conducted a comparison against several models commonly

**Table 3** Performance of different TM-score dimensions on independent test sets

| *k* | Accuracy | AUC | AUPR | F1-score | Recall | Specificity | Precision |
|---|---|---|---|---|---|---|---|
| 1 | 0.8481 | 0.9111 | 0.9048 | 0.8508 | **0.8663** | 0.8299 | 0.8358 |
| 2 | 0.8516 | 0.9193 | 0.9273 | 0.8527 | 0.8594 | 0.8438 | 0.8462 |
| 3 | 0.8498 | **0.9198** | 0.9236 | 0.8515 | 0.8611 | 0.8385 | 0.8421 |
| 4 | **0.8611** | 0.9172 | 0.9223 | **0.8589** | 0.8455 | 0.8767 | 0.8728 |
| 5 | 0.8594 | 0.9043 | 0.8949 | 0.8554 | 0.8316 | **0.8872** | **0.8805** |
| 6 | 0.8438 | 0.9187 | **0.9294** | 0.8435 | 0.8420 | 0.8459 | 0.8449 |
| 7 | 0.8472 | 0.9034 | 0.8775 | 0.8472 | 0.8472 | 0.8472 | 0.8472 |
| 8 | 0.8463 | 0.9153 | 0.9209 | 0.8488 | 0.8628 | 0.8299 | 0.8353 |
| 9 | 0.8438 | 0.9152 | 0.9190 | 0.8451 | 0.8524 | 0.8350 | 0.8379 |
| 10 | 0.8490 | 0.9077 | 0.8987 | 0.8492 | 0.8506 | 0.8472 | 0.8478 |

**Table 4** Hyperparameter settings of baselines

| Methods | Constitution | Setting |
|---|---|---|
| CNN | one CNN layer, one max_pool layer, two MLP layers; | |
| GRU | one GRU layer, one MLP layer; | |
| LSTM | one LSTM layer, one MLP Layer; | learning_rate = 0.001, opt = Adam, |
| Transformer | two Transformer layers, four transformer heads, two MLP layers; | loss = CrossEntropyLoss, seed = 26, |
| GCN | two GCN layers | |
| GAT | two GAT layers | |

**Table 6** Performance comparison of PLMVF with existing methods on the identical independent test set

| | Accuracy | F1-score | Specificity | Sensitivity |
|---|---|---|---|---|
| BLAST | 0.750 | 0.732 | 0.818 | 0.682 |
| MP3 | 0.660 | 0.612 | 0.783 | 0.536 |
| PBVF | 0.794 | 0.790 | 0.814 | 0.774 |
| VirulentPred | 0.607 | 0.620 | 0.573 | 0.641 |
| DeepVF | 0.812 | 0.807 | 0.833 | 0.790 |
| VF-Pred | 0.835 | 0.825 | 0.760 | 0.870 |
| DTVF | 0.845 | 0.839 | **0.889** | 0.802 |
| GTAE-VF | 0.849 | 0.834 | 0.879 | 0.814 |
| PLMVF | **0.861** | **0.859** | 0.877 | **0.873** |

used in the field. The selected models included BLAST [44], MP3 [16], PBVF [37], VirulentPred [15], DeepVF [18], VF-Pred [17], DTVF [19], and GTAE-VF [27]. We conducted the comparison with other recent binary classification models for VFs on the same independent test set. As presented in Table 6 and Fig. 4, our model achieved higher accuracy scores than the other eight models, with accuracy improvements of 11.1%, 20.1%, 6.7%, 25.4%, 4.9%, 2.6%, 1.5%, and 1.2%, respectively. These significant performance gains clearly demonstrated the superior predictive capability of PLMVF. By integrating sequence features extracted from ESM-2 with predicted TM-score structural features, PLMVF effectively captured complementary information from both modalities, thereby enhanced its discriminative power. As a result, PLMVF will serve as an efficient and reliable tool for VF identification.

### Ablation study

To explore the impact of each component on PLMVF's predictive performance, we conducted a comprehensive ablation study. We trained and tested the model using the same dataset, dividing the experiments into three kinds of variant: removing the predicted TM-scores (PLMVF w/o TM), removing the ESM-2 features (PLMVF w/o ESM), and removing the KAN classifier (PLMVF w/o KAN). The detailed results were presented in Table 7. As shown in the table, removing the predicted TM-score features led to a decrease in accuracy to 0.834 for the PLMVF w/o TM model, representing a 2.7% drop compared to the full PLMVF model. This indicated that the TM-score features provided valuable structural information to the model. When the ESM-2 sequence features were removed, all evaluation metrics dropped below those of the original PLMVF model, indicating that structural information alone was insufficient to fully capture protein characteristics. Additionally, replacing the KAN classifier with a simpler prediction module resulted in slightly lower performance across all metrics for PLMVF w/o KAN. This suggested that although the major improvements were driven by the integrated features, the KAN module also played an important role in refining classification boundaries and boosting overall model performance.

To more intuitively illustrate the effect of each component on VF recognition, we used t-SNE to visualize the positive and negative sample distributions in the independent test set. As illustrated in Fig. 5, the complete PLMVF model was significantly more effective in distinguishing between two classes of samples. Overall, the above results strongly supported the effectiveness of multimodal feature integration and validated the

**Table 5** Performance of different methods on an independent test set

| | Accuracy | AUC | AUPR | F1-score | Recall | Specificity | Precision |
|---|---|---|---|---|---|---|---|
| CNN | 0.8220 | 0.8909 | 0.9001 | 0.8151 | 0.7847 | 0.8594 | 0.8480 |
| GRU | 0.8359 | 0.9114 | 0.9192 | 0.8344 | 0.8264 | 0.8455 | 0.8425 |
| LSTM | 0.8307 | 0.9121 | 0.9201 | 0.8303 | 0.8281 | 0.8333 | 0.8325 |
| Transformer | 0.8255 | 0.8909 | 0.8987 | 0.8248 | 0.8212 | 0.8299 | 0.8284 |
| GCN | 0.8412 | 0.9113 | 0.9189 | 0.8397 | 0.8321 | 0.8502 | 0.8474 |
| GAT | 0.8258 | 0.8990 | 0.9111 | 0.8218 | 0.8032 | 0.8484 | 0.8412 |
| PLMVF | **0.8611** | **0.9172** | **0.9223** | **0.8589** | **0.8455** | **0.8767** | **0.8728** |

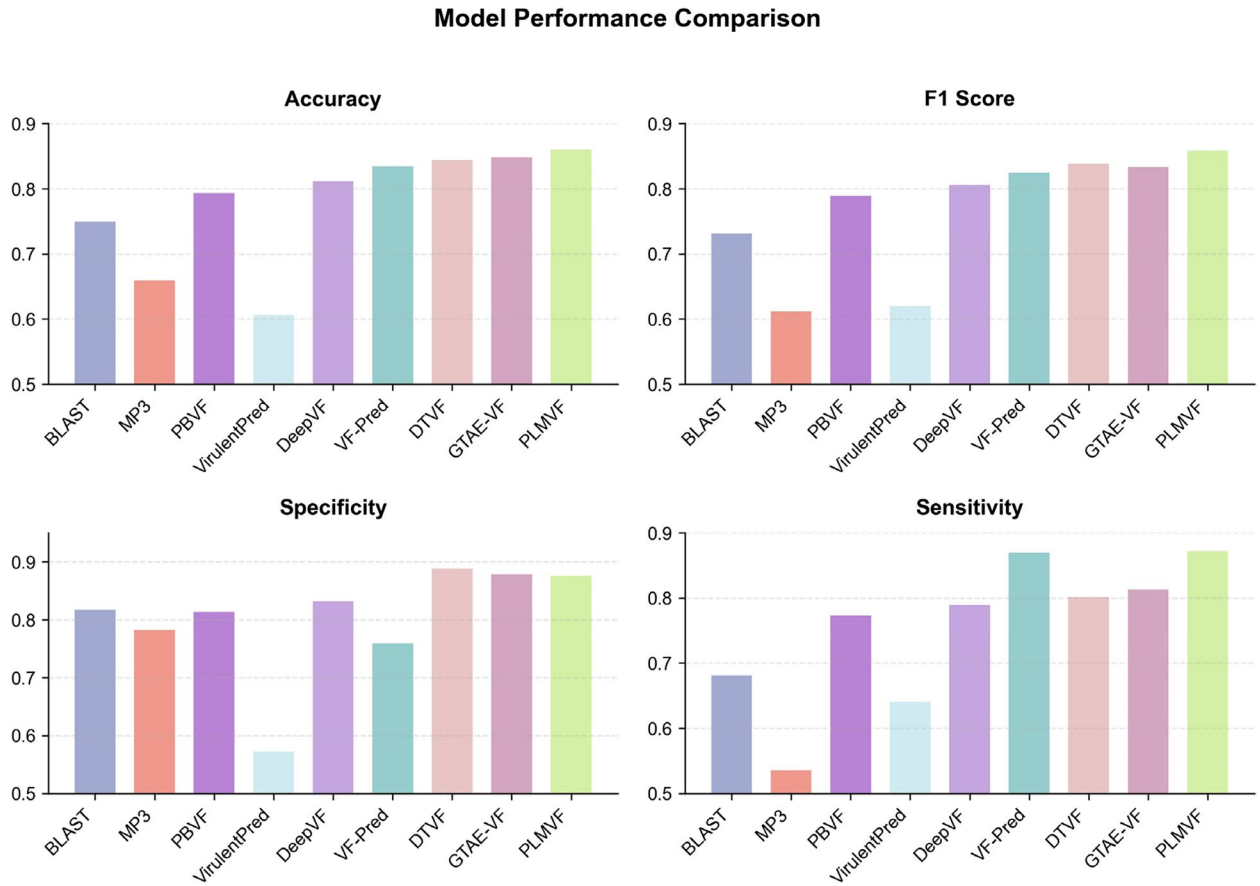


**Fig. 4** Performance comparison of PLMVF with different models on the same independent test set

**Table 7** Ablation study of PLMVF on the independent test set

| | Accuracy | AUC | AUPR | F1-score | Recall | Specificity | Precision |
|---|---|---|---|---|---|---|---|
| PLMVF w/o TM | 0.834 | 0.902 | 0.909 | 0.834 | 0.833 | 0.835 | 0.835 |
| PLMVF w/o ESM | 0.811 | 0.834 | 0.793 | 0.811 | 0.813 | 0.809 | 0.810 |
| PLMVF w/o KAN | 0.852 | 0.913 | 0.915 | 0.853 | **0.847** | 0.863 | 0.861 |
| PLMVF | **0.861** | **0.917** | **0.922** | **0.859** | 0.846 | **0.877** | **0.873** |

advantage of including the KAN classifier, confirmed the superiority of the full PLMVF framework for VF prediction.

### Impact of classifiers on the model

To analyze the effect of classifier selection on protein property prediction accuracy, we conducted experiments using various classification algorithms. Specifically, we applied different classifiers, including SVM, RF, Logistic Regression (LR), and MLP, to the same independent test set. Additionally, we used the classifier employed by the model (PLMVF), which was the KAN classifier. By keeping the feature extraction module unchanged, we evaluated the aforementioned classifiers on the independent test set to isolate the effect of the classifier itself on the final model performance. Table 8 presented a comparison of the performance of different classifiers. The KAN-based model attained an accuracy of 0.861 on the test set, outperforming SVM by 2.9%, RF by 3.5%, LR by 2.4%, and MLP by 2.2%. Furthermore, we also evaluated the performance of each classifier on the validation set, with Fig. 6 showing the AUC of the KAN, MLP, SVM, RF, and

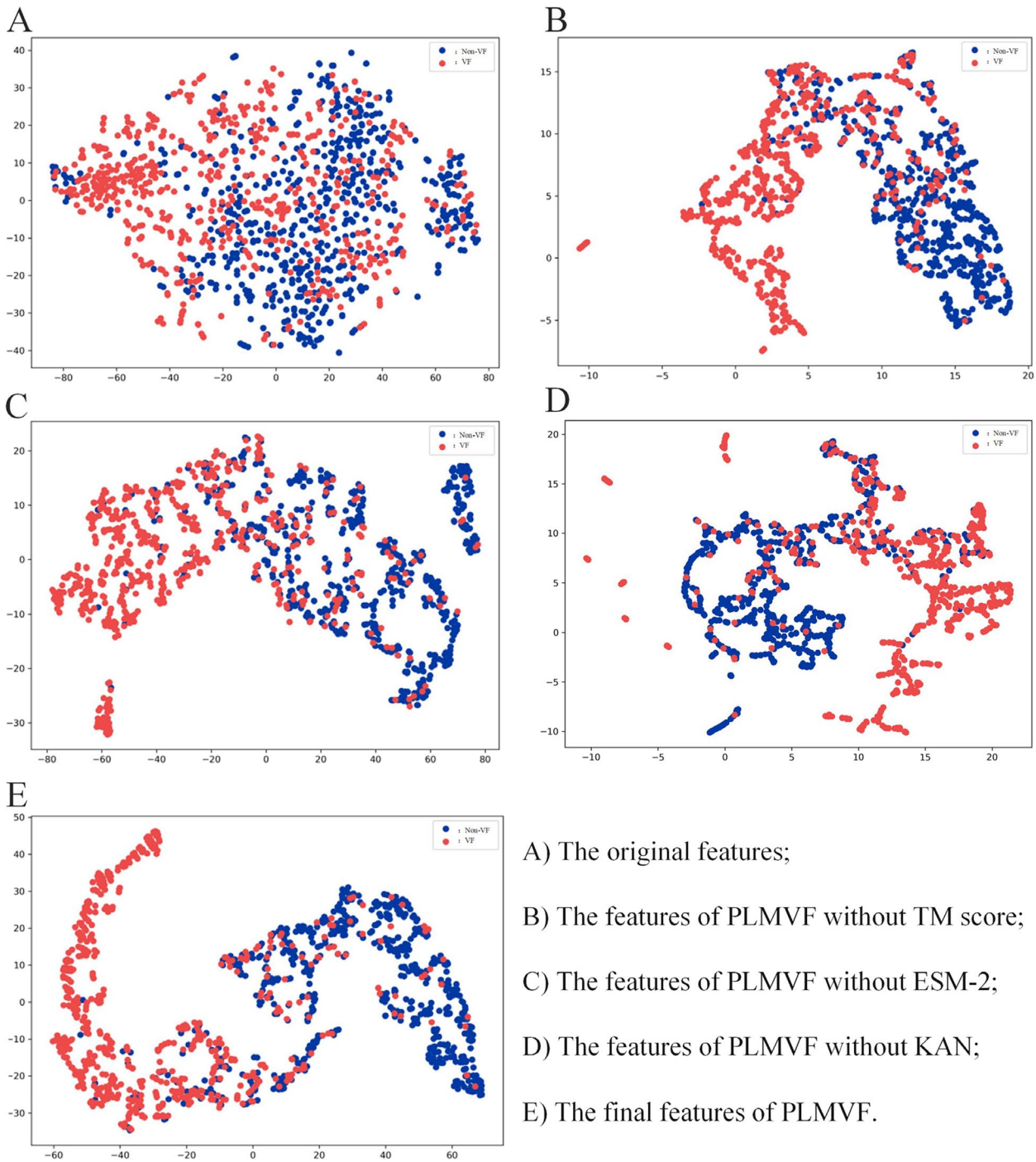


**Fig. 5** The visualization of each module feature of PLMVF using t-SNE

LR classifiers during validation. On the validation set, the KAN classifier once again demonstrated a significant advantage, achieving the highest average AUC of 0.948 among all classifiers. This indicates that the KAN classifier maintained consistently high performance and outperformed the next best classifier, LR, by about 7% in AUC.

By comparing the results on the validation and test sets, it could be observed that the KAN classifier performed robustly on both sets, without significant overfitting. In contrast, other classifiers such as MLP, SVM, RF, and LR generally performed worse on the validation set compared to the test set. KAN classifier not only achieved the

**Table 8** Performance of different classifiers on an independent test set

| | Accuracy | AUC | AUPR | F1-score | Recall | Specificity | Precision |
|---|---|---|---|---|---|---|---|
| SVM | 0.832 | 0.832 | 0.874 | 0.830 | 0.825 | 0.839 | 0.836 |
| RF | 0.826 | 0.826 | 0.871 | 0.824 | 0.813 | 0.840 | 0.836 |
| LR | 0.837 | 0.837 | 0.878 | 0.837 | 0.835 | 0.839 | 0.838 |
| MLP | 0.839 | 0.839 | 0.880 | 0.837 | 0.830 | 0.847 | 0.845 |
| KAN | **0.861** | **0.917** | **0.922** | **0.859** | **0.846** | **0.877** | **0.873** |

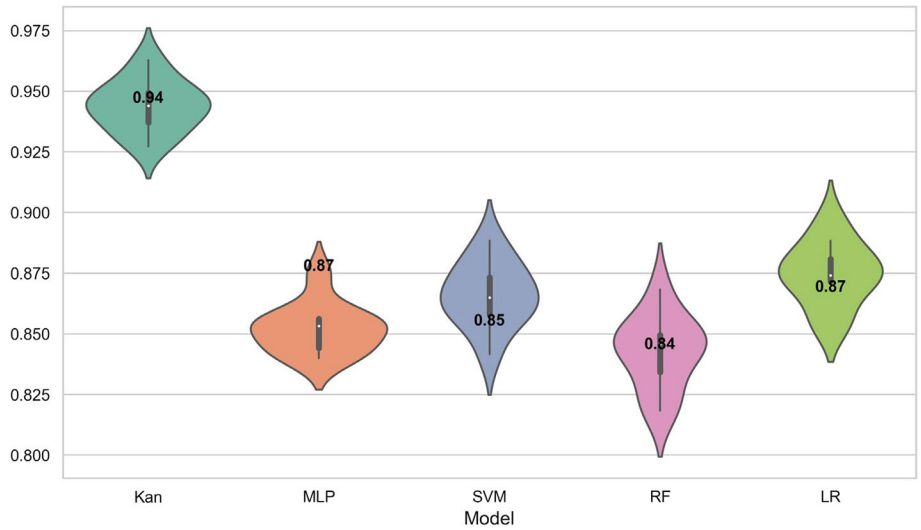


**Fig. 6** Performance comparison of different classifiers on the validation set

best performance on the test set but also demonstrated excellent generalization and stability on the validation set, further proved its position as the preferred classifier for VF identification tasks.

#### Impact of different ensemble methods on the model

This study investigated different ensemble techniques, including stacking methods, voting methods, and boosting methods, to improve the model's accuracy.

Stacking methods constructed a more powerful prediction model by combining the prediction results of multiple different base models. Stacking methods not only enhanced model performance but also leveraged the advantages of different types of models to improve prediction stability and accuracy.

Hard voting improved the overall model performance by combining the prediction results of multiple classifiers. In hard voting, each base classifier generated a prediction for the input sample, with the final result determined by the majority class, or mode, across all classifiers.

Soft voting improved overall model performance by combining the predicted probabilities of multiple classifiers rather than directly combining predicted classes. In soft voting, each base classifier produced a predicted likelihood for the given sample, with the final prediction determined by the weighted or simple average of these values.

Boosting methods sought to enhance the overall model's performance and robustness by integrating multiple weak classifiers. Boosting methods adjusted the model's focus on misclassified samples during training, effectively reducing bias and improving classifier performance.

Figure 7 depicted the performance of various ensemble models. In all metrics, stacking methods outperformed the other ensemble methods, demonstrating that stacking methods achieved the highest accuracy in VF identification. These results indicated that stacking methods had a significant advantage in integrating the predictions of multiple base models, effectively enhancing model performance.

#### PLMVF accurately detects remote homology pairs

Homologous proteins with low sequence similarity but high structural similarity were typically considered as remote homologous pairs, where sequence identity was

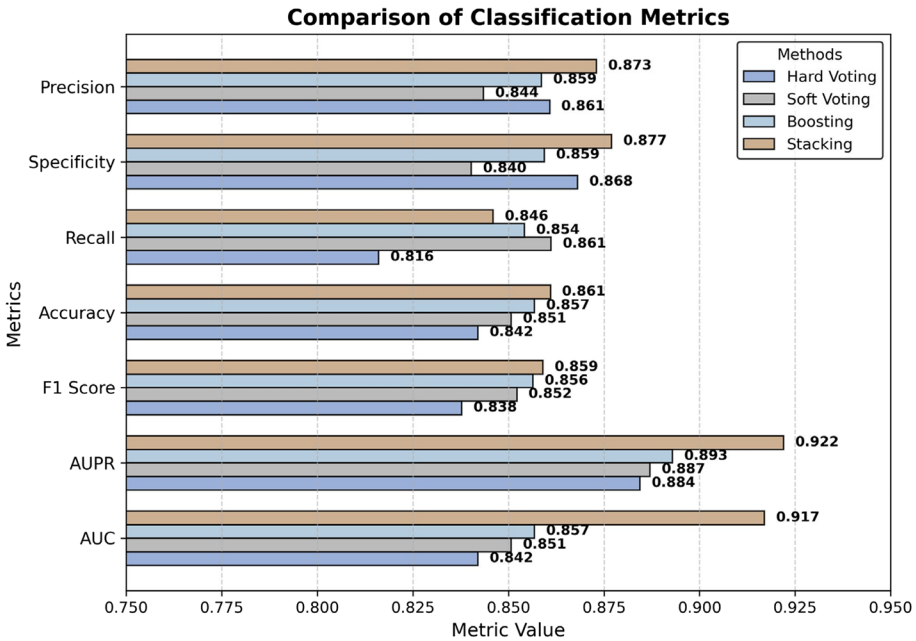


**Fig. 7** Performance comparison of different ensemble methods on the independent set

below 0.3 and the TM-score exceeded 0.5 [45–47]. Due to their limited sequence conservation, these pairs were challenging to detect using traditional sequence alignment methods (e.g., BLASTp [48]). However, structure-based alignment tools like TM-align were able to effectively identify their homology. As shown in Fig. 8 (A-C), although these three protein pairs exhibited extremely low sequence similarity, their structures displayed high similarity. In such cases, sequence alignment methods failed to accurately determine homology, whereas both TM-align and TM-predictor successfully identified the relationships. Notably, TM-predictor maintained high sensitivity without relying on 3D structural input. As illustrated in Fig. 8 (D-F), the predicted TM-scores exhibited a strong linear correlation with the actual TM-scores, where the coefficient of determination ($R^2$) was approximately 0.9 or even higher, indicating a high level of accuracy in the predictions. This ability arose from leveraging protein language models to extract remote homology signals from deep sequence embeddings. Furthermore, TM-predictor was trained using structural similarity (TM-score) as the supervisory signal, enabling the PLMVF model to reliably predict structural similarity even in the absence of structural input.

## Conclusion

Addressing the increasingly severe problem of bacterial antibiotic resistance, this study proposes an innovative method for identifying VFs in pathogens by integrating protein language model with ensemble learning strategies (PLMVF). This approach aims to overcome the limitations of existing sequence similarity-based methods that cannot effectively identify remote homology relationships. Specifically, first, we calculate the true TM-score using the 3D structures of proteins. Next, we train a predictive model for structural similarity (TM-predictor) using a large dataset of known true structural similarities (TM-scores). The goal of this model is to predict the TM-score between new pairs of proteins, thereby capturing hidden remote homology information within sequences. This discovery holds significant implications for the future development of novel anti-virulence therapies, providing new perspectives and tools for identifying effective therapeutic targets. Moreover, compared to existing models, PLMVF demonstrates superior performance, establishing its effectiveness and reliability in accurately identifying VFs.

In summary, The PLMVF model introduced in this study provides an efficient and reliable approach for VF identification and prediction, establishing itself as a key resource in the fight against antibiotic-resistant bacterial infections. With the deepening understanding of bacterial pathogenic mechanisms and technological advancements, we believe that PLMVF and its derivative approaches will play a critical role in the development of future anti-infective therapies, contributing to the effective prevention and control of pathogenic bacterial infections.

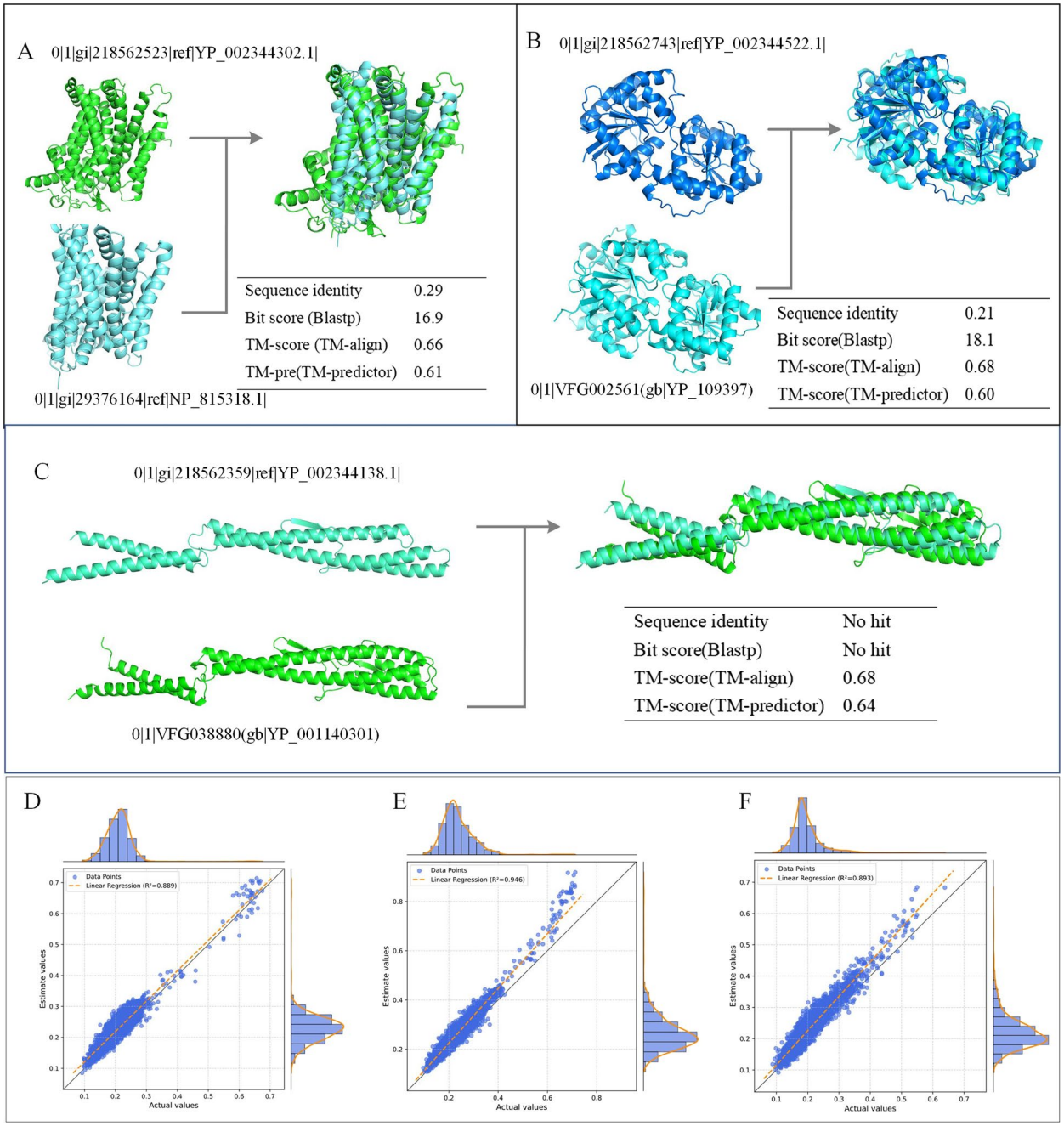


**Fig. 8** Case studies of remote homology pairs using PLMVF. **A**–**C** Structural alignments of proteins 0|1|gi|218,562,523|ref|YP_002344302.1|, 0|1|gi|218,562,743|ref|YP_002344522.1|, and 0|1|gi|218,562,359|ref|YP_002344138.1|, showing high structural similarity despite low sequence identity; **D**–**F** Linear correlation plots between predicted and actual TM-scores for each of the above proteins against all samples

Although PLMVF demonstrates strong predictive performance, several limitations persist, presenting opportunities for future enhancement. First, our model currently operates on protein sequences derived from translated coding regions. However, in practical scenarios—especially in metagenomic or raw genome data analysis—protein sequences are often not directly available. To address this, we plan to extend our framework to use nucleotide (DNA) sequences as input, enabling VF prediction directly from raw genomic data. Second, while PLMVF is a standalone pipeline at this stage, its utility and accessibility could be greatly enhanced by developing

a web-based platform. This platform would allow users to upload sequences and receive VF predictions through an intuitive interface, eliminating the need for local installation or computational resources. Third, although the current model captures sequence-level information through ESM-2 embeddings and structural similarity via predicted TM-scores, it does not directly utilize 3D structural features of proteins. In future work, we aim to integrate structure-aware representations to better characterize the spatial and functional properties of VFs. Collectively, these future directions—including DNA-level prediction, structural feature integration, and deployment as a user-friendly website—will greatly expand the applicability, interpretability, and usability of PLMVF in diverse research and clinical settings.

**Abbreviations**

| | |
|---|---|
| VFs | Virulence factors |
| 3D | Three-dimensional |
| ACC | Accuracy |
| SVM | Support Vector Machine |
| AI | Artificial intelligence |
| DL | Deep learning |
| LSTM | Long Short-Term Memory |
| CNN | Convolutional Neural Networks |
| MSA | Multiple sequence alignment |
| GCN | Graph convolutional networks |
| KAN | Knowledge-Augmented Network |
| GRU | Gated Recurrent Unit |

**Acknowledgements**
Not applicable.

**Authors' contributions**
GL: conceived the study, analyzed the results, drafted the article. JZ: collected the data, designed and performed the experiments, drafted the article. JL: revised the article. CL: supervised the study, revised the article. All authors read and approved the final manuscript.

**Funding**
This work was supported by the National Natural Science Foundation of China [grant numbers 62362034, 62372279]; the Natural Science Foundation of Jiangxi Province of China [grant numbers 20232 ACB202010, 20232 ACB205001]; the Natural Science Foundation of Shandong Province [grant number ZR2023MF119]; the Jiangxi Province Key Laboratory of Advanced Network Computing [grant number 2024SSY03071]; and the Major Discipline Academic and Technical Leaders Training Program of Jiangxi Province [grant number 20232BCJ22025].

**Data availability**
The dataset and source code of PLMVF are available at GitHub (https://github.com/ghli16/PLMVF).

## Declarations

**Ethics approval and consent to participate**
Not applicable.

**Consent for publication**
Not applicable.

**Competing interests**
The authors declare no competing interests.

Received: 25 March 2025 Accepted: 9 May 2025
Published online: 21 May 2025

**References**

1. Van Oosten M, Hahn M, Crane LMA, et al. Targeted imaging of bacterial infections: advances, hurdles and hopes. FEMS Microbiol Rev. 2015;39:892–916.
2. Chen L, Yang J, Yu J, et al. VFDB: a reference database for bacterial virulence factors. Nucleic acids Res. 2005;33:D325–8.
3. Dickey SW, Cheung GYC, Otto M. Different drugs for bad bugs: antivirulence strategies in the age of antibiotic resistance. Nat Rev Drug Discovery. 2017;16:457–71.
4. Liu B, Zheng D, Zhou S, et al. VFDB 2022: a general classification scheme for bacterial virulence factors. Nucleic acids Res. 2022;50:D912–7.
5. Sayers S, Li L, Ong E, et al. Victors: a web-based knowledge base of virulence factors in human and animal pathogens. Nucleic acids Res. 2019;47:D693-700.
6. Zhou CE, Smith J, Lam M, et al. MvirDB—a microbial database of protein toxins, virulence factors and antibiotic resistance genes for bio-defence applications. Nucleic acids Res. 2007;35:D391–4.
7. Finn RD, Clements J, Arndt W, et al. HMMER web server: 2015 update. Nucleic acids Res. 2015;43:W30–8.
8. Johnson M, Zaretskaya I, Raytselis Y, et al. NCBI BLAST: a better web interface. Nucleic acids Res. 2008;36:W5–9.
9. Li J, Tai C, Deng Z, et al. VRprofile: gene-cluster-detection-based profiling of virulence and antibiotic resistance traits encoded within genome sequences of pathogenic bacteria. Brief Bioinform. 2018;19(4):566–74.
10. Liu B, Zheng D, Jin Q, et al. VFDB 2019: a comparative pathogenomic platform with an interactive web interface. Nucleic acids Res. 2019;47:D687–92.
11. Stamatakis A. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. Bioinformatics. 2014;30:1312–3.
12. Zhang Y, Skolnick J. Scoring function for automated assessment of protein structure template quality. Proteins Struct Function and Bioinformatics. 2004;57:702–10.
13. Zhang Y, Skolnick J. TM-align: a protein structure alignment algorithm based on the TM-score. Nucleic acids Res. 2005;33:2302–9.
14. Sachdeva G, Kumar K, Jain P, et al. SPAAN: a software program for prediction of adhesins and adhesin-like proteins using neural networks. Bioinformatics. 2005;21:483–91.
15. Garg A, Gupta D. VirulentPred: a SVM based prediction method for virulent proteins in bacterial pathogens. BMC Bioinformatics. 2008;9:1–12.
16. Gupta A, Kapil R, Dhakan DB, et al. MP3: a software tool for the prediction of pathogenic proteins in genomic and metagenomic data. PLoS ONE. 2014;9: e93907.
17. Singh S, Le NQK, Wang C. VF-Pred: Predicting virulence factor using sequence alignment percentage and ensemble learning models. Comput Biol Med. 2024;168: 107662.
18. Xie R, Li J, Wang J, et al. DeepVF: a deep learning-based hybrid framework for identifying virulence factors using the stacking strategy. Brief bioinform. 2021;22(3): bbaa125.
19. Sun J, Yin H, Ju C, et al. DTVF: A User-Friendly Tool for Virulence Factor Prediction Based on ProtT5 and Deep Transfer Learning Models. Genes. 2024;15: 1170.
20. Dill KA, Ozkan SB, Shell MS, et al. The protein folding problem. Annu Rev Biophys. 2008;37:289–316.
21. Onuchic JN, Wolynes PG. Theory of protein folding. Curr Opin Struct Biol. 2004;14:70–5.
22. Matthews CR. Pathways of protein folding. Annu Rev Biochem. 1993;62:653–83.
23. Baek M, DiMaio F, Anishchenko I, et al. Accurate prediction of protein structures and interactions using a three-track neural network. Science. 2021;373:871–6.
24. Jumper J, Evans R, Pritzel A, et al. Highly accurate protein structure prediction with AlphaFold. Nature. 2021;596:583–9.
25. Varadi M, Anyango S, Deshpande M, et al. AlphaFold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. Nucleic acids Res. 2022;50:D439–44.

26. Lin Z, Akin H, Rao R, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. Science. 2023;379:1123–30.
27. Li G, Bai P, Chen J, et al. Identifying virulence factors using graph transformer autoencoder with ESMFold-predicted structures. Comput Biol Med. 2024;170: 108062.
28. Ying C, Cai T, Luo S, et al. Do transformers really perform badly for graph representation? Adv Neural Inf Process Systems(NeurIPS). 2021;34:28877–88.
29. Lyu H, Sha N, Qin S, et al. Manifold denoising by nonlinear robust principal component analysis. Adv Neural Inf Process Systems(NeurIPS). 2019;32:13390-400.
30. Ju F, Zhu J, Shao B, et al. CopulaNet: Learning residue co-evolution directly from multiple sequence alignment for protein structure prediction. Nat Commun. 2021;12(1):2535.
31. Zhang C, Shine M, Pyle AM, et al. US-align: universal structure alignments of proteins, nucleic acids, and macromolecular complexes. Nat Methods. 2022;19:1109–15.
32. Liu W, Wang Z, You R, et al. PLMSearch: Protein language model powers accurate and fast sequence search for remote homology. Nat Commun. 2024;15:2775.
33. Hamamsy T, Morton JT, Blackwell R, et al. Protein remote homology detection and structural alignment using deep learning. Nat Biotechnol. 2024;42:975–85.
34. Liu Z, Wang Y, Vaidya S, et al. Kan: Kolmogorov-arnold networks. arXiv preprint. 2024. https://doi.org/10.48550/arXiv.2404.19756.
35. Mao C, Abraham D, Wattam AR, et al. Curation, integration and visualization of bacterial virulence factors in PATRIC. Bioinformatics. 2015;31:252–8.
36. Wattam AR, Davis JJ, Assaf R, et al. Improvements to PATRIC, the all-bacterial bioinformatics database and analysis resource center. Nucleic acids Res. 2017;45:D535–42.
37. Rentzsch R, Deneke C, Nitsche A, et al. Predicting bacterial virulence factors–evaluation of machine learning and negative data strategies. Brief Bioinform. 2020;21:1596–608.
38. Fu L, Niu B, Zhu Z, et al. CD-HIT: accelerated for clustering the next-generation sequencing data. Bioinformatics. 2012;28:3150–2.
39. Luo Z, Wang R, Sun Y, et al. Interpretable feature extraction and dimensionality reduction in ESM2 for protein localization prediction. Brief Bioinform. 2024;25(2): bbad534.
40. Koushik J. Understanding convolutional neural networks. arXiv preprint. 2016. https://doi.org/10.48550/arXiv.1605.09081.
41. Chung J, Gulcehre C, Cho K H, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint. 2014. https://doi.org/10.48550/arXiv.1412.3555.
42. Shi H, Liu S, Chen J, et al. Predicting drug-target interactions using Lasso with random forest based on evolutionary information and chemical structure. Genomics. 2019;111:1839–52.
43. Huo Y, Xin L, Kang C, et al. SGL-SVM: a novel method for tumor classification via support vector machine with sparse group Lasso. J Theor Biol. 2020;486: 110098.
44. Ye J, McGinnis S, Madden TL. BLAST: improvements for better sequence analysis. Nucleic acids Res. 2006;34:W6-9.
45. Rost B. Twilight zone of protein sequence alignments. Protein Eng. 1999;12:85–94.
46. Xu J, Zhang Y. How significant is a protein structure similarity with TM-score= 0.5? Bioinformatics. 2010;26:889–95.
47. Zhang Y, Hubner IA, Arakaki AK, et al. On the origin and highly likely completeness of single-domain protein structures. Proc Natl Acad Sci. 2006;103:2605–10.
48. Altschul SF, Gish W, Miller W, et al. Basic local alignment search tool. J Mol Biol. 1990;215:403–10.


## Publisher's Note