frontiers | Frontiers in Microbiology

TYPE Original Research
PUBLISHED 04 November 2022
DOI 10.3389/fmicb.2022.1061122

OPEN ACCESS

EDITED BY
Hao Lin,
University of Electronic Science and Technology of China, China

REVIEWED BY
Leyi Wei,
Shandong University, China
Yongqiang Xing,
Inner Mongolia University of Science and Technology, China

*CORRESPONDENCE
Hilal Tayara
hilaltayara@jbnu.ac.kr
Kil To Chong
kitchong@jbnu.ac.kr

SPECIALTY SECTION
This article was submitted to Evolutionary and Genomic Microbiology, a section of the journal Frontiers in Microbiology

RECEIVED 04 October 2022
ACCEPTED 18 October 2022
PUBLISHED 04 November 2022

CITATION
Shujaat M, Jin JS, Tayara H and Chong KT (2022) iProm-phage: A two-layer model to identify phage promoters and their types using a convolutional neural network. *Front. Microbiol.* 13:1061122. doi: 10.3389/fmicb.2022.1061122

iProm-phage: A two-layer model to identify phage promoters and their types using a convolutional neural network

Muhammad Shujaat[1], Joe Sung Jin[2], Hilal Tayara[3]* and Kil To Chong[1,4]*

[1]Department of Electronics and Information Engineering, Jeonbuk National University, Jeonju, South Korea, [2]Graduate School of Integrated Energy AI, Jeonbuk National University, Jeonju, South Korea, [3]School of International Engineering and Science, Jeonbuk National University, Jeonju, South Korea, [4]Advances Electronics and Information Research Center, Jeonbuk National University, Jeonju, South Korea

The increased interest in phages as antibacterial agents has resulted in a rise in the number of sequenced phage genomes, necessitating the development of user-friendly bioinformatics tools for genome annotation. A promoter is a DNA sequence that is used in the annotation of phage genomes. In this study we proposed a two layer model called "iProm-phage" for the prediction and classification of phage promoters. Model first layer identify query sequence as promoter or non-promoter and if the query sequence is predicted as promoter then model second layer classify it as phage or host promoter. Furthermore, rather than using non-coding regions of the genome as a negative set, we created a more challenging negative dataset using promoter sequences. The presented approach improves discrimination while decreasing the frequency of erroneous positive predictions. For feature selection, we investigated 10 distinct feature encoding approaches and utilized them with several machine-learning algorithms and a 1-D convolutional neural network model. We discovered that the one-hot encoding approach and the CNN model outperformed based on performance metrics. Based on the results of the 5-fold cross validation, the proposed predictor has a high potential. Furthermore, to make it easier for other experimental scientists to obtain the results they require, we set up a freely accessible and user-friendly web server at http://nsclbio.jbnu.ac.kr/tools/iProm-phage/.

KEYWORDS

DNA promoters, convolutional neural networks, bioinformatics, computational biology, phages

Introduction

Bacteriophages, commonly referred to as phages, are viruses that infect and destroy bacteria (Salmond and Fineran, 2015). The number of sequenced phage genomes has increased exponentially in recent decades, primarily owing to their small size and ability to bacterial infections (Silva and Echeverrigaray, 2012). This richness of genomic data necessitates the development of user-friendly bioinformatics tools to aid biologists in genome analyses. Recognition of regulatory elements is the most difficult phase in phage genome analysis. Promoters are DNA sequences responsible for transcription initiation. These sequences are difficult to identify because they are composed of short, nonconserved components. However, it is essential to comprehend and describe the genetic regulatory networks of phages, which may permit the engineering of improved phages for medicinal or biotechnological applications (Guzina and Djordjevic, 2015).

Several attempts have been made to develop promoter prediction tools for bacterial genomes. The majority of these tools use computational techniques based on-10 and-35 motifs (Sierro et al., 2008; Mishra et al., 2020; Wang et al., 2020). In contrast to these promoters with typical motifs, phage genome promoters are composed of host and phage promoters with varying motifs (Sampaio et al., 2019).

Therefore, existing tools are not suitable for identifying promoters in phages. Computational tools are required to predict promoters in phages. Prediction of phage promoters has seldom been studied. The PHIRE method (Lavigne et al., 2004) systematically scans a bacteriophage genome to determine the frequency of subsequences in a sequence. All sequences are compared, which significantly increases the running time. PromoterHunter (Klucar et al., 2010) is an online tool to identify phage promoters; however, it requires additional information as input, such as weight matrices of the two promoter elements and is limited concerning the size of the input genome sequences. The PhagePromoter tool (Sampaio et al., 2019) can be used to identify promoters across the entire phage genome. It was created using machine learning (ML) methods, such as artificial neural networks or support vector machines, in conjunction with sequence characteristics (size and score of motifs, frequency of adenine and thymine, and free energy value). Additionally, PhagePromoter can distinguish host promoters from phage promoters. However, PhagePromoter has to be used in a deterministic manner with some previous experimental or predictive knowledge, such as phage family, host bacterium species, and phage type (temperature or virulence), which limits the effectiveness of PhagePromoter. DPProm (Wang et al., 2022) is a proposed convolutional neural network (CNN)-based method for predicting phage promoters and their types as phages or hosts. However, the proposed sequence-processing workflow requires a long time for a query sequence.

Significant progress has been achieved in the essential aspects of phage promoter identification, although improvements are required in different aspects. We identified the following shortcomings of prior research:

1. Most of the aforementioned studies only predicted the promoter sequence as phage or non-promoter. Classification of predicted promoter sequences as phages or hosts was rare.
2. Most studies utilized ML models to classify predicted sequences.
3. Not all studies created a user-friendly and publicly available web server, which has proven inconvenient for practical use by experimental scientists.
4. Performance analysis of different feature encoding schemes on different ML and CNN models was not performed.
5. In the previously proposed tools, the number of false positive values for promoter prediction requires further improvement.
6. Previous studies selected non-coding regions as negative dataset, that's makes a very easy task for the classifier on other hand trained model cannot perform well on difficult test datasets.

In this study, we focused on overcoming these drawbacks to improve the prediction capabilities in identifying phage promoters. First, high-quality benchmark datasets were constructed. Subsequently, we extracted the best feature representation vector and model from a variety of encoding techniques, ML, and CNN models. To achieve this, we sequentially fed encoded vector sequences from all encoding methods into various ML and CNN algorithms. Based on performance evaluation, we chose the one-hot encoding technique and CNN algorithm. We investigated the sequence and properties of phage promoters and presented a two-layer model designated "iProm-phage." In the first layer model, the query sequence is identified as a promoter or non-promoter. If it is a promoter sequence, then the second layer classifies the identified sequence as a phage promoter or host promoter. To assess model performance, we measured the accuracy (Acc), sensitivity (Sn), specificity (Sp), and Matthew's correlation coefficient (MCC). All these parameters are frequently used in state-of-the-art methods in computational biology and bioinformatics (Rahman et al., 2019; Ali et al., 2020; Shujaat et al., 2020; Rehman et al., 2021). In addition, we evaluated the model using five-fold cross validation and receiver operating characteristic (ROC) curves. Finally, the iProm-phage web server was built in compliance with the suggested paradigm. The proposed flow diagram of the study is shown in Figure 1.

Materials and methods

Benchmark dataset

While developing an effective biological predictor, it is critical to select an appropriate benchmark dataset to evaluate the proposed predictive model. We prepared separate datasets for

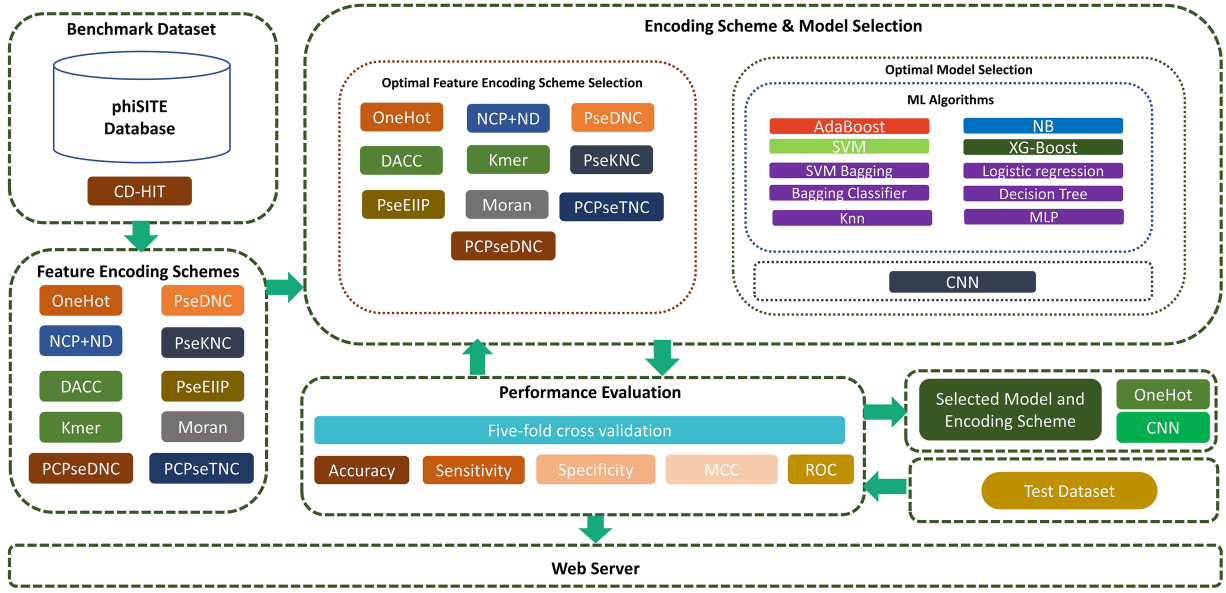

FIGURE 1
Flow diagram of iProm-phage.

TABLE 1 Summary of the Benchmark dataset.

Model Layer	Dataset	Promoter	Non-promoter
First layer	Training	901	901
	Test	198	198
Second layer		**Phage**	**Host**
	Training	111	382
	Test	28	96

each layer of the model, as described in Sections "Dataset for the first layer" and "Dataset for the second layer."

Dataset for the first layer

The promoters of phage genomes have been poorly characterized. Only the phiSITE database has identified the promoters of phage genomes (Klucar et al., 2010). The phage promoter sequence utilized in this study is the same as that used in previous studies (Sampaio et al., 2019; Wang et al., 2022). For the model's first layer, 1,140 promoter sequences from 69 phages were collected and divided into training and test datasets; 901 promoter sequences were utilized as the training dataset and 198 promoter sequences were utilized as the test dataset. Supplementary Table S1 in Supplementary file summarize the promoter sequences from each phage genome.

The selection of a negative dataset is an important step in ensuring model performance. In previous studies, non-promoter regions were randomly selected to build a negative dataset. However, this method tends to be illogical because there is no intersection between positive and negative sets. Consequently, the model immediately detected the key differences between the two groups. Therefore, precision could not be maintained when tested on more difficult datasets. To overcome this problem, we propose a negative dataset generation technique. We created a negative dataset from positive promoter sequences by the following three steps. First, each positive sequence is divided into eight subsequences. Second, five subsequences are randomly selected and placed. Thirdly, the remaining three subsequences are placed at the same position. Using this method, each positive promoter sequence creates one negative sequence with 35–40% conserved portions from the promoter sequence. This proportion is ideal as a reliable predictor of promoter activity.

Dataset for the second layer

To create the positive and negative sets for the second layer of the model, promoter sequence type information as a host or phage was retrieved. The collection contains several promoters of unknown types. Finally, we collected 139 phage promoter-negative and 478 host promoter-positive samples. We randomly chose 80% of these positive and negative samples as the training dataset and 20% as the test dataset. Table 1 lists the dataset parameters for both layers.

Methods

In this section, we briefly explain the proposed model, feature encoding techniques, and baseline models.

Proposed model

The proposed two-layer model is designated "iProm-phage." The model's first layer predicts the query sequence as a phage

promoter or non-promoter. If the predicted sequence is a phage promoter then the model's second layer classifies it as a phage or host. Figure 2 illustrates the proposed model.

Based on performance measures, we opted for the CNN model and one-hot encoding technique for this two-layer predictor. The selection of the model and encoding technique are briefly explained in the performance measure section.

Convolutional neural network model architecture

The CNN is composed of 2 one-dimensional convolutional layers (Conv1D), which are followed by maximum (max) pooling and dropout layers. The filter and kernel sizes of both Conv1D is 16 and 5, respectively. The max pooling size is four with strides of two in both the max pooling layers. A dropout layer is utilized after each max pooling layer, with a value of 0.5. A flattened layer was utilized, followed by a dense layer with 64 nodes. Subsequently, we used a dropout layer with a value of 0.5. The ReLU activation function was utilized in all the Conv1D and dense layers. Finally, the dense layer is employed as an output layer with a single node and sigmoid activation function that classifies the input sequence as positive or negative based on the probability scores. The mathematical expression for the sigmoid activation function is as follows:

$$S(p) = \frac{1}{1+\exp(-p)}$$

We used L2 regularization and bias regularization in the convolution and dense layers to ensure that the model did not overfit. The values for both regularizations were set to 0.0001. The loss function of the model is binary cross-entropy. Adam was used as the optimizer. The batch size was set to 20 with a total of 85 epochs. iProm-phage was created and trained using the Keras framework. The CNN architecture is illustrated in Figure 3.

Feature encoding techniques

A DNA sequence is comprised of the *A*, *C*, *G*, and *T* nucleotides. To perform computational operations, the sequence must be translated into a numerical representation. Feature encoding schemes play a vital role in creating optimal predictors. The input size should be the same for all sequences. We apply the zero-filled method to make every DNA sequence with an equal length of 99 bp. This technique was previously applied by DPProm (Wang et al., 2022). In this study, we find the best feature encoding technique among the 10 different techniques. The details of each encoding scheme are presented below.

One-hot feature encoding

One-hot encoding techniques are used by many state-of-the-art bioinformatics tools (Umarov and Solovyev, 2017; Liu and Li, 2019; Shujaat et al., 2021; Kim et al., 2022). Each nucleotide in a DNA sequence is represented by a four-dimensional vector, which is a vector of zeros with a single one. Nucleotide *A* is encoded as (1,0,0,0), *C* (0,1,0,0), *G* (0,0,1,0), and *T* (0, 0,0,1). Each DNA sequence can be represented by a (99,4) two-dimensional vector.

Nucleotide chemical property feature encoding

The chemical characteristics of the four DNA nucleic acids differ (Jeong et al., 2014). Nucleotides are classified into three types based on their chemical characteristics: hydrogen-bond strength, base type, and functional groups. Purines with two rings are represented by the letters *A* and *G*, whereas pyrimidines with one ring are represented by the letters *C* and *T*. The hydrogen bonds between *A* and *T* are weak, whereas the hydrogen bonds between *C* and *G* are strong. In terms of functional groups, the amino group includes *A* and *C*, whereas the keto group includes *G* and *T*. Each DNA sequence is represented by a three-dimensional vector (*b*, *c*, *p*) based on chemical properties, where n_i denotes the nucleotide *n* at position *i;* hence, *b*, *c*, and, *p* were computed as follows:

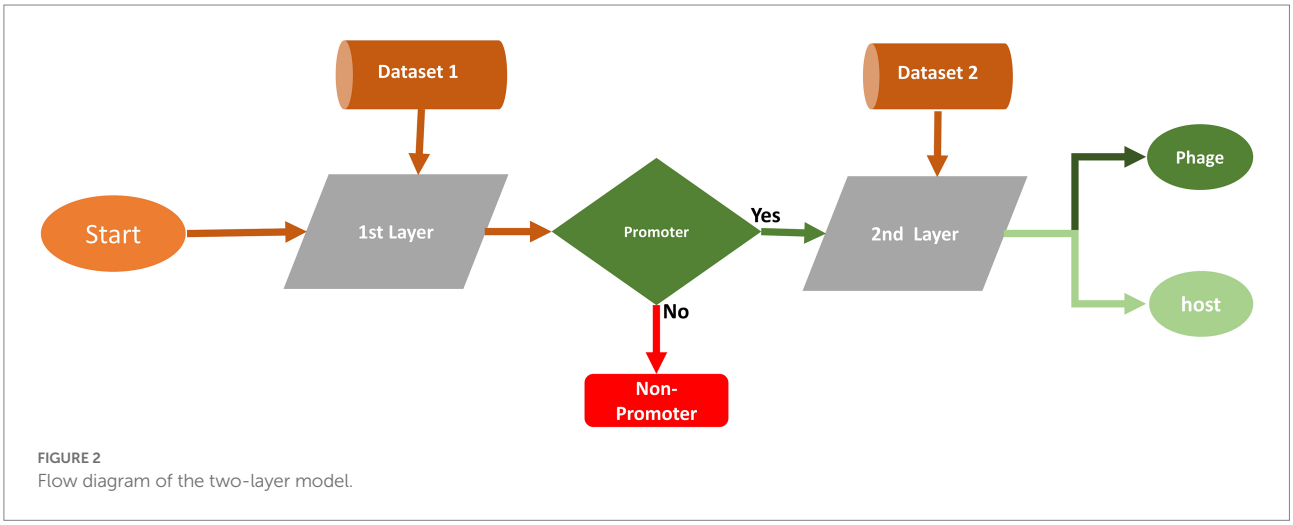

FIGURE 2
Flow diagram of the two-layer model.

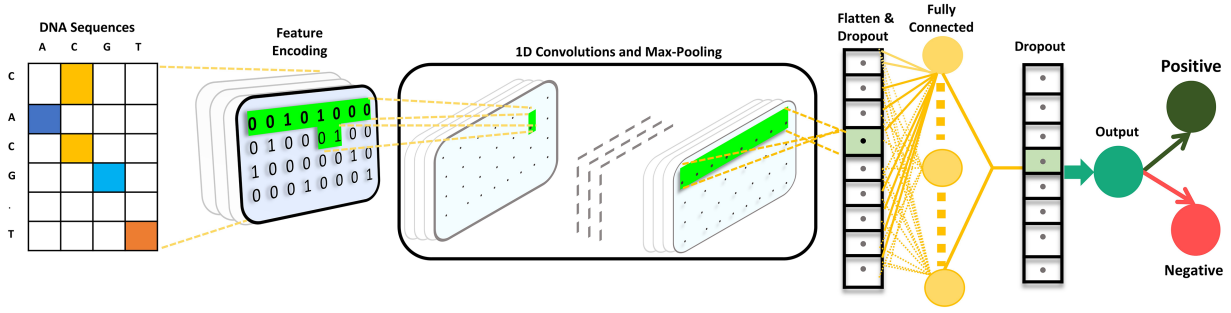

FIGURE 3
iProm-phage CNN architecture.

$$b_i = \begin{cases} 1 \text{ if } n_i \in \{A,C\} \\ 0 \text{ if } n_i \in \{G,T\} \end{cases}, \quad c_i = \begin{cases} 1 \text{ if } n_i \in \{A,G\} \\ 0 \text{ if } n_i \in \{C,T\} \end{cases},$$
$$p_i = \begin{cases} 1 \text{ if } n_i \in \{A,T\} \\ 0 \text{ if } n_i \in \{C,G\} \end{cases}$$

Dinucleotide-based auto-cross covariance feature encoding

DACC is a combination of dinucleotide-based auto-covariance (DAC) and dinucleotide-based cross covariance (DCC) encoding. DAC computes the correlation of the same physicochemical index between two dinucleotides separated by a lag distance along the sequence. DAC is calculated as:

$$\mathrm{DAC}(u,\mathrm{lag}) = \sum_{i=1}^{L-\mathrm{lag}-1} \begin{pmatrix} \left(P_u\left(R_iR_{i+1}\right) - \bar{P}_u\right) \\ \left(P_u\left(R_{i+\mathrm{lag}}R_{i+\mathrm{lag}+1}\right) - \bar{P}_u\right) / \left(L - \mathrm{lag} - 1\right) \end{pmatrix}$$

where u, L represent the physicochemical index and length of the sequence, respectively, and the physicochemical index u for the dinucleotide $\left(R_iR_{i+1}\right)$ at position i is expressed numerically as $P_u\left(R_iR_{i+1}\right)$. $\bar{P}_u$ represents the average value of the physicochemical index u along the whole sequence, and is calculated as:

$$\bar{P}_u = \sum_{j=1}^{L-1} P_u\left(R_jR_{j+1}\right) / \left(L-1\right)$$

The DAC feature vector has a dimension of $N \times \mathrm{LAG}$, where LAG is the maximum lag (lag = 1, 2,…, LAG) and N is the total number of physicochemical indices. DCC computes the correlation of two different physicochemical indices between two dinucleotides along the sequence separated by *lag* nucleic acids. Mathematically, DCC can be represented as

$$\mathrm{DCC}(u_1,u_2,\mathrm{lag}) = \sum_{i=1}^{L-\mathrm{lag}-1} \left(P_{u_1}\left(R_iR_{i+1}\right) - \bar{P}_{u_1}\right) \left(P_{u_2}\left(R_{i+\mathrm{lag}}R_{i+\mathrm{lag}+1}\right) - \bar{P}_{u_2}\right) / \left(L - \mathrm{lag} - 1\right)$$

where u_1, u_2 and L represent the physicochemical indices and length of the nucleotide sequence, respectively, $P_{u_1}\left(R_iR_{i+1}\right)$ is the numerical value of the physicochemical index u_1 for the dinucleotide $\left(R_iR_{i+1}\right)$ at position i, and $\bar{P}_{ua}$ is the average value for the physicochemical index u_a along the whole sequence, calculated as:

$$\bar{P}_{ua} = \sum_{j=1}^{L-1} P_{ua}\left(R_jR_{j+1}\right) / \left(L-1\right)$$

The DCC feature vector has dimensions of $N \times \left(N-1\right) \times \mathrm{LAG}$, where LAG is the maximum lag (lag = 1, 2,.., LAG) and N is the total number of physicochemical indices. Thus, the dimension of the DACC encoding is $N \times N \times \mathrm{LAG}$, where N is the number of physicochemical indices and LAG is the maximum lag (lag = 1, 2, …, LAG).

Pseudo dinucleotide composition

PseDNC encoding incorporates both contiguous local and global sequence order information into a feature vector of the nucleotide sequence. PseDNC is mathematically defined as follows:

$$S = \left[s_1, s_2, \ldots, s_{16}, s_{16+1}, \ldots, s_{16+1}, \ldots, s_{16+\lambda}\right]^T$$

Whereas:

$$s_k = \begin{cases} \dfrac{f_k}{\sum_{i=1}^{16} f_i + w\sum_{j=1}^{\lambda}\theta_j}, \left(1 \le k \le 16\right) \\ \dfrac{w\theta_{k-16}}{\sum_{i=1}^{16} f_i + w\sum_{j=1}^{\lambda}\theta_j}, \left(17 \le \boldsymbol{k} \le 16+\lambda\right) \end{cases}$$

where f_k $(k = 1, 2, \ldots, 16)$ is the normalized frequency of dinucleotide occurrence in the nucleotide sequence, λ

represents the highest counted rank (or tie) of the correlation along the nucleotide sequence, w is the weight factor ranging from 0 to 1, and θ_j $(j=1,2,\ldots,\ \lambda)$ is the jth correlation factor and is defined as

$$\begin{cases} \theta_1 = \dfrac{1}{L-2}\sum_{i=1}^{L-2}\theta\left(R_iR_{i+1},R_{i+1}R_{i+2}\right) \\ \theta_1 = \dfrac{1}{L-2}\sum_{i=1}^{L-3}\theta\left(R_iR_{i+1},R_{i+2}R_{i+3}\right) \\ \theta_1 = \dfrac{1}{L-2}\sum_{i=1}^{L-4}\theta\left(R_iR_{i+1},R_{i+3}R_{i+3}\right)(\lambda < L) \\ \cdots \\ \theta_\lambda = \dfrac{1}{L-1-\lambda}\sum_{i=1}^{L-1-\lambda}\theta\left(R_iR_{i+1},R_{i+\lambda}R_{i+\lambda+1}\right) \end{cases}$$

The correlation function is given as follows:

$$\theta\left(R_iR_{i+1},R_{j+1}R_{j+1}\right) = \frac{1}{\mu}\sum_{u=1}^{\mu}\left[P_\mu\left(R_iR_{i+1}\right) - P_\mu\left(R_jR_{j+1}\right]^2\right.$$

where physicochemical indices are represented by μ, $P_\mu\left(R_iR_{i+1}\right)$ measures are the numerical values of the u-th ($u=1$, 2, …, μ) physicochemical index of the dinucleotide R_iR_{i+1} at position i and $P_\mu\left(R_jR_{j+1}\right)$ represents the corresponding value of the dinucleotide R_jR_{j+1} at position j .Pseudo k-tupler composition (PseKNC).

PseKNC encoding uses a k-tuple nucleotide composition defined as

$$D = \left[d_1,d_2,\ldots,d_{4^k},d_{4^k+1},\ldots,d_{4^k+\lambda}\right]^T$$

Whereas:

$$\begin{cases} \dfrac{f_u}{\sum_{i=1}^{4^k}f_i + w\sum_{j=1}^{\lambda}\theta_j}, (1 \le u \le 4) \\ \dfrac{w\theta_{u-4^k}}{\sum_{i=1}^{4^k}f_i + w\sum_{j=1}^{\lambda}\theta_j}, \left(4^k \le u \le 4^k+\lambda\right) \end{cases}$$

where λ is the total number of ranks of correlations along a nucleotide sequence, $f_u\left(u=1,2,\ldots,4^k\right)$ is the frequency of oligonucleotides normalized to $\sum_{i=1}^{4^k}f_i = 1$, w is the factor, and θ_j is defined as follows:

$$\theta_j = \frac{1}{L-j-1}\sum_{i=1}^{L-j-1}\Theta\left(R_iR_{i+1},R_{i+j}R_{i+j+1}\right),$$
$$\left(j=1,,,;2,,,;\ldots,,,;\lambda,,,;\lambda < L\right)$$

The correlation function is defined as:

$$\tilde{}\ \left(R_iR_{i+1},R_{i+j}R_{i+j+1}\right) = \frac{1}{\mu}\sum_{v=1}^{\mu}\left[P_v\left(R_iR_{i+1}\right) - P_v\left(R_{i+j}R_{i+j+1}\right)\right]^2$$

where μ represents the physicochemical index. $P_v\left(R_iR_{i+1}\right)$ is a numerical value v-th ($v=1$, 2, …, μ). The physicochemical index of dinucleotide $\left(R_iR_{i+1}\right)$ at position i and $P_v\left(R_{i+j}R_{i+j+1}\right)$ represents the corresponding value of dinucleotide $\left(R_{i+j}R_{i+j+1}\right)$ at position $i+j$.

Electron-ion interaction pseudopotentials of trinucleotide

The values of nucleotides *A*, *G*, *C*, and *T* electron-ion interaction pseudopotentials (EIIP) were determined as previously described using Nair (Lavigne et al., 2004; *A*: 0.1260, *C*: 0.1340, *G*: 0.0806, *T*: 0.1335). Nucleotides in the DNA sequence are directly represented by EIIP using the EIIP value. EIIPA, EIIPT, EIIPG, and EIIPC represent the EIIP values of nucleotides *A*, *T*, *G*, and *C*, respectively, in PseEIIP encoding. A feature vector is created using the mean EIIP value of the trinucleotides in each sample, as follows:

$$V = \left[EIIP_{AAA}\cdot f_{AAA},EIIP_{AAC}\cdot f_{AAC},\ldots,EIIP_{TTT}\cdot f_{TTT}\right]$$

Parallel correlation pseudo dinucleotide composition

Similar to PseDNC, PCPseDNC encoding differs in that it uses 38 default physiochemical indices for DNA instead of the six indices used in PseDNC encoding. Supplementary Table S2 in Supplementary file presents a list of 38 physicochemical indices.

Parallel correlation pseudo trinucleotide composition

PCPseTNC encoding is described as:

$$S = \left[s_1,s_2,\ldots,s_{64},s_{64+1},\ldots,s_{64+\lambda}\right]^T$$

Whereas:

$$s_k = \begin{cases} \dfrac{f_k}{\sum_{i=1}^{64}f_i + w\sum_{j=1}^{\lambda}\theta_j}, (1 \le k \le 64) \\ \dfrac{w\theta_{k-64}}{\sum_{i=1}^{64}f_i + w\sum_{j=1}^{\lambda}\theta_j}, (65 \le k \le 64+\lambda) \end{cases}$$

where f_k ($k = 1, 2, \ldots, 64$) is the normalized frequency of dinucleotide occurrence in the nucleotide sequence, λ represents the highest counted rank (or tie) of the correlation along the nucleotide sequence, w is the weight factor ranging from 0 to 1, and θ_j ($j = 1,2,\ldots, \lambda$) is the jth correlation factor and is defined as:

$$\begin{cases} \theta_1 = \frac{1}{L-3}\sum_{i=1}^{L-3}\Theta\left(R_iR_{i+1}R_{i+2}, R_{i+1}R_{i+2}R_{i+3}\right) \\ \theta_2 = \frac{1}{L-4}\sum_{i=1}^{L-4}\Theta\left(R_iR_{i+1}R_{i+2}, R_{i+2}R_{i+3}R_{i+4}\right) \\ \theta_3 = \frac{1}{L-5}\sum_{i=1}^{L-5}\Theta\left(R_iR_{i+1}R_{i+2}, R_{i+3}R_{i+4}R_{i+5}\right)\left(\lambda < L\right) \\ \theta_\lambda = \frac{1}{L-2-\lambda}\sum_{i=1}^{L-2-\lambda}\Theta\left(R_iR_{i+1}R_{i+2}, R_{i+\lambda}R_{i+\lambda+1}R_{i+\lambda+2}\right) \end{cases}$$

The correlation function is defined as:

$$\Theta\left(R_iR_{i+1}R_{i+2}, R_{j+1}R_{j+1}R_{j+2}\right) = \frac{1}{\mu}\sum_{u=1}^{\mu}\left[P_u\left(R_iR_{i+1}R_{i+2}\right) - P_u\left(R_jR_{j+1}R_{j+2}\right)\right]^2$$

where physicochemical indices are represented by μ, $P_\mu\left(R_iR_{i+1}R_{i+2}\right)$ measures are the numerical values of the u-th ($u = 1, 2, \ldots, \mu$) physicochemical index of the dinucleotide $R_iR_{i+1}R_{i+2}$ at position i and $P_\mu\left(R_jR_{j+1}R_{j+2}\right)$ represents the corresponding value of the dinucleotide $R_jR_{j+1}R_{j+2}$ at position j.

Moran correlation

The distribution of amino acid characteristics along the sequence is used to create autocorrelation descriptors (Horne, 1988; Feng and Zhang, 2000; Sokal and Thomson, 2006). The amino acid properties used here are different types of amino acid indices retrieved from the AAindex Database (Kawashima et al., 2008) available at http://www.genome.jp/dbget/aaindex.html.

kmer

DNA sequences are represented as the occurrence frequencies of k adjacent nucleic acids in the kmer descriptor, which has been effectively used for human gene regulatory sequence prediction. The kmer descriptor ($k = 3$) is calculated as follows:

$$f(t) = \frac{N(t)}{N}, t \varepsilon \{AAA, AAC, AAG, \ldots, TTT\}$$

where $N(t)$ represents the number of kmer types (t) and N is the length of the sequence.

Baseline models

Selection of the optimal model is a vital step in developing a novel predictor. We have utilized different ML and CNN models and, based on performance measures, selected the best model. ML models include the Adaboost (AdB) classifier, multinomial naive Bayes, extreme gradient boosting (XGboost), gradient boosting (Gboost), logistic regression (LR), K-nearest neighbor, decision tree classifier, support vector machine (SVM), multilayer perceptron classifier, and SVM bagging. A CNN is composed of two convolution layers. We used hyperparameter tuning to determine the best convolution, pooling, dropout, and dense layer parameters.

Performance measures

In this section, we explain the evolution metrics, selection of the best model and feature encoding scheme, model performance, and model comparison.

Evaluation metrics

In the performance assessment matrix, we used the accuracy (Acc), sensitivity (Sn), specificity (Sp), and MCC. These parameters have been used in several cutting-edge studies. The numerical representation of an evaluation matrix is expressed using the following equations:

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{Sn} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Sp} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

$$\text{MCC} = \frac{\text{TP} * \text{TN} - \text{FP} * \text{FN}}{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}$$

The terms TP, TN, FP, and FN in the aforementioned equations represent the appropriate numbers of true positives, true negatives, false positives, and false negatives, respectively.

Selection of best model and feature encoding

To generate an optimum model, we compared all the encoding strategies stated above to the baseline approaches. Supplementary Tables S3, S4 in Supplementary file, and

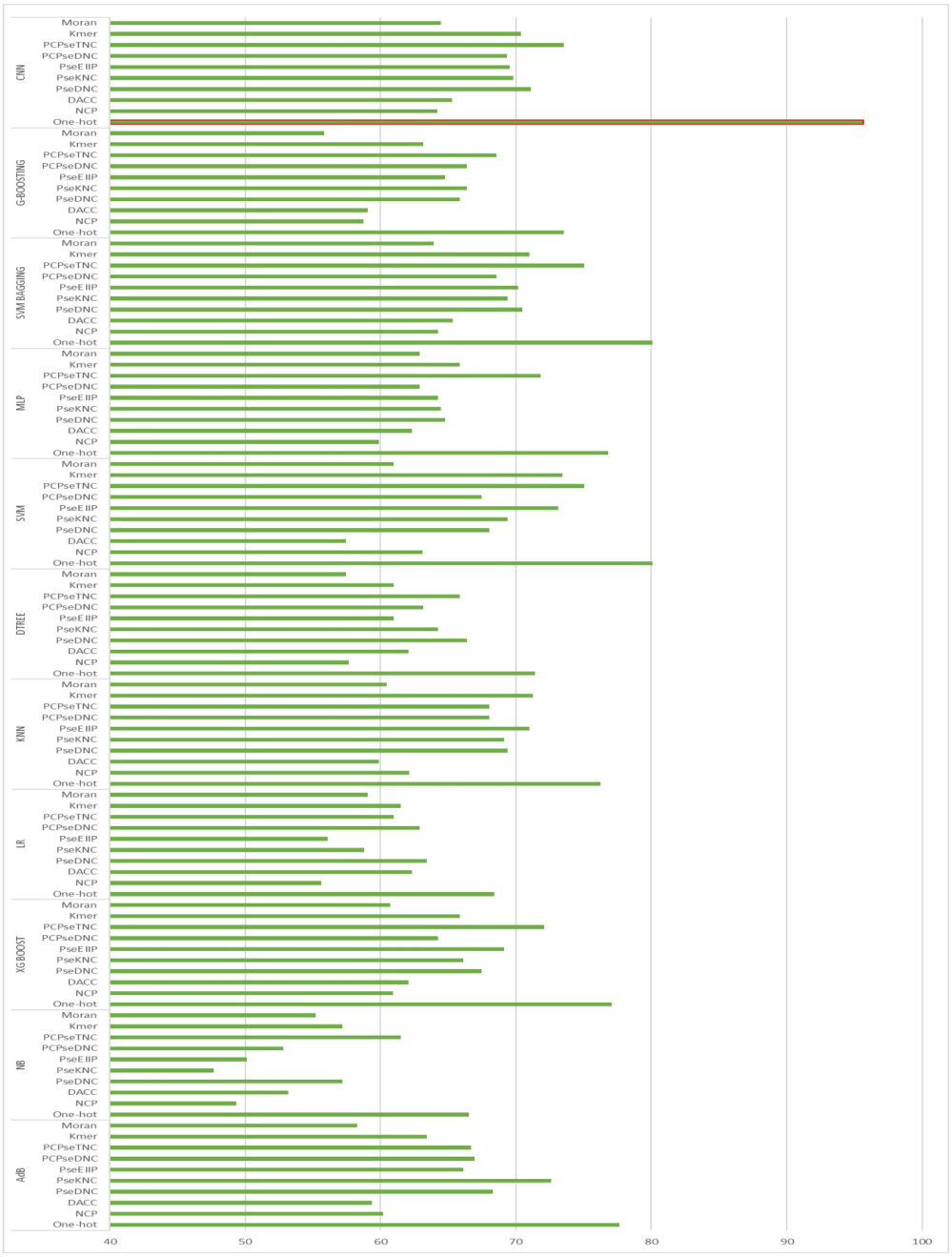

FIGURE 4
Accuracy of First layer baseline models.

Figures 4, 5 illustrate the performance of each method on various encoding schemes for the first and second layers. For the first layer of the model CNN and one-hot encoding outperformed after that AdB performed better on PseKNC feature encoding and for the second layer almost every feature encoding scheme performed good on ML and CNN algorithms, but one-hot and CNN outperformed in the second layer as well. Therefore, based on performance evaluation, we chose the CNN and one-hot encoding technique for both layers and the proposed tool "iProm-phage."

Model performance

The prediction performance of iProm-phage was evaluated using 5-fold cross validation. We employed the same parameters

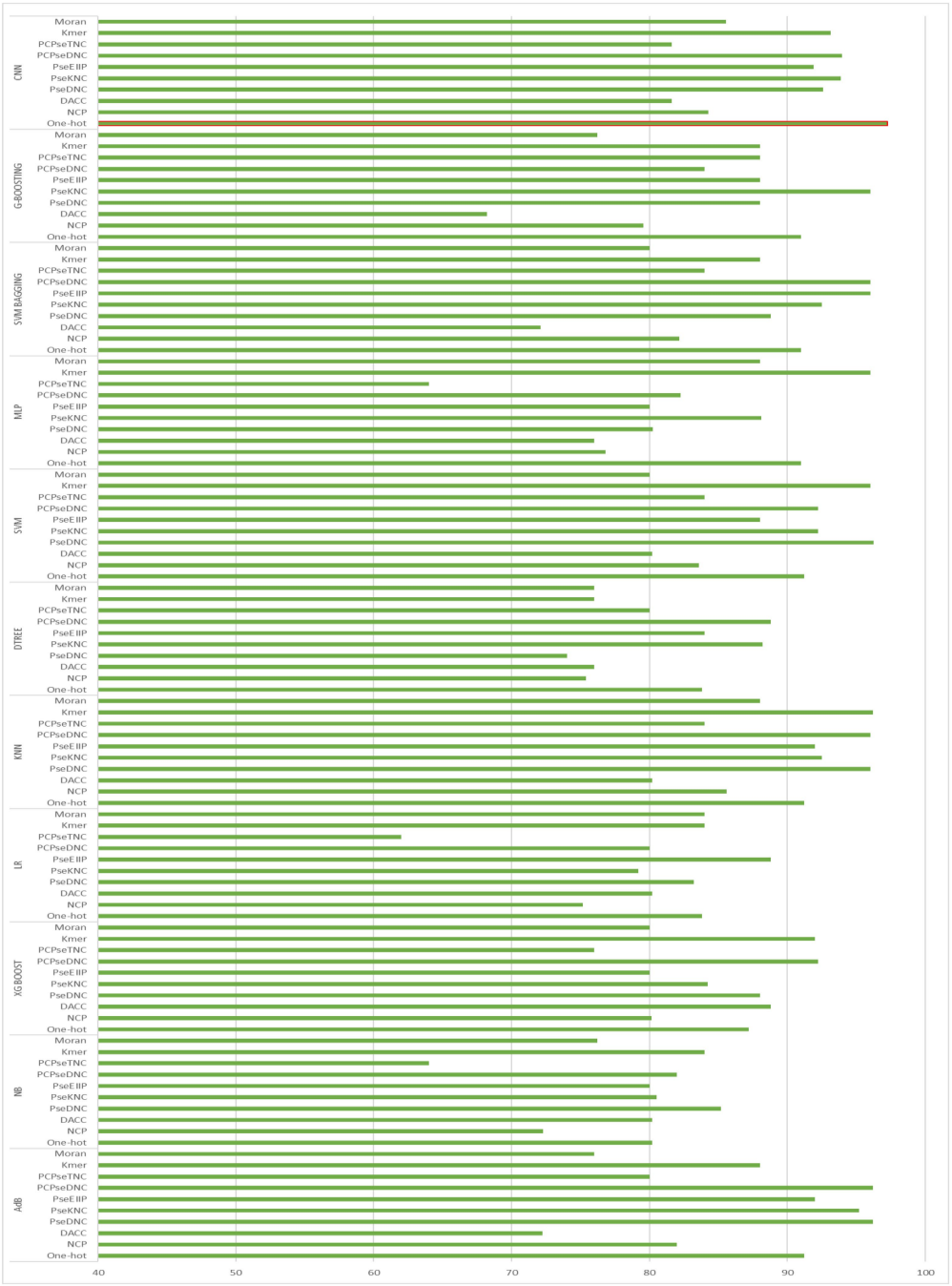

FIGURE 5
Accuracy of Second layer baseline models.

used in choosing the best model and also considered ROC curve data. The first layer of iProm-phage achieved an Acc of 95.68 93.47%, Sn of 96.12%, Sp of 92.63%, MCC of 0.872, and AUROC of 0.99 during cross validation. These findings suggest that our predictor is capable of properly recognizing whether a query sequence is a promoter. The second layer of iProm-Zea achieved values of 97.25, 94.32, 98.5%, 0.8619, and 0.97, respectively. In the test dataset model, the first layer achieved an accuracy of 94.2%, Sn 90%, Sp 90%, and MCC 0.88. The second layer obtained accuracies of 95.2%, 94.37%, 97.14%, and 0.88% for the test dataset. Figures 6, 7 depict the ROC curves for both layers of the iProm-phage model.

Comparison with existing models

We compared iProm-phage with state-of-the-art promoter identification tools PhagePromoter and DPProm for the identification of query sequences as promoters or promoters.

TABLE 2 First layer performance comparison.

Methods	Acc%	Precision%	Recall%
PhagePromoter	92	89	87
DPProm	85.5	88.9	83
iProm-phage	95.68	94.2	93.5

TABLE 3 Second layer performance comparison.

Methods	Acc%	Precision%	Recall%
DPProm	93.0	95.2	96.4
iProm-phage	95.2	96.5	97.2

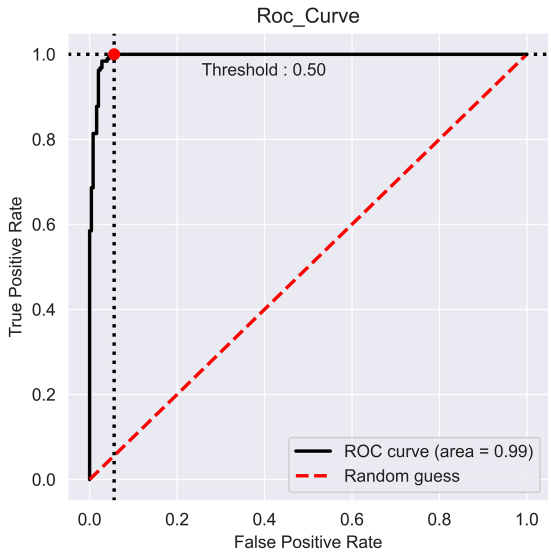

FIGURE 6
First layer ROC curve.

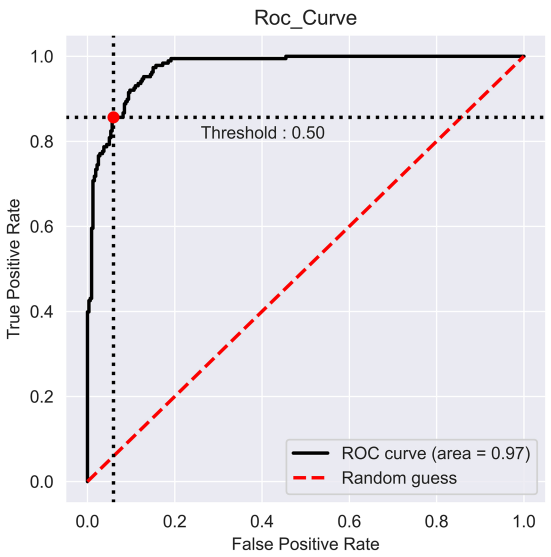

FIGURE 7
Second layer ROC curve.

We measured the precision and recall for both layers to compare them with state-of-the-art methods. The following equations express precision and recall:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Precison} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

A performance comparison of the methods used for promoter identification is presented in Table 2. The superior performance of the proposed iProm-phage tool can be observed in all four performance metrics for this particular task.

We demonstrate the performance comparison between DPProm in Table 3 for promoter classification as a phage or host. The iProm-phage tool was superior to DPProm in performance for all classification tasks. The precision and recall of iProm-phage for promoter identification and classification were higher than those of DPProm, and the values were more consistent. As a result, iProm-phage showed a considerably higher score than the state-of-the-art methods in all cases.

Webserver

A web server hosting the high performance iProm-phage tool is freely available at the following link[1] to enable easy access to the proposed tool for the scientific community. This approach has been adopted by several scholars (Chantsalnyam et al., 2020; Ali SD et al., 2022). iProm-phage is an easy-to-use tool that can be utilized by researchers and specialists in bioinformatics. It consists of two stages first is input and second is output. To input it uses two input methods: direct sequence input and uploading a file containing sequences for prediction. Each sequence should be 99 bp long and contain the letters *A*, *C*, *G*, and *T*. Figures 8, 9 depict web server snippets; Figure 8 is an example of adding sequences for prediction and Figure 9 provides the predictor's output. We also provide an example to better understand how to use the webserver.

1 http://nsclbio.jbnu.ac.kr/tools/iProm-phage/

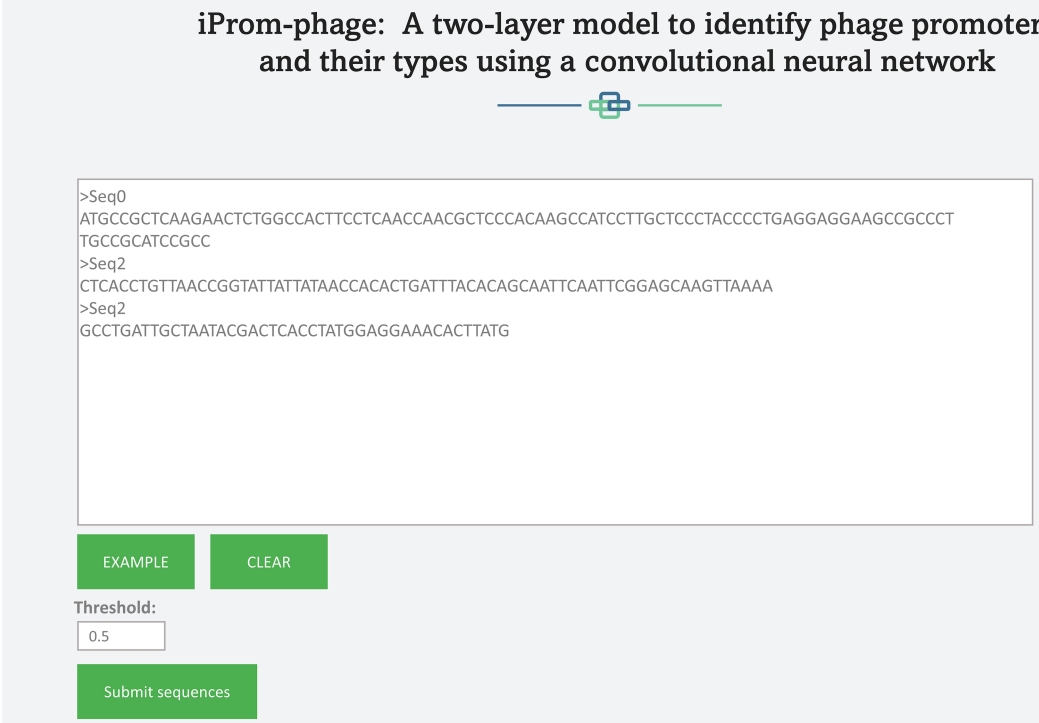

FIGURE 8
Webserver adding query sequence.

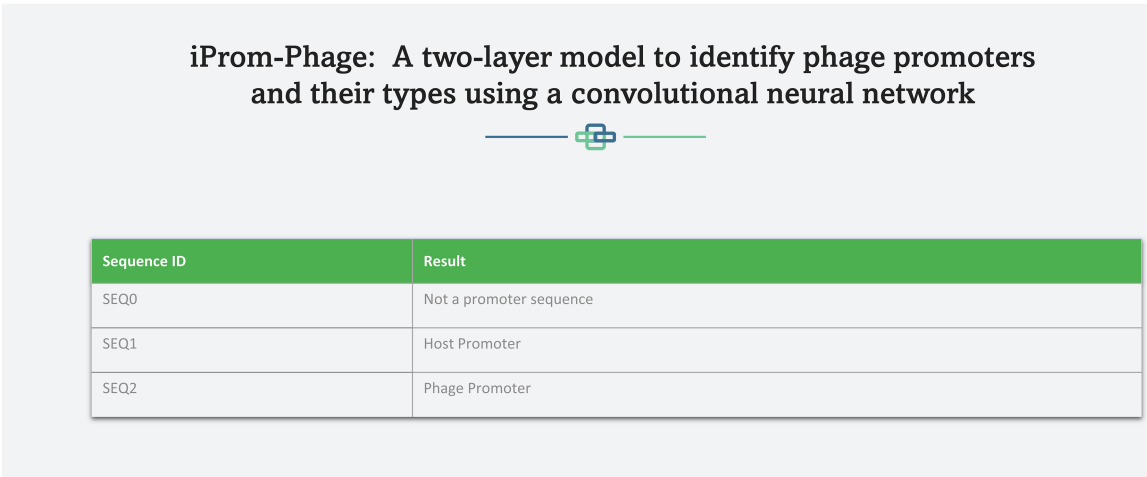

FIGURE 9
Predictor output.

Conclusion

This work presents iProm-phage, a two-layer technique for identifying phage promoters and classifying them as phages or hosts. We developed a new method for generating negative datasets to create a robust model that performs well on tough datasets. Based on cutting-edge performance tests, we also found the best model among several ML and CNN algorithms, as well as the best feature encoding method among the 10 distinct methods. The architecture of the proposed model was evaluated using publicly available datasets. Compared to earlier techniques, the program had superior overall results. Finally, we created a web server that is available online and will be extremely useful to other experimental scientists.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary material, further inquiries can be directed to the corresponding authors.

Author contributions

MS: conceptualization, methodology, software, writing–original draft, and writing–review and editing. JJ: methodology and writing–review and editing. HT: supervision and writing–review and editing. KC: conceptualization, validation, supervision, writing–review and editing, and funding acquisition. All authors contributed to the article and approved the submitted version.

Funding

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT; nos. 2020R1A2C2005612 and 2022R1G1A1004613). This work was supported by "Human Resources Program in Energy Technology" of the Korea Institute of Energy Technology Evaluation and Planning (KETEP), granted financial resource from the Ministry of Trade, Industry & Energy, Republic of Korea (no. 20204010600470).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fmicb.2022.1061122/full#supplementary-material

References

Ali, S. D., Alam, W., Tayara, H., and Chong, K. (2020). Identification of functional pi RNAs using a convolutional neural network. *IEEE/ACM Trans. Comput. Biol. Bioinforma.* 14:1. doi: 10.1109/tcbb.2020.3034313

Ali, S. D., Alam, W., Tayara, H., and Chong, K. T. (2022). Identification of functional piRNAs using a convolutional neural network. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 19, 1661–1669. doi: 10.1109/TCBB.2020.3034313

Chantsalnyam, T., Lim, D. Y., Tayara, H., and Chong, K. T. (2020). ncRDeep: non-coding RNA classification with convolutional neural network. *Comput. Biol. Chem.* 88:107364. doi: 10.1016/j.compbiolchem.2020.107364

Feng, Z. P., and Zhang, C. T. (2000). Prediction of membrane protein types based on the hydrophobic index of amino acids. *J. Protein Chem.* 19, 269–275. doi: 10.1023/A:1007091128394

Guzina, J., and Djordjevic, M. (2015). Bioinformatics as a first-line approach for understanding bacteriophage transcription. *Bacteriophage* 5:e1062588. doi: 10.1080/21597081.2015.1062588

Horne, D. S. (1988). Prediction of protein helix content from an autocorrelation analysis of sequence hydrophobicities. *Biopolymers* 27, 451–477. doi: 10.1002/bip.360270308

Jeong, B.-S., Golam Bari, A. T. M., Rokeya Reaz, M., Jeon, S., Lim, C.-G., and Choi, H.-J. (2014). Codon-based encoding for DNA sequence analysis. *Methods* 67, 373–379. doi: 10.1016/j.ymeth.2014.01.016

Kawashima, S., Pokarowski, P., Pokarowska, M., Kolinski, A., Katayama, T., and Kanehisa, M. (2008). AAindex: amino acid index database, progress report 2008. *Nucleic Acids Res.* 36, D202–D205. doi: 10.1093/nar/gkm998

Kim, J., Shujaat, M., and Tayara, H. (2022). Iprom-zea: a twolayer model to identify plant promoters and their types using convolutional neural network. *Genomics* 114:110384. doi: 10.1016/j.ygeno.2022.110384

Klucar, L., Stano, M., and Hajduk, M. (2010). Phi SITE: database of gene regulation in bacteriophages. *Nucleic Acids Res.* 38, D366–D370. doi: 10.1093/nar/gkp911

Lavigne, R., Sun, W. D., and Volckaert, G. (2004). PHIRE, a deterministic approach to reveal regulatory elements in bacteriophage genomes. *Bioinformatics* 20, 629–635. doi: 10.1093/bioinformatics/btg456

Liu, B., and Li, K. (2019). Ipromoter-2l2. 0: identifying promoters and their types by combining smoothing cutting window algorithm and sequence-based features. *Mol. Ther. Nucleic Acids* 18, 80–87. doi: 10.1016/j.omtn.2019.08.008

Mishra, A., Dhanda, S., Siwach, P., Aggarwal, S., and Jayaram, B. (2020). A novel method seprom for prokaryotic promoter prediction based on dna structure and energetics. *Bioinformatics* 36, 2375–2384. doi: 10.1093/bioinformatics/btz941

Rahman, M. S., Aktar, U., Jani, M. R., and Shatabda, S. (2019). iPro70-FMWin: identifying sigma 70 promoters using multiple windowing and minimal features. *Mol. Gen. Genomics.* 294, 69–84. doi: 10.1007/s00438-018-1487-5

Rehman, M. U., Hong, K. J., Tayara, H., and Chong, K. T. (2021). To Chong, m6A-neural tool: convolution neural tool for RNA N6-methyladenosine site identification in different species. *IEEE Access* 9, 17779–17786. doi: 10.1109/ACCESS.2021.3054361

Salmond, G. P., and Fineran, P. C. (2015). A century of the phage: past, present and future. *Nat. Rev. Microbiol.* 13, 777–786. doi: 10.1038/nrmicro3564

Sampaio, M., Rocha, M., Oliveira, H., and Dias, O. (2019). Predicting promoters in phage genomes using phage promoter. *Bioinformatics* 35, 5301–5302. doi: 10.1093/bioinformatics/btz580

Shujaat, M., Lee, S. B., Tayara, H., and Chong, K. T. (2021). Crprom: a convolutional neural network-based model for the prediction of rice promoters. *IEEE Access* 9, 81485–81491. doi: 10.1109/ACCESS.2021.3086102

Shujaat, M., Wahab, A., Tayara, H., and Chong, K. T. (2020). Chong, pc promoter-CNN: a CNN-based prediction and classification of promoters. *Genes (Basel)* 11:1529. doi: 10.3390/genes11121529

Sierro, N., Makita, Y., de Hoon, M., and Nakai, K. (2008). Dbtbs: a database of transcriptional regulation in bacillus subtilis containing upstream intergenic conservation information. *Nucleic Acids Res.* 36, D93–D96. doi: 10.1093/nar/gkm910

Silva, S., and Echeverrigaray, S. (2012). Bacterial promoter features description and their application on *E. coli* in silico prediction and recognition approaches. *Bioinformatics. InTech* 1, 241–260. doi: 10.5772/48149

Sokal, R. R., and Thomson, B. A. (2006). Population structure inferred by local spatial autocorrelation: an example from an Amerindian tribal population. *Am. J. Phys. Anthropol.* 129, 121–131. doi: 10.1002/ajpa.20250

Umarov, R. K., and Solovyev, V. V. (2017). Recognition of prokaryotic and eukaryotic promoters using convolutional deep learning neural networks. *PLoS One* 12:e0171410. doi: 10.1371/journal.pone.0171410

Wang, Y., Wang, H., Wei, L., Li, S., Liu, L., and Wang, X. (2020). Synthetic promoter design in escherichia coli based on a deep generative network. *Nucleic Acids Res.* 48, 6403–6412. doi: 10.1093/nar/gkaa325

Wang, C., Zhang, J., Cheng, L., Wu, J., Xiao, M., Xia, J., et al. (2022). DPProm: a two-layer predictor for identifying promoters and their types on phage genome using deep learning. *IEEE J. Biomed. Health Inform.* 26, 5258–5266. doi: 10.1109/JBHI.2022.3193224