Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

PhageVir: An evaluation of computational intelligence models for the precise identification of phage virion proteins

Nashwan Alromema [1, *], Hussnain Arshad [2], Sharaf J. Malebary [3], Faisal Binzagr [1], Yaser Daanial Khan [4]

[1]*Department of Computer Science, Faculty of Computing and Information Technology-Rabigh, King Abdulaziz University, Jeddah, Saudi Arabia*
[2]*Department of Artificial Intelligence, School of Systems and Technology, University of Management and Technology, Lahore, Pakistan*
[3]*Department of Information Technology, Faculty of Computing and Information Technology-Rabigh, King Abdulaziz University, Jeddah, Saudi Arabia*
[4]*Department of Computer Science, School of Systems and Technology, University of Management and Technology, Lahore, Pakistan*

ARTICLE INFO

Article history:
Received 8 January 2025
Received in revised form
29 April 2025
Accepted 3 May 2025

Keywords:
Phage virion proteins
Computational model
Feature selection
Deep learning
Phage therapy

ABSTRACT

This study presents PhageVir, an enhanced computational model developed to predict Phage Virion Proteins (PVPs), which are essential for bacteriophage infection and replication. PhageVir integrates advanced feature selection methods, including the Position Relative Incidence Matrix (PRIM) and the Reverse Position Relative Incidence Matrix (RPRIM), to effectively capture key sequence features and positional dependencies within protein sequences. Several machine learning and deep learning algorithms were employed, including LightGBM, Random Forest, Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Recurrent Neural Network (RNN), and Artificial Neural Network (ANN), to classify PVPs based on sequential data. Model performance was evaluated through independent set testing, self-consistency testing, and cross-validation, using metrics such as accuracy (ACC), specificity (Sp), sensitivity (SN), Z-score, and Matthews correlation coefficient (MCC). The CNN model demonstrated strong performance in cross-validation, achieving an accuracy of 0.833, sensitivity of 0.832, specificity of 0.834, a correlation coefficient of 0.665, an AUC score of 0.927, and a Z-score of 1.37. The results confirm the effectiveness of the proposed computational approach for accurate PVP classification. Beyond its predictive power, PhageVir offers valuable biological insights into phage infection mechanisms, supporting advancements in phage therapy and antibacterial treatments.

© 2025 The Authors. Published by IASE. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

Bacterial infections remain a major global health concern, contributing significantly to morbidity and mortality. The increasing prevalence of antibiotic-resistant bacteria has necessitated the exploration of alternative therapeutic approaches. Bacteriophages (phages), viruses that specifically infect and replicate within bacterial cells, have emerged as a promising solution due to their host specificity and potential to target antibiotic-resistant strains (Yang et al., 2020). Unlike broad-spectrum antibiotics, phages selectively target bacterial hosts, minimizing off-target effects and microbiome disruption. Phages possess a distinctive structure comprising a nucleic acid core enclosed within a protein capsid. Among these proteins, phage virion proteins (PVPs) play a crucial role in host recognition, attachment, and infection (Emon et al., 2024). Identifying and characterizing PVPs is essential for understanding phage biology and advancing phage therapy applications. Computational methods have gained traction for predicting PVPs based on protein sequence data, offering high-throughput and cost-effective alternatives to experimental approaches (Gao et al., 2024). Computational approaches for predicting PVPs from protein sequence data have recently been developed (Ji et al., 2024). These methods range from simple sequence-based techniques to more complex machine-learning ones.

* Corresponding Author.
Email Address: nalromema@kau.edu.sa (N. Alromema)
https://doi.org/10.21833/ijaas.2025.05.013
Corresponding author's ORCID profile:
https://orcid.org/0000-0001-6208-2863
2313-626X/© 2025 The Authors. Published by IASE.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

Deep learning methods, including CNN, RNN, and GRU, have demonstrated potential in various protein prediction applications, including PVP prediction. Recently, there has been massive interest in using bacteriophages, or phages, as a viable alternative to antibiotics for treating bacterial infections (Bajiya et al., 2023). Viruses that infect and multiply within bacterial cells are known as phages and are very specialized to their bacterial hosts (Manavalan et al., 2018).

To forecast PVPs, various computational methods have been developed, ranging from simple sequence-based ways to more complicated machine learning algorithms. Table 1 illustrates the models. Ru et al. (2019) employed a Random Forest classification technique with a 10-fold cross-validation and obtained a 93.5 percent accuracy. It identified charge property as the most significant factor in classification. In contrast, Feng et al. (2022) reported DeepPVP, a deep learning-based technique for finding and classifying PVPs within phage genomes. It achieved an accuracy of 90.19 percent on a 10-fold cross-validation test.

Furthermore, Charoenkwan et al. (2020a) introduced PV-Pred-SCM, a scoring card system (SCM), and a dipeptide composition-based method for identifying and characterizing phage virion proteins. It obtained a 10-fold cross-validation accuracy of 92.52 percent and an MCC score of 0.846. In another experiment, Charoenkwan et al. (2020b) published Meta-iPVP, which identifies PVPs using probabilistic information. It had an 84.6 percent cross-validation accuracy. Furthermore, Han et al. (2021) introduced iPVP-MCV, an ensemble model for precise PVP annotation based on protein sequences. It scored an 84.6 percent accuracy in 10-fold cross-validation.

Bao et al. (2022) proposed Phage UniR LGBM, a model for classifying virion proteins that uses the UniRep feature set in conjunction with LightGBM as the classification technique. Its accuracy was 89.18% when tested with the LGBM model using a 10-fold cross-validation. Finally, Ahmad et al. (2022) introduced SCORPION, a model for computationally classifying phage virion proteins (PVPs) using just the primary sequences of proteins. The approach employs 13 different feature descriptors from various aspects and ten machine learning algorithms. It attained an accuracy of 86.8 percent in 10-fold cross-validation.

Despite these advancements, existing methods exhibit limitations in feature selection, predictive performance, and biological interpretability. To address these gaps, this study introduces PhageVir, a novel computational framework for PVP prediction. The key contributions of this work include:

- Construction of a robust feature vector incorporating PRIM, RPRIM, AAPIV, RAAPIV, and FV descriptors from a curated benchmark dataset.
- Implementation of dimensionality reduction techniques leveraging Hahn, Raw, and central statistical moments to enhance model efficiency.
- Evaluation of multiple deep learning architectures (ANN, CNN, RNN, LSTM, GRU) alongside machine learning classifiers (RF, LGBM) for PVP prediction.
- Validation through independent testing, self-consistency, and cross-validation to ensure model robustness.
- Performance assessment using accuracy, specificity, recall, MCC, and AUC scores.

By integrating advanced feature representations and diverse predictive models, PhageVir aims to enhance PVP prediction accuracy and contribute to the development of computational tools for phage-based therapeutics.

2. Materials and methods

The section discusses the methods involved in obtaining benchmark datasets, extracting features, selecting appropriate models, and evaluating their performance.

2.1. Benchmark dataset collection

A benchmark dataset was constructed by collecting protein sequences from UniProt, yielding 464 PVP samples and 1,429 non-PVP samples. For PVPs, the Organism [OS] field was set to “phage,” and the Subcellular location was specified as “virion.” In contrast, for non-PVPs, the Organism [OS] field remained “phage,” but the Subcellular location was explicitly set to NOT “virion.” Only reviewed sequences were included in the dataset. All sequences were formatted in FASTA, a widely used standard for managing protein and DNA sequences (Arora et al., 2024).

To ensure data quality, BioEdit software was used to assess sequence integrity. Any sequences of low quality or containing ambiguous bases were removed. The dataset was then processed with CD-HIT, a clustering tool that reduces redundancy by grouping highly similar sequences, thereby producing a streamlined, non-redundant dataset for further analysis.

To improve dataset balance and reduce potential biases, an under-sampling technique was applied, yielding a final dataset of 393 PVPs and 393 non-PVPs. This curated dataset was subsequently used to train and evaluate multiple machine learning and deep learning models, including LightGBM (LGBM), Random Forest (RF), Convolutional Neural Networks (CNN), Long Short-Term Memory Networks (LSTM), Gated Recurrent Units (GRU), Artificial Neural Networks (ANN), and Recurrent Neural Networks (RNN). Fig. 1 provides an overview of the entire preprocessing workflow.

Biotechnology has made remarkable strides thanks to advances in information technology. One of the biggest challenges in designing computational algorithms is converting primary sequences into a collection of fixed-sized numeric features based on context-specific functional information. CNN, LSTM, and RNN are deep learning algorithms that have

been created to handle vector input and have been used to evaluate proteomic data. A discrete model can transform sequential data into a fixed-sized vector while preserving characteristic information regarding the sequence (Khan et al., 2021; Perveen et al., 2023).

2.2. Feature formulation

Feature extraction plays a crucial role in analyzing proteomic sequences. Various computational techniques, such as position-based variation and composition-specific feature extraction, help derive meaningful characteristics from these sequences. Below, we outline the key feature extraction methods and their relevance in capturing essential attributes of protein sequences.

2.2.1. Position relative incidence matrix (PRIM)

The latent attributes of a protein can be unveiled through an analysis of the dispersed sequences of amino acid residues within a protein sequence (Perveen et al., 2023). The sequence of the polypeptide chain embeds within itself characteristics important in discerning the attributes of the polypeptide chain. A constructed matrix examines the positional correlations among all these residues to extract meaningful insights and reveal patterns formed by the arrangement of residues (Suleman et al., 2023). Referred to as PRIM, this matrix assumes a dimension of 20x20, corresponding to each residue present in the arbitrary polypeptide chain. It is derived from the relative disposition of residues within a sample polypeptide and estimates the positional information about a protein. The details are as follows:

$$R_{PRIM} = \begin{bmatrix} R_{1\to1} & R_{1\to2} & \cdots & R_{1\to y} & \cdots & R_{1\to20} \\ R_{2\to1} & R_{2\to2} & \cdots & R_{2\to y} & \cdots & R_{2\to20} \\ \vdots & \vdots & & \vdots & & \vdots \\ R_{x\to1} & R_{x\to2} & \cdots & R_{x\to y} & \cdots & R_{i\to20} \\ \vdots & \vdots & & \vdots & & \vdots \\ R_{A\to1} & R_{A\to2} & \cdots & R_{A\to y} & \cdots & R_{A\to20} \end{bmatrix} \quad (1)$$

Every element R_{ij} of the matrix denotes the accumulation of positional details of the i^{th} residue in correlation with the j^{th} ordinal residue (Allehaibi et al., 2021). The resultant matrix consists of a total of 400 coefficients. Statistical moments are then computed for dimensionality reduction, as a result, 400 coefficients of the matrix are reduced to just 30. Unlike simple frequency-based methods, PRIM preserves the positional dependency of residues, which is crucial for understanding protein functionality and structure.

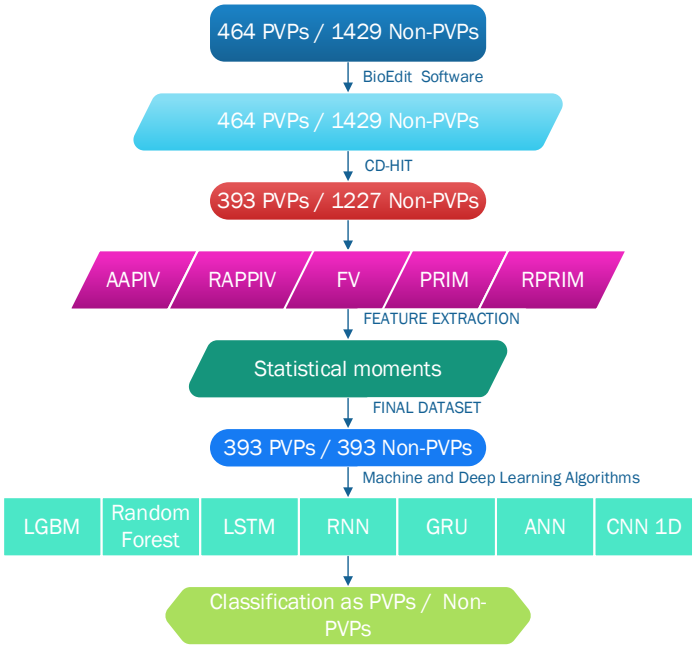

Fig. 1: Prediction model of phage virion proteins

2.2.2. Reverse position relative incidence matrix (RPRIM)

The RPRIM method, just like earlier enumeration techniques, delves into discovering hidden features of homologous peptide sequences (Butt et al., 2022). To calculate RPRIM, the reverse of the original sequence is used. Below, you can see the RPRIM matrix that resulted from this method:

$$Q_{RPRIM} = \begin{bmatrix} Q_{1\to1} & Q_{1\to2} & \cdots & Q_{1\to y} & \cdots & Q_{1\to20} \\ Q_{2\to1} & Q_{2\to2} & \cdots & Q_{2\to y} & \cdots & Q_{2\to20} \\ \vdots & \vdots & & \vdots & & \vdots \\ Q_{x\to1} & Q_{x\to2} & \cdots & Q_{x\to y} & \cdots & Q_{i\to20} \\ \vdots & \vdots & & \vdots & & \vdots \\ Q_{A\to1} & Q_{A\to2} & \cdots & Q_{A\to y} & \cdots & Q_{A\to20} \end{bmatrix} \quad (2)$$

The RPRIM matrix, like the PRIM matrix, consists of 400 coefficients. Both RPRIM and PRIM reduce

their feature dimension down to 30 coefficients through the use of statistical moments.

2.2.3. Frequency vector (FV)

The distribution of residue within the polypeptide chain is illustrated through the frequency vector. It provides crucial sequential information regarding the protein sample. It determines how frequently a protein has specified residues. Thereby preserving information about the sequence's composition and distribution (Alghamdi et al., 2021). The FV is represented below:

$$FV = [f_1, f_2, f_3, \dots, f_{20}] \quad (3)$$

The frequency vector illustrates the occurrence of arbitrary amino acids in a protein sample. It is arranged in alphabetical order. Since it holds the information of each residue therefore its length is 20. Understanding protein composition is critical for classification tasks. FV captures sequence diversity while being computationally efficient.

2.2.4. Accumulative absolute position incidence vector (AAPIV)

The frequency vector collects positional statistics on amino acid residues within a sequence, uncovering indeterminate characteristics linked to compositional details. However, it does not offer insights into the positional relationships among the amino acid residues (Barukab et al., 2022). The AAPIV is introduced to overcome this limitation. It calculates the relative placement information of native amino acids in the following manner:

$$K = [\forall_1, \forall_2, \forall_3, \dots, \forall_n] \quad (4)$$

The calculation of the i^{th} segment of *AAPIV* is expressed as follows:

$$\forall_i = \Sigma^n{}_{k=1}\, \beta_k \quad (5)$$

The sum of ordinal positions within the primary sequence for the i^{th} residue is held by the i^{th} element of AAPIV. AAPIV adds positional relevance, essential for distinguishing functionally similar sequences with different distributions.

2.2.5. Reverse accumulative absolute position incidence vector (RAAPIV)

RAAPIV is a vector similar to AAPIV but is computed using the reverse sequence of the actual sequence (Baig et al., 2022). It provides further insight into the positional information necessary for uncovering hidden properties of sequences. RAAPIV complements AAPIV by incorporating directionality awareness, improving feature robustness. The RAAPIV vector is represented as:

$$RAAPIV = [n_1, n_2, n_3, \dots, n_m] \quad (6)$$

2.2.6. Statistical moments

Statistical moments are crucial in converting proteomic sequences into a fixed-size vector. Each of the moments used represents specific information about the characteristics of the data. Enormous work has been conducted for translation of data into moments of varying distributions such that the formed coefficients preserve the semantics of original sequential data (Butt et al., 2023).

The feature vector is formed based on the Hahn, central, and raw moments computed from the descriptor matrices derived from proteomic data. These moments are used as a succinct representation of elements that contribute to the identification of attributes of the input vector. It is well understood among researchers that the characterization of multiomic sequences depends on the relative positioning and composition of their basic components. Therefore, mathematical and computational models have emphasized the interrelated placement of amino acid residues in proteomic sequences to boost the feature vector. This aspect ensures a consistent and diligent feature vector (Suleman et al., 2022).

A two-dimensional organization of data is the basic requirement for the computation of Hahn moments, therefore, arbitrary sequences are mapped onto a two-dimensional matrix denoted as G', with size k*k. This matrix holds the exact information as G but is a two-dimensional representation. The ceiling of the square root of n determines the value of k.

$$k = \lceil \sqrt{n} \rceil \quad (7)$$

The representation of G' is as follows:

$$G' = \begin{bmatrix} G_{11} & G_{12} & \cdots & G_{1n} \\ G_{21} & G_{22} & \cdots & G_{2n} \\ \vdots & \vdots & & \vdots \\ G_{m1} & G_{m2} & \cdots & G_{mn} \end{bmatrix} \quad (8)$$

Calculating statistical moments from the square matrix above not only reduces the dimensionality but also converts the variable-length sequence into a fixed-size representation. As mentioned earlier, this study uses raw, Hahn, and central moments.

The following equation expresses how raw moments of order a+b are computed:

$$W_{ab} = \sum_{e=1}^{k} \sum_{f=1}^{k} G_{ef} e^a f^b \quad (9)$$

where, W_{ab} is the raw moment of a+b, G_{ef} denotes the value at the grid cell located at row e and column f, and k is the size of grid.

Most of the important information embedded in the sequences can be sieved out by using moments up to the third order, represented by $W_{00}, W_{10}, W_{01}, W_{11}, W_{20}, W_{02}, W_{21}, W_{12}, W_{03}$ and W_{30}. Computation of central moments requires the data's

centroid (x, y), which can be easily calculated using the first three raw moments. The centroid is used:

$$v_{ab} = \Sigma^{n}{}_{e=1}\Sigma^{n}{}_{f=1}\,(e-\bar{x})^{a}(f-\bar{y})^{b}G_{ef} \quad (10)$$

where, $\bar{x}$ and $\bar{y}$ represent the coordinates of the centroid, G_{ef} denotes the value at the grid cell located at row e and column f.

A square grid is employed as input for computing Hahn moments. Hahn moments have reversible properties, which essentially means inverse Hahn moments can be applied to sparely rebuild the original data. This reversible property ensures that latent information transformed and embedded within multiomic sequences stays preserved. Ultimately, this characteristic information is blended into the feature set. The following equation illustrates the computation of Hahn moments:

$$h_n^{x,y}(p,Q) = (Q+V-1)_n(Q-1)_n \times \Sigma^{n}{}_{z=0}(-1)^{z}\frac{(-n)_z(-p)_z(2Q+x+y-n-1)_z}{(Q+y-1)_z(Q-1)_z}\frac{1}{z!} \quad (11)$$

where, $h_n^{x,y}(p,Q)$ Hahn polynomial of order n, parameterized by p, Q, and spatial indices x, y.

Eq. 11 employs the Gamma operator and Pochhammer notation, which Akmal and Coulton (2020) explained. The coefficients yielded via Hahn moments using the above equation are characteristically normalized based on the coefficient stated in the equation below:

$$H_{pq} = \Sigma^{G-1}{}_{j=0}\Sigma^{G-1}{}_{i=0}\,\delta_{pq}h^{a,b}{}_{p}(j,Q)\,h^{a,b}{}_{q}(i,Q), \quad m,n = 0,1,2,\dots,Q-1 \quad (12)$$

The features chosen—PRIM, RPRIM, FV, AAPIV, RAAPIV, and Statistical Moments—each provide complementary insights into protein sequence composition, structure, and positional relationships. Their combination ensures a robust, multi-perspective feature representation, crucial for accurately characterizing and distinguishing proteomic sequences.

2.3. Machine learning models

The study included two machine learning models: Random Forest and Light Gradient Boosting Machine. Each of these models has distinct qualities and capabilities and has widespread application in various machine learning applications.

2.3.1. LGBM (light gradient boosting machine)

Light Gradient Boosting Machine (LGBM) uses a supervised learning algorithm built on the gradient boosting framework, but each boosting iteration employs a different technique for tree construction. Using a histogram-based method, LGBM determines the optimum split points and can handle massive datasets efficiently. It is renowned for its excellent accuracy, speed, and capability to handle skewed data. The model is depicted in Fig. 2. Additionally, the model's performance is enhanced through fine-tuning various hyperparameters, including learning rate, number of estimators, max depth, and regularization parameters. These hyperparameters, detailed in Table 1, allow for improved optimization and better predictive accuracy.

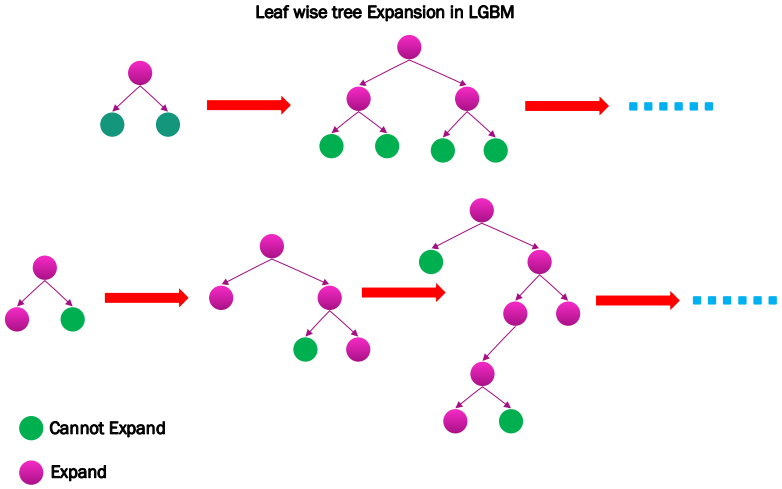

Fig. 2: LGBM model used in this study

Table 1: Hyperparameters used for the LGBM model

Hyperparameter	Description	Value
Learning rate	Controls the step size at each iteration	0.05
Number of estimators	Number of boosting rounds	1000
Max depth	Maximum depth of each tree	-1 (unlimited)
Min data in leaf	Minimum number of data points in a leaf node	20
Feature fraction	Fraction of features used per iteration	0.8
Bagging fraction	Fraction of data used for training each tree	0.8
Lambda L1	L1 regularization parameter	0.1
Lambda L2	L2 regularization parameter	0.1
Boosting type	Type of boosting used	gbdt
Objective	Loss function	binary

2.3.2. Random forest

Random Forest (RF) is a widely used ensemble learning method based on decision trees. It trains numerous decision trees on bootstrap data samples and aggregates their predictions. It yields remarkable accuracy and resists overfitting (Alzahrani et al., 2021). It also includes randomness factors in the way the trees are built, such as feature sampling and bootstrap sampling, which can further increase the diversity of the trees and reduce overfitting. Hyperparameters are shown in Table 2. Random Forest is a popular solution for many machine learning problems, particularly those involving high-dimensional data. Fig. 3 depicts the model.

2.3.3. Deep learning

CNN1D, LSTM, RNN, GRU, and ANN were the deep learning algorithms employed for PVP prediction. All models were built with the Keras framework with TensorFlow as a backend. Each model has a distinct architecture built particularly for different kinds of data and applications.

2.3.4. CNNID

The CNN1D model is a famous sequential data processing architecture. This study used a CNN1D model with one convolutional layer, a max-pooling layer, and a fully connected layer. A batch size of 64 and 150 epochs was used during the training phase.

The Rectified Linear Unit (ReLu) activation function was used in the hidden layer to incorporate non-linearity and improve the model's representative capability. The output layer was furnished with a sigmoid activation function because it is most appropriate for the binary classification problem. The hyperparameters are shown in Table 3.

Table 2: Hyperparameters used for the random forest model

Hyperparameter	Description	Value
Number of estimators	Number of trees in the forest	100
Max depth	Maximum depth of each tree	None
Min samples split	Minimum samples required to split a node	2
Min samples leaf	Minimum samples required at a leaf node	1
Max features	Number of features considered for the best split	SQRT
Bootstrap	Whether bootstrap samples are used when building trees	True
Criterion	Function to measure the quality of a split	Gini

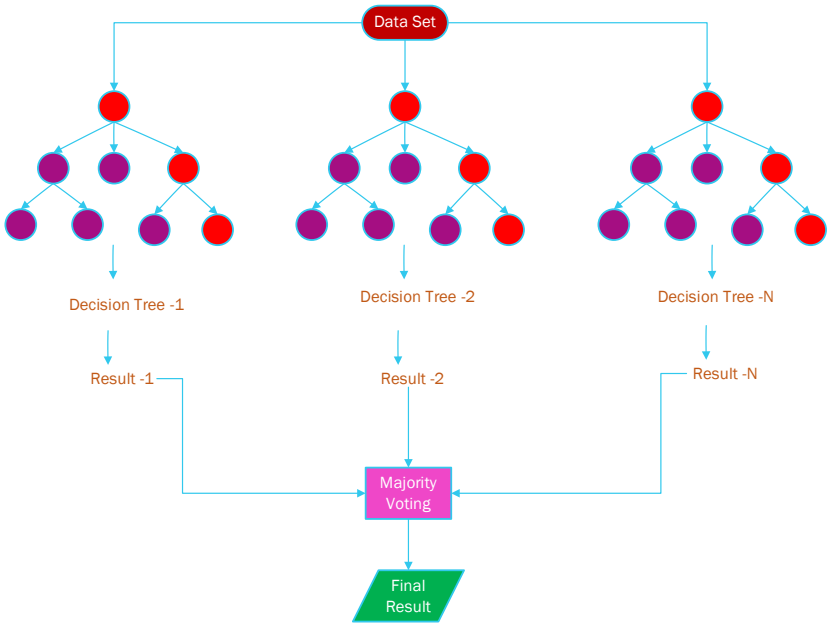

Fig. 3: Random forest model used in this study

Table 3: Hyperparameters used for the CNN1D model

Hyperparameter	Description	Value
Batch size	Number of training examples per batch	64
Epochs	Number of times the model is trained on the dataset	150
Learning rate	Controls step size during optimization	0.001
Kernel size	Size of the convolutional filter	3
Number of filters	Number of filters in the convolutional layer	64
Pool size	Size of the max-pooling window	2
Activation function	Non-linear activation function used	ReLU
Optimizer	Optimization algorithm	Adam
Loss function	Function used to evaluate model performance	Binary cross-entropy

The model's performance was further enhanced by combining a binary cross-entropy loss function with the Adam optimizer. This combination is successful in training neural networks to perform similar tasks. The convolutional layer within the CNN1D model applied a series of filters to the input sequence, allowing for the extraction of local features and patterns. Subsequently, the max-pooling layer reduced the output of the convolutional layer into a single feature vector, focusing on the most significant features (Barshai et al., 2021). Finally, the fully connected layer performed a linear transformation on the feature vector, mapping it to a binary output determining whether PVPs are present. A visual representation of the CNN1D model proposed in this work is depicted in Fig. 4, illustrating the flow of information through the various layers of the model.

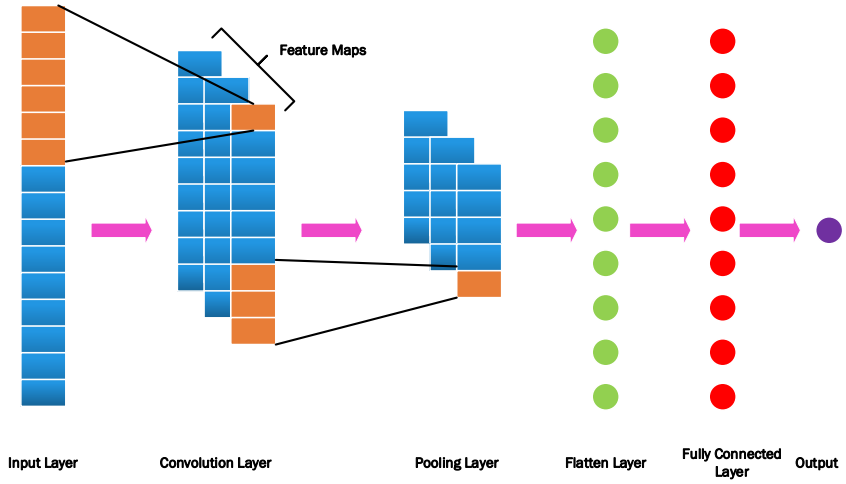

Fig. 4: CNN model used in this study

2.3.5. Long short-term memory (LSTM)

The LSTM model is an extension of the recurrent neural network, developed primarily to identify long-term dependencies in sequential data. Fig. 5 shows a modified version of the LSTM model used in this study. It was trained by running it for 150 epochs with a batch size of 64. Hyperparameters are shown in Table 4.

Table 4: Hyperparameters used for the LSTM model

Hyperparameter	Description	Value
Batch size	Number of training examples per batch	64
Epochs	Number of times the model is trained on the dataset	150
Learning rate	Controls step size during optimization	0.001
Number of LSTM units	Number of memory units in the LSTM layer	128
Dropout rate	Fraction of neurons dropped for regularization	0.2
Activation function	Non-linear activation function used in hidden layers	ReLU
Optimizer	Optimization algorithm	Adam
Loss function	Function used to evaluate model performance	Binary cross-entropy

In the proposed model, the LSTM model is clamped to a fully connected layer (FCL). The LSTM layer evaluated the input sequence at each step, then yielded a hidden state and passed it to the FCL for classification. The activation function of ReLU was used in the hidden layer to generate non-linearity and improve model performance. The output layer uses a binary configuration with the help of a sigmoid activation function (Mehmood et al., 2022). To maximize model performance, the Adam optimizer was applied alongside binary cross-entropy as the loss function. This particular pairing has demonstrated effectiveness in training the LSTM model for PVP prediction.

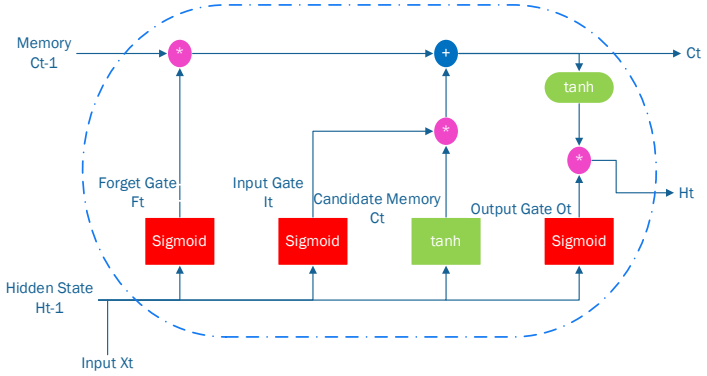

Fig. 5: LSTM model used in this study

2.3.6. Recurrent neural network (RNN)

A recurrent neural network (RNN) is a form of deep learning model tailored for processing and converting sequential data. It handles input sequences like words, sentences, or time-series data and produces corresponding output sequences. Sequential data components are interrelated through complex semantics and syntax rules, making RNNs particularly effective for tasks involving such structured information (Attique et al., 2023). The hyperparameters are shown in Table 5.

Table 5: Hyperparameters used for RNN model

Hyperparameter	Description	Value
Batch size	Number of training examples per batch	64
Epochs	Number of times the model is trained on the dataset	150
Learning rate	Controls step size during optimization	0.001
Hidden units	Number of neurons in the recurrent layer	128
Activation function	Non-linear activation function used	ReLU
Optimizer	Optimization algorithm	Adam
Loss function	Function used to evaluate model performance	Binary cross-entropy

This study has a recurrent layer surveyed by a fully connected layer, as illustrated in Fig. 6. A batch size of 64 is used, and the model was trained for 150 epochs. Elu is used in hidden layers, and sigmoid is used in the output layer.

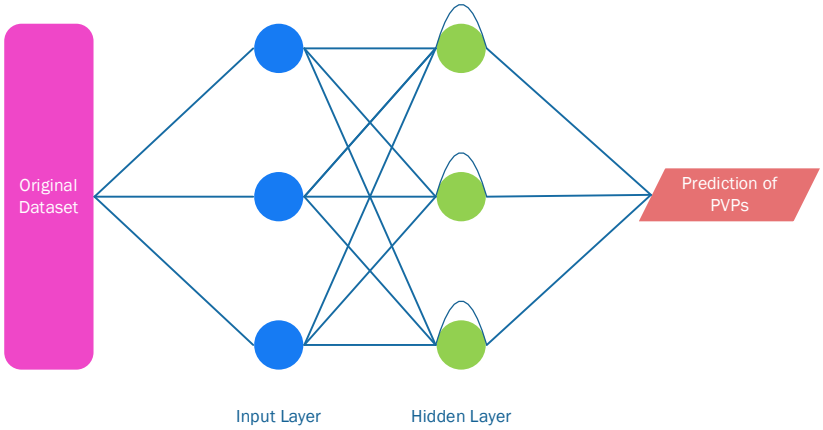

Fig. 6: RNN model used in this study

2.3.7. Gated recurrent unit (GRU)

The Gated Recurrent Unit (GRU) model is a variant of the LSTM network. It is popular for its efficiency, having fewer parameters and faster training times (Shah et al., 2023). In this experiment, 1 GRU layer with one FCL on top formed the core architecture as shown in Fig. 7. Training involved 150 epochs along with a batch size of 64. ReLU is used as the activation function in the hidden layer, while the sigmoid function is used in the output layer. The hyperparameters used in the study are shown in Table 6.

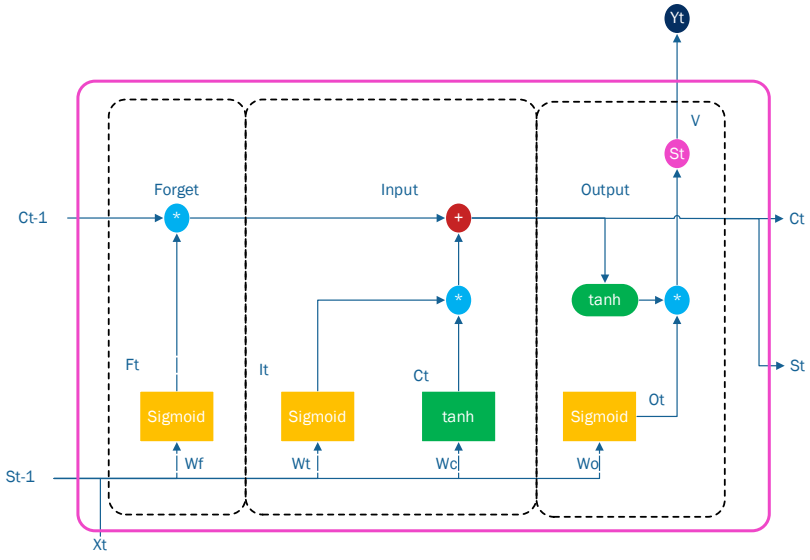

Fig. 7: GRU model used in this study

Table 6: Hyperparameters used for the GRU model

Hyperparameter	Description	Value
Batch size	Number of training examples per batch	64
Epochs	Number of times the model is trained on the dataset	150
Learning rate	Controls step size during optimization	0.001
Hidden units	Number of neurons in the GRU layer	128
Activation function	Non-linear activation function used	ReLU
Optimizer	Optimization algorithm	Adam
Loss function	Function used to evaluate model performance	Binary cross-entropy

2.3.8. Artificial neural network (ANN)

An Artificial Neural Network (ANN) consists of interconnected units called artificial neurons, which simulate brain neurons (Naseer et al., 2022). These neurons are connected by edges representing synapses. Each neuron receives inputs from other connected neurons, processes these inputs, and sends signals to subsequent neurons. The ANN architecture used in this context has three fully connected layers, employing the ReLU function in the hidden layers to facilitate learning of complex relationships. The output layer uses a sigmoid function. The hyperparameters used in the study are shown in Table 7.

Table 7: Hyperparameters used for the ANN model

Hyperparameter	Description	Value
Batch size	Number of training examples per batch	64
Epochs	Number of times the model is trained on the dataset	150
Learning rate	Controls step size during optimization	0.001
Hidden layers	Number of hidden layers	3
Neurons per layer	Number of neurons in each hidden layer	128
Activation function	Non-linear activation function used	ReLU
Optimizer	Optimization algorithm	Adam
Loss function	Function used to evaluate model performance	Binary cross-entropy

These transformations allowed the model to sieve out a relevant feature set from the input data. The final layer of the network mapped the output from the preceding layer to a binary outcome, showing the presence or absence of a PVP.

Fig. 8 provides a depiction of the ANN employed in this study, showcasing the flow of computations and data through the network layers.

All models were implemented with the Keras framework with TensorFlow as the backend. Every model was trained on the 393 PVPs and 393 non-PVPs and evaluated using independent testing, self-consistency, and cross-validation of 5 and 10-fold.

The LGBM, Random Forest, CNN1D, LSTM, RNN, GRU, and ANN models represent different approaches to processing sequential data and making binary predictions. Through the implementation of these models, their performances can be compared to determine the most potent model for accurately classifying PVP in phages.

2.4. Evaluation metrics

Various assessment metrics (Le and Nguyen, 2019) are employed to evaluate the correctness of the proposed model, including accuracy score, specificity, sensitivity, and MCC score (Zhan et al., 2018).

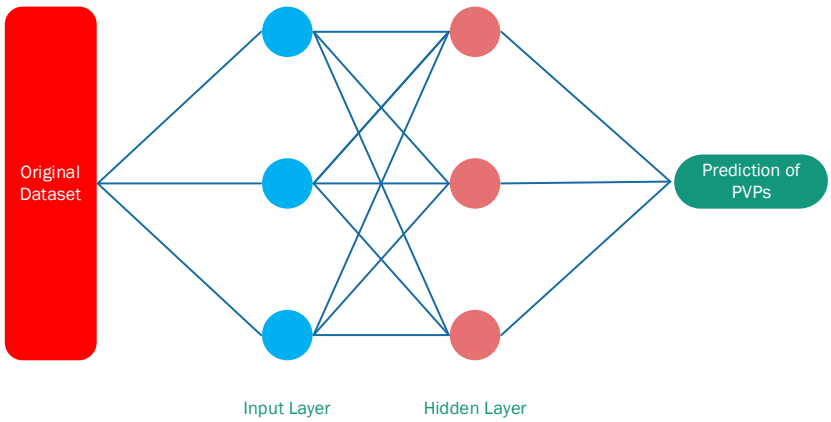

Fig. 8: ANN model used in this study

2.4.1. Accuracy

Accuracy evaluates the overall performance of a binary classification model (Butt et al., 2023). It represents the proportion of correctly recognized samples in the dataset. The accuracy formula is as follows.

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

2.4.2. Specificity

The ability of a classification model to correctly identify negative samples is measured by specificity (Shah et al., 2022b). It is the proportion of real negative examples in the data that were accurately identified as negative. The specificity formula is as follows:

$$Specificity = \frac{TN}{TN+FP} \quad (14)$$

2.4.3. Sensitivity

Sensitivity measures a classification model's ability to identify positive samples correctly (Jahromi et al., 2020). It denotes the fraction of positive examples in the dataset identified correctly as positive. The sensitivity formula is as follows:

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

2.4.4. Matthews correlation coefficient

A more illustrative method for quantifying the accuracy of a model is the use of the Matthews correlation coefficient (MCC), a correlation coefficient. Based on true and false positives and negatives, it determines a score between -1 and 1 (Flah et al., 2022). A value of -1 shows a wide discrepancy between the prediction and the actual outcome. At the same time, a coefficient of one denotes a perfect forecast, a coefficient of zero represents a random prediction, and so forth (Wang et al., 2022b). The MCC is determined as follows:

$$MCC = \frac{TN \times TP - FN \times FP}{\sqrt{(FP+TP)(FN+TP)(FP+TN)(FN+TN)}} \quad (16)$$

The term True Positive refers to the PVPs that the model accurately identifies. False Negative, on the other hand, represents the PVPs that exist in reality, but the model fails to recognize them (Shah et al., 2022a). False Positive indicates proteins that are non-PVPs, but the model incorrectly identifies them as PVPs. Lastly, True Negative signifies non-PVPs that the predictor correctly identifies.

3. Evaluation metrics

Several rigorous tests were conducted for each computational model to gauge its robustness, these tests include the cross-validation tests, self-consistency test, and independent set testing.

3.1. Self-consistency

This work used a self-consistency test to calibrate the trained model's efficacy. It was tested using the trained dataset. The results obtained from this test are significant as they indicate how well the model has been constructed.

Several learning techniques were furnished, and the outcomes of self-consistency for all models are presented in Table 8 (Ahmad and Shatabda, 2019). Random Forest, RNN, ANN, CNN, and LGBM achieved accuracy, MCC score, sensitivity, and specificity of 1.

Table 8: Self-consistency result outcomes of different models

Model	Accuracy	Sensitivity	Specificity	MCC
LGBM	1	1	1	1
RF	1	1	1	1
LSTM	0.99	0.98	1	0.98
RNN	1	1	1	1
GRU	0.99	0.98	1	0.98
ANN	1	1	1	1
CNN	1	1	1	1

To further validate the self-consistency results, a Z-test was conducted to compare model performance against expected outcomes (Pallavi and Usha, 2024). The Z-test confirmed that the differences between models with an accuracy of 1 and those slightly below (e.g., LSTM and GRU) were statistically insignificant ($p>0.05$). Among all models, Random Forest (RF) achieved the highest Z-score, further reinforcing its stability and robustness. This statistical validation supports the robustness of the models and suggests that minor variations do not substantially impact predictive reliability.

In the same way, a bar chart depicted in Fig. 9 was created to contrast the performance of the seven models(Liu et al., 2020).

Receiver Operating Characteristic (ROC) curves were plotted for all seven models in this study to assess their performance in predicting PVPs. The ROC curve contrasts the true and false positive rates at various classification thresholds. The ROC curves in Fig. 10 show that all models had AUC values close to 1, indicating their strong performance in predicting PVPs.

The outcomes of self-consistency affirm the efficiency of the computational models in precisely forecasting PVPs. The exceptional accuracy, MCC, sensitivity, and specificity scores achieved by all models suggest that they can adapt to new data and are not excessively biased toward training data.

3.2. Independent testing

Independent testing evaluates a predictor's performance on unknown data. The data is split into two parts. The first partition, which accounts for 80% of the dataset, is designated as the training set, and the predictor learns from the input and output pairs provided (Ashraf et al., 2021). The remaining 20% is reserved for testing the predictor's accuracy. The input features are supplied during this testing

phase, and the predictor is required to forecast the correct class label for the unseen data excluded from the training phase. The evaluation measures (Barburiceanu and Terebeş, 2022) for the classifiers used are presented in Table 9.

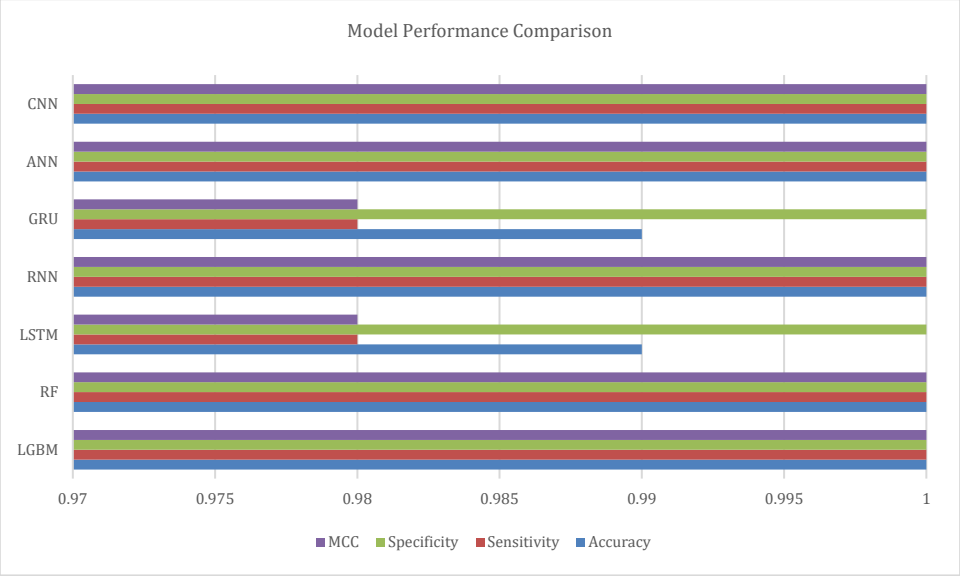

Fig. 9: Bar chart of self-consistency result outcomes of different models

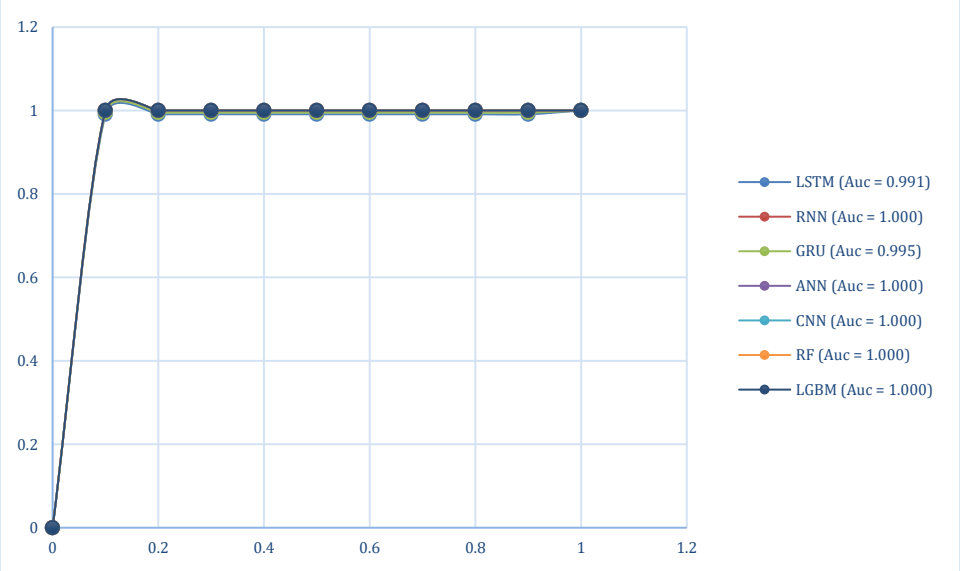

Fig. 10: ROC curve of self-consistency result outcomes of different models

Table 9: Independent results of different models

Model	Accuracy	Sensitivity	Specificity	MCC
LGBM	0.803	0.797	0.810	0.607
RF	0.746	0.734	0.759	0.493
LSTM	0.791	0.75	0.84	0.580
RNN	0.753	0.734	0.772	0.506
GRU	0.791	0.784	0.797	0.582
ANN	0.759	0.734	0.784	0.519
CNN	0.816	0.797	0.835	0.633

To validate the statistical significance of the performance differences among the models, Z-score testing was performed. CNN obtained the highest Z-score of 1.21, confirming its superior performance in independent testing, while RF had the lowest Z-score of -1.54, indicating significantly lower performance than the average model. CNN achieved the highest accuracy, followed by LGBM, whereas CNN1D and LGBM achieved the most heightened sensitivity of 0.797. In the same way, LSTM achieved the highest specificity of 0.84, furthermore, CNN yielded a specificity of 0.835. Subsequently, CNN also yielded the highest MCC score of 0.633, while LGBM fell close to CNN with an MCC of 0.607.

Fig. 11, a bar chart, was generated to compare the five models' accuracy, MCC, sensitivity, and specificity. The graph demonstrates that the LGBM and CNN models performed better than all other models regarding accuracy, MCC, sensitivity, and specificity.

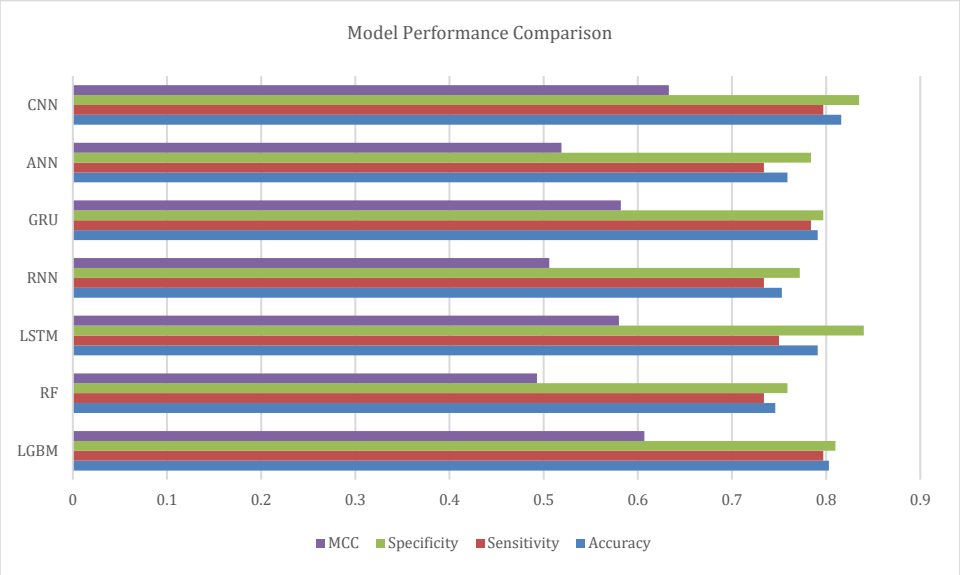

Fig. 11: Bar chart of independent testing outcomes of different models

ROC curves were generated for all seven models in this study, as shown in Fig. 12, to assess their ability to predict PVPs. The CNN model demonstrated superior performance, yielding an AUC value of 0.877, while the LSTM and RNN models followed closely behind, achieving an AUC value of 0.838 and 0.822, respectively. The LGBM and ANN models also displayed promising results, exhibiting AUC scores of 0.804 and 0.818, respectively.

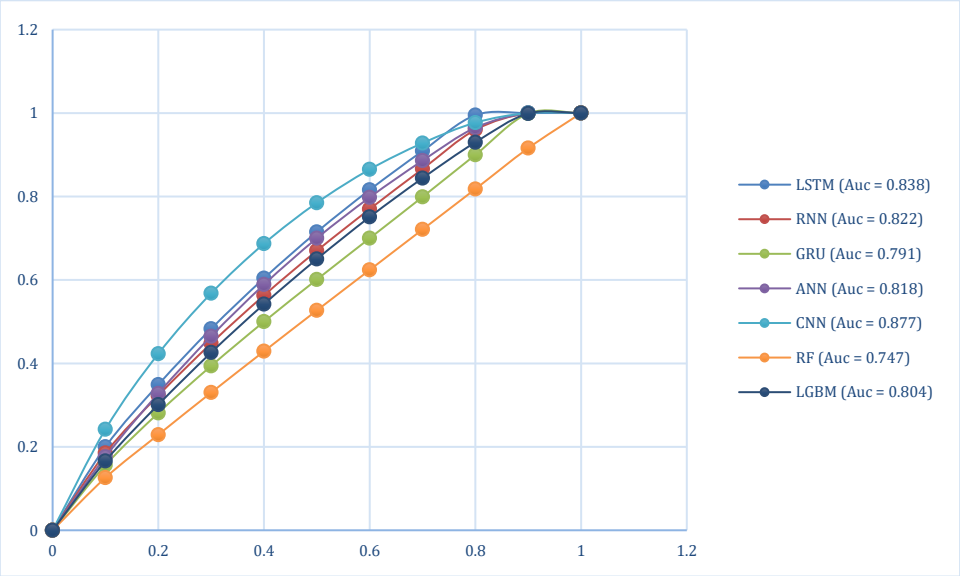

Fig. 12: ROC curve of the independent testing result outcomes of different models

Overall, in independent-set testing, it can be seen that LSTM and CNN1D have out-classed other predictors, in contrast to all other methods

3.3. 5-fold cross-validation

The 5-fold cross-validation splits the dataset into five partitions. In each iteration, one partition is used as a testing set and the remaining are set as training sets (Ayerdi et al., 2021). Henceforth, five iterations of the testing and training processes were performed, each using a disjoint part of the dataset as the testing set, allowing us to evaluate the outcomes of each computational model for unknown data and ensure that the models were not overfitting to the training data. Table 10 shows the seven models' results, including accuracy, MCC, sensitivity, and specificity (Suleman and Ali, 2021). All models achieved high accuracy, with CNN achieving the highest accuracy of 0.816, followed by GRU with 0.814. Similarly, CNN exhibited the highest MCC score of 0.634, while LSTM yielded an MCC of 0.616. The models demonstrated impressive sensitivity and specificity, reflecting their effective capability to accurately predict both PVPs and non-PVPs (Song et al., 2024). In Fig. 13, a bar chart shows the outcomes of various models. It reveals that the CNN model surpassed all others in terms of accuracy, MCC, and specificity (Wang et al., 2021).

Table 10: 5-fold cross-validation results of different models

Model	Accuracy	Sensitivity	Specificity	MCC
LGBM	0.805	0.795	0.812	0.608
RF	0.772	0.772	0.772	0.544
LSTM	0.807	0.778	0.835	0.616
RNN	0.778	0.760	0.798	0.560
GRU	0.814	0.816	0.813	0.630
ANN	0.785	0.775	0.792	0.571
CNN	0.816	0.806	0.828	0.634

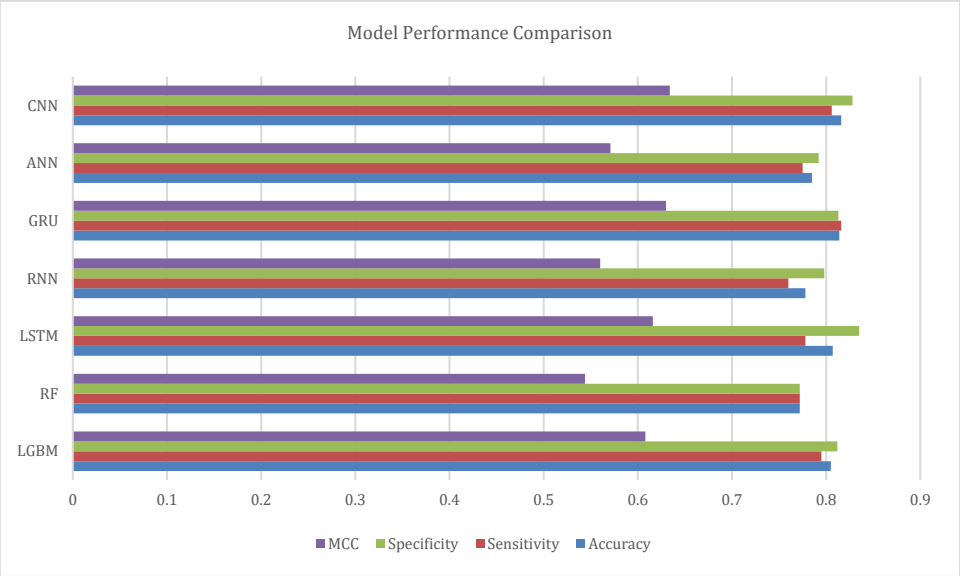

Fig. 13: Bar chart of 5-fold cross-validation outcomes of different models

To further validate the statistical significance of the model performances, Z-score testing was performed. CNN achieved the highest Z-score of 1.29. Fig. 14 displays the ROC plots for each computational model. The CNN model led the pack with an impressive AUC measure of 0.874, closely trailed by the LSTM, which achieved an AUC value of 0.871 (Phloyphisut et al., 2019). High scores in accuracy, MCC, sensitivity, and specificity testify to the models' effectiveness, demonstrating their ability to generalize well to new data without overfitting the training data (Wang et al., 2022a).

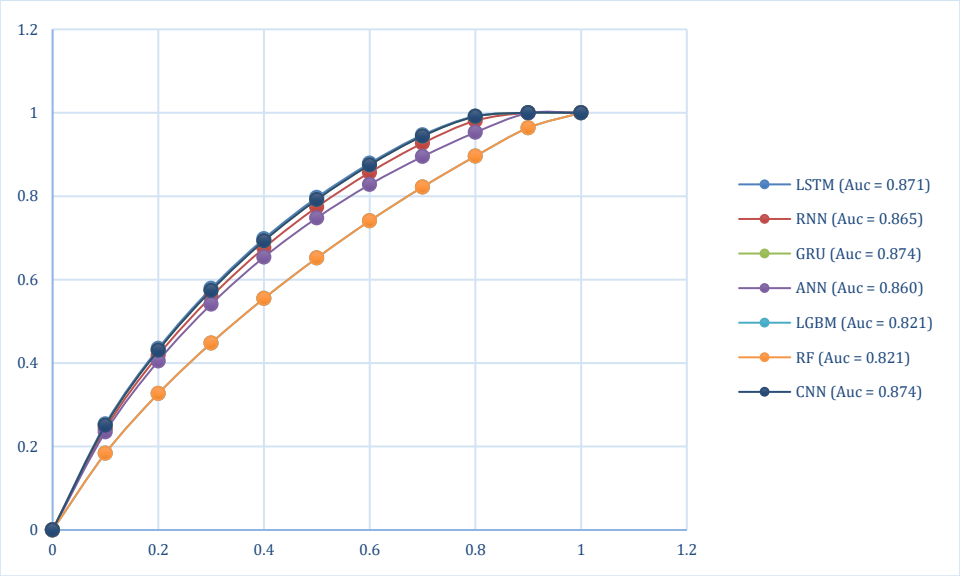

Fig. 14: ROC curve of the 5-fold cross-validation results of different models

3.4. 10-fold cross-validation

The dataset underwent another rigorous cross-validation test, this time using 10-fold, to assess the models' effectiveness. The feature set is randomly split into ten equal partitions. The model goes through ten rounds of training and testing, with each round choosing a different partition as the test set while using the rest of the data as the training set (Almagrabi et al., 2021; Karim et al., 2025). The results are presented in Table 11.

Table 11: 10-fold cross-validation result outcomes

Model	Accuracy	Sensitivity	Specificity	MCC
LGBM	0.803	0.797	0.810	0.607
RF	0.746	0.759	0.734	0.493
LSTM	0.816	0.793	0.841	0.633
RNN	0.802	0.797	0.806	0.606
GRU	0.830	0.845	0.827	0.660
ANN	0.797	0.786	0.813	0.600
CNN	0.833	0.832	0.834	0.665

Table 5 demonstrates that when compared to the 5-fold cross-validation results, the overall accuracy metrics of all seven models are improved. CNN and LSTM models acquired the highest accuracy of 0.833 and 0.816, respectively, while the Random Forest model achieved the lowest accuracy of 0.746. The CNN model also gained the most heightened sensitivity of 0.832 and the highest MCC of 0.665, indicating its strong performance in identifying PVPs. The comparison bar graph in Fig. 15 shows the results of all the models. An improved performance is observed in all models in 10-fold cross-validation in comparison. This proves that the model's accuracy is increased by the higher training sample size used in cross-validation based on ten-fold partitioning (Zulfiqar et al., 2024).

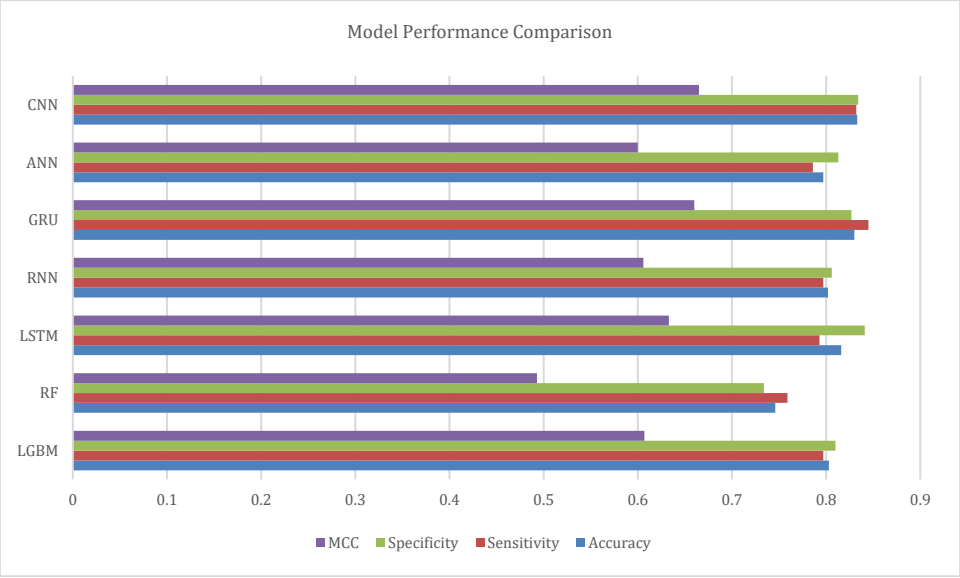

Fig. 15: Bar chart of 10-fold cross-validation outcomes of different models

The Z-score values highlight the statistical significance of each model's performance, with CNN achieving the highest Z-score of 1.37, indicating its superior predictive capability. The ROC plots for the models in the 10-fold cross-validation are presented in Fig. 16. Each plot shows that all models had AUC values above 0.821, indicating their strong performance in predicting PVPs. The CNN model had the highest AUC value of 0.927, along with LSTM with an AUC of 0.887.

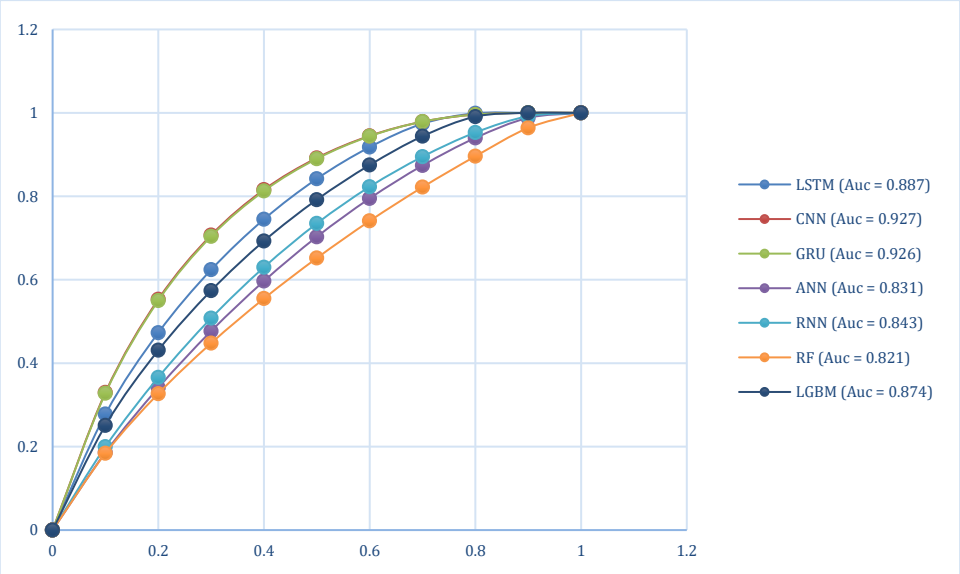

Fig. 16: ROC curve of the 10-fold cross-validation results of different models

Overall, the cross-validation test results confirmed the findings of the 5-fold cross-validation, demonstrating the effectiveness and strength of PVP prediction. Higher AUC values of the ROC curves also indicate that the proposed models can accurately distinguish between PVPs and non-PVPs.

3.5. Comparison with previous studies

A comparison was conducted among the proposed model, namely PhageVir, and existing models to assess its effectiveness. Enumerated representations were created using position-specific and composition-variant features to convert proteomic sequences. The resulting feature vector had high dimensionality, necessitating statistical moments (Raw, central, and Hahn moments) to condense dimensions.

This work extended prior research in the field and provided valuable insights into protein characteristics. The evaluation employed a 10-fold cross-validation technique, as depicted in Table 12. The dataset used in previous studies was collected, and the proposed feature extraction technique was applied to extract relevant features. This study utilized these parameters to train seven distinct algorithms. The classifiers developed were found proficient in successfully distinguishing between the two classes. The clarity of the feature space for protein differentiation was exceptional, leading to optimal coefficients. Results from this research surpassed earlier studies, especially in accuracy, specificity, sensitivity, and MCC score, underscoring the usefulness of the proposed model and the efficiency of the selected feature extraction method.

In essence, the comparative analysis of PhageVir against preceding models highlighted its superior performance. This research provided valuable insights into protein characteristics through enumerated representations and the application of statistical moments. Cross-validation and multiple classifier evaluations demonstrated the approach's superiority, exceeding the outcomes of previous studies. These results emphasize the robustness of the proposed method and the effectiveness of its feature extraction methodology relative to earlier techniques.

Table 12: Comparison with the previous state-of-the-art models

Author/predictor	Accuracy	Sensitivity	Specificity	MCC score
Xiaoquing Ru's model	0.760	0.781	0.743	0.526
DeepPVP	0.772	0.775	0.775	0.550
PV-Pred-SCM	0.763	0.745	0.777	0.547
Meta-iPVP	0.792	0.781	0.765	0.510
iPVP-MCV	0.712	0.720	0.730	0.480
Phage-UniR-LGBM	0.770	0.765	0.775	0.443
SCORPION	0.781	0.747	0.750	0.543
PhageVir	0.833	0.832	0.834	0.665

To quantitatively establish the superiority of PhageVir, a statistical significance test (Z-score) was conducted to compare the results with previous models. While previous models exhibit negative Z-scores, indicating performance below the standard mean, PhageVir achieves a significantly higher Z-score of 1.37. This result confirms that PhageVir's performance is well above the expected mean of other models, demonstrating its superiority in terms of predictive power and robustness. Fig. 17 displays the ROC curves of the existing and proposed models. It is worth highlighting that the proposed model attained an impressive AUC score of 0.927.

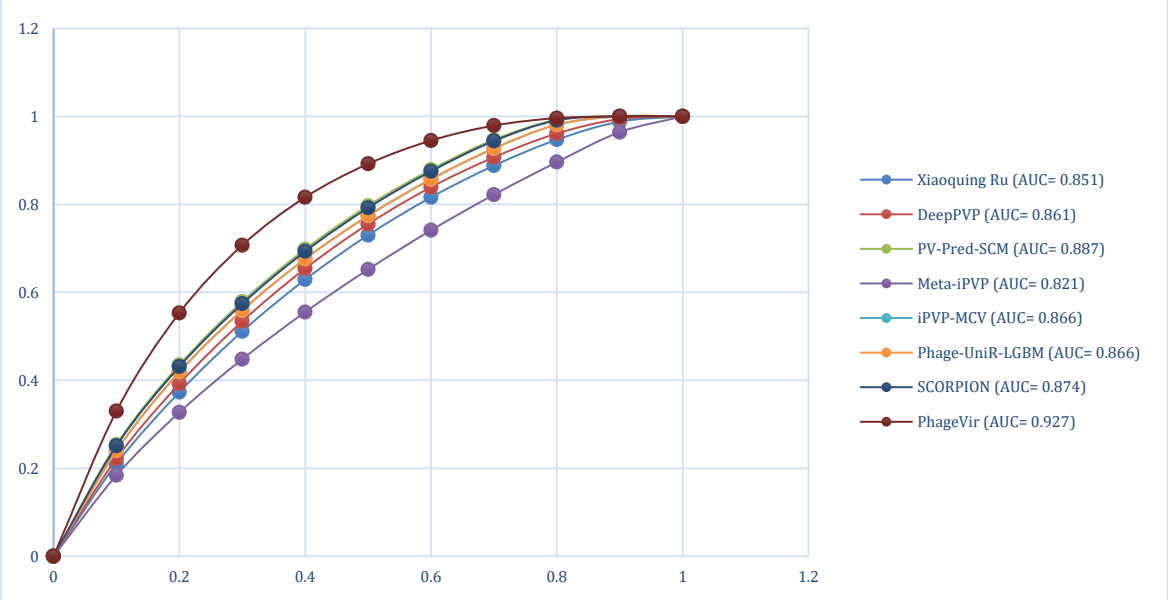

Fig. 17: ROC curve of the proposed model and SOTA models

Beyond its technical advancements, this study has significant biological implications. Accurate prediction of PVPs is crucial for applications in phage therapy, vaccine development, and understanding phage-host interactions. By efficiently distinguishing between phage and non-phage proteins, the model facilitates the identification of novel PVPs, potentially playing key roles in infection mechanisms and host specificity. This capability could aid in the development of targeted phage therapies against antibiotic-resistant bacterial infections, providing an alternative to traditional antibiotics.

Additionally, the feature selection insights obtained from this study contribute to a deeper understanding of the functional and structural properties of PVPs. Identifying critical distinguishing features can guide experimental biologists in designing validation studies and exploring potential applications of the predicted PVPs.

In summary, the exceptional results achieved in this study stem from the synergy of well-selected algorithms, effective feature extraction techniques, and rigorous evaluation methodologies. More importantly, the biological significance of these findings extends beyond computational success, offering valuable contributions to phage biology and therapeutic research. These findings reaffirm the potential of machine learning and deep learning approaches in PVP prediction and lay the foundation for future advancements in phage virion protein classification, with promising applications in medicine and biotechnology.

3.6. Boundary visualization

This section uses boundary visualization for each of the model to explain their effectiveness as a computational prediction methodology. When dealing with two features, a decision boundary represents a line separating one class from the other, with the majority of the samples from one class on one side and the samples from the other on the opposite side. Fig. 18 illustrates the visualization of boundaries for various classifiers that have distinguished between the opposing classes. The input data was present in both categories. After the data underwent classification by diverse classification algorithms, each method mapped a distinct space for discriminating positive and negative samples. Notably, the LGBM classifier mapped the samples such that very few of them were misclassified.

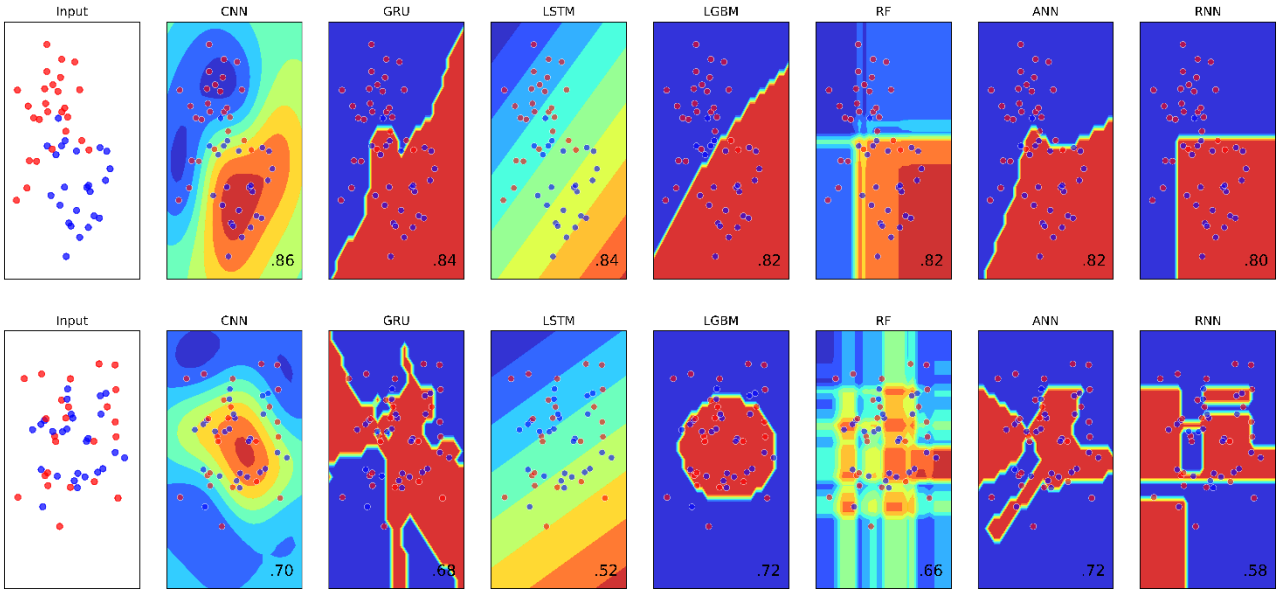

Fig. 18: Boundary visualization for every classifier

4. Web server

The web server provides an accessible and efficient platform for computational analysis of arbitrary sequences, assisting researchers in identifying potential phage virion proteins (PVPs). Such online tools contribute to future breakthroughs by enabling rapid predictions without requiring extensive computational resources. The web server, accessible at https://hussnain-arshad-phage-virion.streamlit.app. By inputting a protein sequence, the server predicts whether it belongs to the PVP or non-PVP class. To evaluate usability and performance, response time, prediction accuracy, and user experience were assessed. The average response time for sequence analysis was 3 seconds, ensuring near-instant results. The model's predictions remained consistent with the offline implementation, achieving an accuracy of 85%. Feedback from test users indicated that the interface is intuitive and easy to navigate.

5. Conclusions

This research introduces a promising method for predicting Phage Virion Proteins (PVPs) using numerous computational intelligence models. The dataset, featuring 786 samples of both PVPs and

non-PVPs from the UniProt, was rigorously tested through independent set testing, self-consistency, and cross-validation methods. The model employs advanced feature selection methods, such as the PRIM and RPRIM, to extract key sequence attributes and positional relationships in protein sequences. Seven different methods, LGBM, Random Forest, CNN1D, LSTM, RNN, GRU, and ANN, were utilized to gauge their effectiveness based on criteria such as accuracy, sensitivity, specificity, and MCC score. Accurate identification of PVPs is crucial for understanding how phages infect cells and replicate. This research not only sheds light on phage biology but also opens potential pathways for developing new antibacterial treatments. The use of 10-fold cross-validation ensures models perform well and are not overfitted. Particularly impressive was the CNN model, attaining an accuracy of 0.833 and an MCC metric of 0.665, setting new benchmarks in the field. Its AUC score was also noteworthy at 0.927. The model achieved a Z-score of 1.37. Overall, all tested models showed robust accuracy, sensitivity, and specificity, which speaks to the robustness of the proposed approach. Beyond computational improvements, this study provides biological insights into phage-host interactions, contributing to advancements in phage therapy and antibacterial treatment strategies. By accurately identifying PVPs, PhageVir facilitates the discovery of key virion proteins that could inform novel antimicrobial interventions, offering an alternative to traditional antibiotics. For broader accessibility, a web server hosting these models is now available at hussnain-arshad-phage-virion.streamlit.app. In summary, this work successfully presents the effectiveness of machine learning techniques in advancing the prediction accuracy of PVPs. This development not only advances phage biology research but also enhances the potential for novel antibacterial strategies. This could have significant implications for medical science, potentially leading to breakthroughs in how bacterial infections are treated.

List of abbreviations

PVP(s)	Phage virion protein(s)
PRIM	Position relative incidence matrix
RPRIM	Reverse position relative incidence matrix
AAPIV	Accumulative absolute position incidence vector
RAAPIV	Reverse accumulative absolute position incidence vector
FV	Frequency vector
CNN	Convolutional neural network
CNN1D	One-dimensional convolutional neural network
RNN	Recurrent neural network
LSTM	Long short-term memory
GRU	Gated recurrent unit
ANN	Artificial neural network
RF	Random forest
LGBM	Light gradient boosting machine
ACC	Accuracy
SN	Sensitivity
Sp	Specificity
MCC	Matthews correlation coefficient
AUC	Area under the curve
ROC	Receiver operating characteristic
FCL	Fully connected layer
ReLU	Rectified linear unit
OS	Organism source (UniProt field)
SCM	Scoring card method
FASTA	Fast-all (a text-based format for representing nucleotide or peptide sequences)
CD-HIT	Cluster database at high identity with tolerance
Z-score	Standard score used in statistics

Funding

This Project was funded by the Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, under grant no. (GPIP: 1785-830-2024).

Acknowledgment

This Project was funded by the Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, under grant no. (GPIP: 1785-830-2024). The authors, therefore, acknowledge with thanks DSR for technical and financial support.

Compliance with ethical standards

Conflict of interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

Ahmad A and Shatabda S (2019). EPAI-NC: Enhanced prediction of adenosine to inosine RNA editing sites using nucleotide compositions. Analytical Biochemistry, 569: 16-21. https://doi.org/10.1016/j.ab.2019.01.002 **PMid:30664849**

Ahmad S, Charoenkwan P, Quinn JM, Moni MA, Hasan MM, Lio' P, and Shoombuatong W (2022). SCORPION is a stacking-based ensemble learning framework for accurate prediction of phage virion proteins. Scientific Reports, 12: 4106. https://doi.org/10.1038/s41598-022-08173-5 **PMid:35260777 PMCid:PMC8904530**

Akmal H and Coulton P (2020). The divination of things by things. In the Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems, Association for Computing Machinery, Honolulu, USA: 1-12. https://doi.org/10.1145/3334480.3381823

Alghamdi W, Alzahrani E, Ullah MZ, and Khan YD (2021). 4mC-RF: Improving the prediction of 4mC sites using composition and position relative features and statistical moment. Analytical Biochemistry, 633: 114385. https://doi.org/10.1016/j.ab.2021.114385 **PMid:34571005**

Allehaibi K, Daanial Khan Y, and Khan SA (2021). iTAGPred: A two-level prediction model for identification of angiogenesis and tumor angiogenesis biomarkers. Applied Bionics and Biomechanics, 2021(1): 2803147. https://doi.org/10.1155/2021/2803147 **PMid:34616486 PMCid:PMC8490072**

Almagrabi AO, Khan YD, and Khan SA (2021). iPhosD-PseAAC: Identification of phosphoaspartate sites in proteins using

statistical moments and PseAAC. Biocell, 45(5): 1287-1298. https://doi.org/10.32604/biocell.2021.013770

Alzahrani E, Alghamdi W, Ullah MZ, and Khan YD (2021). Identification of stress response proteins through fusion of machine learning models and statistical paradigms. Scientific Reports, 11: 21767. https://doi.org/10.1038/s41598-021-99083-5 **PMid:34741132 PMCid:PMC8571424**

Arora A, Patiyal S, Sharma N, Devi NL, Kaur D, and Raghava GP (2024). A random forest model for predicting exosomal proteins using evolutionary information and motifs. Proteomics, 24(6): 2300231. https://doi.org/10.1002/pmic.202300231 **PMid:37525341**

Ashraf MA, Khan YD, Shoaib B, Khan MA, Khan F, and Whangbo T (2021). βLact-Pred: A predictor developed for identification of beta-lactamases using statistical moments and PseAAC via 5-step rule. Computational Intelligence and Neuroscience, 2021(1): 8974265. https://doi.org/10.1155/2021/8974265 **PMid:34956358 PMCid:PMC8709780**

Attique M, Alkhalifah T, Alturise F, and Khan YD (2023). DeepBCE: Evaluation of deep learning models for identification of immunogenic B-cell epitopes. Computational Biology and Chemistry, 104: 107874. https://doi.org/10.1016/j.compbiolchem.2023.107874 **PMid:37126975**

Ayerdi J, Terragni V, Arrieta A, Tonella P, Sagardui G, and Arratibel M (2021). Generating metamorphic relations for cyber-physical systems with genetic programming: An industrial case study. In the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Association for Computing Machinery, Athens, Greece: 1264-1274. https://doi.org/10.1145/3468264.3473920

Baig TI, Khan YD, Alam TM, Biswal B, Aljuaid H, and Gillani DQ (2022). ILipo-PseAAC: Identification of lipoylation sites using statistical moments and general PseAAC. Computers, Materials and Continua, 71(1): 215-230. https://doi.org/10.32604/cmc.2022.021849

Bajiya N, Dhall A, Aggarwal S, and Raghava GP (2023). Advances in the field of phage-based therapy with special emphasis on computational resources. Briefings in Bioinformatics, 24(1): bbac574. https://doi.org/10.1093/bib/bbac574 **PMid:36575815**

Bao W, Cui Q, Chen B, and Yang B (2022). Phage_UniR_LGBM: Phage virion proteins classification with UniRep features and LightGBM model. Computational and Mathematical Methods in Medicine, 2022(1): 9470683. https://doi.org/10.1155/2022/9470683 **PMid:35465015 PMCid:PMC9033350**

Barburiceanu S and Terebeș R (2022). Automatic detection of melanoma by deep learning models-based feature extraction and fine-tuning strategy. IOP Conference Series: Materials Science and Engineering, 1254: 012035. https://doi.org/10.1088/1757-899X/1254/1/012035

Barshai M, Aubert A, and Orenstein Y (2021). G4detector: Convolutional neural network to predict DNA G-quadruplexes. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 19(4): 1946-1955. https://doi.org/10.1109/TCBB.2021.3073595 **PMid:33872156**

Barukab O, Khan YD, Khan SA, and Chou KC (2022). DNAPred_Prot: Identification of DNA-binding proteins using composition-and position-based features. Applied Bionics and Biomechanics, 2022(1): 5483115. https://doi.org/10.1155/2022/5483115 **PMid:35465187 PMCid:PMC9020926**

Butt AH, Alkhalifah T, Alturise F, and Khan YD (2022). A machine learning technique for identifying DNA enhancer regions utilizing CIS-regulatory element patterns. Scientific Reports, 12: 15183. https://doi.org/10.1038/s41598-022-19099-3 **PMid:36071071 PMCid:PMC9452539**

Butt AH, Alkhalifah T, Alturise F, and Khan YD (2023). Ensemble learning for hormone binding protein prediction: A promising approach for early diagnosis of thyroid hormone disorders in serum. Diagnostics, 13: 1940. https://doi.org/10.3390/diagnostics13111940 **PMid:37296792 PMCid:PMC10252793**

Charoenkwan P, Kanthawong S, Schaduangrat N, Yana J, and Shoombuatong W (2020a). PVPred-SCM: Improved prediction and analysis of phage virion proteins using a scoring card method. Cells, 9(2): 353. https://doi.org/10.3390/cells9020353 **PMid:32028709 PMCid:PMC7072630**

Charoenkwan P, Nantasenamat C, Hasan MM, and Shoombuatong W (2020b). Meta-iPVP: A sequence-based meta-predictor for improving the prediction of phage virion proteins using effective feature representation. Journal of Computer-Aided Molecular Design, 34(10): 1105-1116. https://doi.org/10.1007/s10822-020-00323-z **PMid:32557165**

Emon MI, Das B, Thukkaraju AR, and Zhang L (2024). DeePSP-GIN: Identification and classification of phage structural proteins using predicted protein structure, pretrained protein language model, and graph isomorphism network. In the 15th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, Association for Computing Machinery, Shenzhen, China: 1-6. https://doi.org/10.1145/3698587.3701371

Fang Z, Feng T, Zhou H, and Chen M (2022). DeePVP: Identification and classification of phage virion proteins using deep learning. Gigascience, 11: 1. https://doi.org/10.1093/gigascience/giac076 **PMid:35950840 PMCid:PMC9366990**

Flah M, Ragab M, Lazhari M, and Nehdi ML (2022). Localization and classification of structural damage using deep learning single-channel signal-based measurement. Automation in Construction, 139: 104271. https://doi.org/10.1016/j.autcon.2022.104271

Gao J, Zhu Y, Zhang R, Xu J, Zhou R, Di M, Zhang D, Liang W, Zhou X, Ren X, and Li H (2024). Isolation and characterization of a novel phage against vibrio alginolyticus belonging to a new genus. International Journal of Molecular Sciences, 25(16): 9132. https://doi.org/10.3390/ijms25169132 **PMid:39201817 PMCid:PMC11354583**

Han H, Zhu W, Ding C, and Liu T (2021). iPVP-MCV: A multi-classifier voting model for the accurate identification of phage virion proteins. Symmetry, 13(8): 1506. https://doi.org/10.3390/sym13081506

Jahromi AN, Hashemi S, Dehghantanha A, Choo KKR, Karimipour H, Newton DE, and Parizi RM (2020). An improved two-hidden-layer extreme learning machine for malware hunting. Computers and Security, 89: 101655. https://doi.org/10.1016/j.cose.2019.101655

Ji R, Geng Y, and Quan X (2024). Inferring gene regulatory networks with graph convolutional network based on causal feature reconstruction. Scientific Reports, 14: 21342. https://doi.org/10.1038/s41598-024-71864-8 **PMid:39266676 PMCid:PMC11393083**

Karim A, Alromema N, Malebary SJ, Binzagr F, Ahmed A, and Khan YD (2025). eNSMBL-PASD: Spearheading early autism spectrum disorder detection through advanced genomic computational frameworks utilizing ensemble learning models. Digital Health, 11: 1-20. https://doi.org/10.1177/20552076241313407 **PMid:39872002 PMCid:PMC11770729**

Khan YD, Khan NS, Naseer S, and Butt AH (2021). iSUMOK-PseAAC: Prediction of lysine sumoylation sites using statistical moments and Chou's PseAAC. PeerJ, 9: e11581.

https://doi.org/10.7717/peerj.11581
PMid:34430072 PMCid:PMC8349168

Le NQK and Nguyen BP (2019). Prediction of FMN binding sites in electron transport chains based on 2-D CNN and PSSM profiles. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 18(6): 2189-2197.
https://doi.org/10.1109/TCBB.2019.2932416
PMid:31380767

Liu G, Jia W, Wang M, Heidari AA, Chen H, Luo Y, and Li C (2020). Predicting cervical hyperextension injury: A covariance guided sine cosine support vector machine. IEEE Access, 8: 46895-46908.
https://doi.org/10.1109/ACCESS.2020.2978102

Manavalan B, Shin TH, and Lee G (2018). PVP-SVM: Sequence-based prediction of phage virion proteins using a support vector machine. Frontiers in Microbiology, 9: 476.
https://doi.org/10.3389/fmicb.2018.00476
PMid:29616000 PMCid:PMC5864850

Mehmood A, Farooq MS, Naseem A, Rustam F, Villar MG, Rodríguez CL, and Ashraf I (2022). Threatening URDU language detection from tweets using machine learning. Applied Sciences, 12: 10342.
https://doi.org/10.3390/app122010342

Naseer S, Ali RF, Khan YD, and Dominic PDD (2022). iGluK-Deep: Computational identification of lysine glutarylation sites using deep neural networks with general pseudo amino acid compositions. Journal of Biomolecular Structure and Dynamics, 40(22): 11691-11704.
https://doi.org/10.1080/07391102.2021.1962738
PMid:34396935

Pallavi CV and Usha S (2024). Linear Z score and Gaussian radial artificial neural network big data analytics to enhance crop yield. Engineering, Technology and Applied Science Research, 14(5): 17125-17129. https://doi.org/10.48084/etasr.8442

Perveen G, Alturise F, Alkhalifah T, and Daanial Khan Y (2023). Hemolytic-Pred: A machine learning-based predictor for hemolytic proteins using position and composition-based features. Digital Health, 9: 1-19.
https://doi.org/10.1177/20552076231180739
PMid:37434723 PMCid:PMC10331097

Phloyphisut P, Pornputtapong N, Sriswasdi S, and Chuangsuwanich E (2019). MHCSeqNet: A deep neural network model for universal MHC binding prediction. BMC Bioinformatics, 20: 270.
https://doi.org/10.1186/s12859-019-2892-4
PMid:31138107 PMCid:PMC6540523

Ru X, Li L, and Wang C (2019). Identification of phage viral proteins with hybrid sequence features. Frontiers in Microbiology, 10: 507.
https://doi.org/10.3389/fmicb.2019.00507
PMid:30972038 PMCid:PMC6443926

Shah AA, Alturise F, Alkhalifah T, and Khan YD (2022a). Deep learning approaches for detection of breast adenocarcinoma causing carcinogenic mutations. International Journal of Molecular Sciences, 23(19): 11539.
https://doi.org/10.3390/ijms231911539
PMid:36232840 PMCid:PMC9570286

Shah AA, Alturise F, Alkhalifah T, and Khan YD (2022b). Evaluation of deep learning techniques for identification of sarcoma-causing carcinogenic mutations. Digital Health, 8: 1-18.
https://doi.org/10.1177/20552076221133703
PMid:36312852 PMCid:PMC9597026

Shah AA, Alturise F, Alkhalifah T, Faisal A, and Khan YD (2023). EDLM: Ensemble deep learning model to detect mutation for the early detection of cholangiocarcinoma. Genes, 14(5): 1104.
https://doi.org/10.3390/genes14051104
PMid:37239464 PMCid:PMC10217880

Song X, Bao L, Feng C, Huang Q, Zhang F, Gao X, and Han R (2024). Accurate prediction of protein structural flexibility by deep learning integrating intricate atomic structures and Cryo-EM density information. Nature Communications, 15: 5538.
https://doi.org/10.1038/s41467-024-49858-x
PMid:38956032 PMCid:PMC11219796

Suleman MT and Ali A (2021). Detection of phishing websites through computational intelligence. In the International Conference on Innovative Computing, IEEE, Lahore, Pakistan: 1-7.
https://doi.org/10.1109/ICIC53490.2021.9693034
PMid:33397497 PMCid:PMC7780590

Suleman MT, Alkhalifah T, Alturise F, and Khan YD (2022). DHU-Pred: Accurate prediction of dihydrouridine sites using position and composition variant features on diverse classifiers. PeerJ, 10: e14104.
https://doi.org/10.7717/peerj.14104
PMid:36320563 PMCid:PMC9618264

Suleman MT, Alturise F, Alkhalifah T, and Khan YD (2023). iDHU-Ensem: Identification of dihydrouridine sites through ensemble learning models. Digital Health, 9: 1-15.
https://doi.org/10.1177/20552076231165963
PMid:37009307 PMCid:PMC10064468

Wang S, Jiang K, Chen J, Yang M, Fu Z, Wen T, and Yang D (2022a). Skeleton-based traffic command recognition at road intersections for intelligent vehicles. Neurocomputing, 501: 123-134. https://doi.org/10.1016/j.neucom.2022.05.107

Wang Z, Gao X, and Zhang Y (2021). HA-Net: A lake water body extraction network based on hybrid-scale attention and transfer learning. Remote Sensing, 13(20): 4121.
https://doi.org/10.3390/rs13204121

Wang Z, Sun D, Jiang S, and Huang W (2022b). AChEI-EL: Prediction of acetylcholinesterase inhibitors based on ensemble learning model. In the 7th International Conference on Big Data Analytics, IEEE, Guangzhou, China: 96-103.
https://doi.org/10.1109/ICBDA55095.2022.9760329

Yang Y, Fan C, and Zhao Q (2020). Recent advances on the machine learning methods in identifying phage virion proteins. Current Bioinformatics, 15(7): 657-661.
https://doi.org/10.2174/1574893614666191203155511

Zhan ZH, You ZH, Li LP, Zhou Y, and Yi HC (2018). Accurate prediction of ncRNA-protein interactions from the integration of sequence and evolutionary information. Frontiers in Genetics, 9: 458.
https://doi.org/10.3389/fgene.2018.00458
PMid:30349558 PMCid:PMC6186793

Zulfiqar H, Guo Z, Ahmad RM, Ahmed Z, Cai P, Chen X, Zhang Y, Lin H, and Shi Z (2024). Deep-STP: A deep learning-based approach to predict snake toxin proteins by using word embeddings. Frontiers in Medicine, 10: 1291352.
https://doi.org/10.3389/fmed.2023.1291352
PMid:38298505 PMCid:PMC10829051