RESEARCH PAPER

C-LSTM SMOTE-RFE & XG BOOST Crispr: Deep Learning Models for Predicting CRISPR/Cas12 Guide RNA Activity

R. Sulakshana[1], Dr. R. Lakshmi[2]

[1]Research Scholar, Department of Computer Science, Pondicherry University (Karaikal Campus), India
Email: sulopragash@hotmail.com

[2]Associate Professor, Department of Computer Science, Pondicherry University (Karaikal Campus), India
Email: prof.rlakshmi@gmail.com

ABSTRACT

The most significant device in the enhancement of human genes is clustered regularly interspaced palindromic repeat (CRISPR) period, which can be directed to any gene target the usage of gRNA and Cas enzyme. Due to the shortcoming of low gRNA activity of the CRISPR structures, it is very much advanced where its activity of gRNA can be predicted. The gRNA activity can be calculated using the frequency of insertion or deletion (indel). In this current study, CNN get optimized through the method of determining by means of different conv layers intensity and clear out kernel length. The study also identifies conventional MLR, CNN and C-SVR, and CNN with LSTM and SMOTE-RFE & XG to enhance the model performance. CNN-LSTM SMOTE-RFE and XG have been applied to predict gRNA activity, and this variant became tested the application of Accuracy, Sensitivity, Specificity, F1-score, Spearman correlation, Root Mean Squared error and Mean Squared Error and it performed excellently. The hybrid version is also better than the ultra-modern version in forecasting gRNA pastime through way of means of up to 45%. Lastly, to prove the hybrid model, the study predicts a frequency of indel in the gRNA sequences used in COVID-19 detection; this could be a useful feature in finding the best gRNA to be used in identification of COVID-19 by CRISPR/Cas12 virus.

Keywords: CRISPR, Indel frequency, COVID-19, Convolutional Neural Network, Xtreme Gradient Boosting, Long Short Term Memory Network, SMOTE-RFE, Guide RNA activity, Deep Learning.

How to cite this article: Sulakshana R, Lakshmi R. C-LSTM SMOTE-RFE & XG BOOST Crispr: Deep Learning Models for Predicting CRISPR/Cas12 Guide RNA Activity. Int J Drug Deliv Technol. 2026;16(6s): 1025-1035; DOI: 10.25258/ijddt.16.6s.134

Source of support: None

Conflict of interest: None

1. Introduction

The genetic material was efficiently edited with the help of use of CRISPR/Cas12 system. In human, animal and other type of cell, DNA and RNA was modified by using CRISPR/Cas12 system. The development of new treatment, diagnostics and the implementation of basic biological research depend on accuracy and flexibility of CRISPR-Cas enzymes [1, 2]. The Cas9 or Cas12a enzymes are used to edit DNA. On the other hand, the enzyme Cas9 was different from Cas12a enzymes. The higher target specificity with greater accuracy was achieved by Cas12a enzymes during fewer unintended cuts in the genome considered as major and first advantage in the implementation of therapeutic gene editing. The alteration of several genes at the same time was achieved by self-processing capability of Cas13 has the capability to process own multiple guide RNAs to be generated from a single transcript based on intrinsic crRNA self-processing capability considered as second major advantage in compared Cas12a with Cas9 in CRISPR-based gene editing [5, 6]. The edition in the number of places was reduced due to occurrence of bonding with DNA regions to G-rich PAM sequences in Cas9 considered as third major advantage. In compared with Cas9, wider range of DNA sites was targeted by Cas12a. Moreover, the process of gene editing was efficiency carried out by using Cas12a due to Trans-cleave ss DNA [7, 8].

The process of initiating gene editing procedure before experiment evaluation and guiding of RNA consumes large amount of time. The successful genome editing was impacted by silico design of guide RNA leads to cause for critical issues. The implementation of RNA design was guided by numerous number of software platform and online tool includes comparison of various type of tools and comprehensive review of the summary. On the other hand, the accurate prediction of specificity of guide RNAs still faces numerous number of challenges despite significant effort. In compared with Cpf1, Cas9 had more number of tools and methodologies developed for implementation. Regarding Cpf1, the need for developing computational methods was considered as mandatory.

Many type of genome engineering task such as targeting RNA molecules, modifying epigenetic marks, making precise DNA base changes, enhancing gene activity, turning genes on or off and modifying epigenetic marks was performed efficiently with the help of CRISPR technology. The rules of the life was rewritten by versatile and powerful CRISPR considered a the era of genome engineering [7]. The diagnosis of infectious disease such as COVID-19 and SARS-CoV-2 virus was diagnosed accurately with the help of CRISPR technologies [8, 9].

The rapid spread of COVID-19 infected 21.2 million confirmed cases and 761,779 fatalities as of August 16, 2020 was predicated with the help of CRISPR based technologies [10]. To design CRISPR/Cas12 (Cpf1) guide RNAs (gRNAs) for the detection of infectious disease, machine learning based approaches was used includes various thermodynamic and sequence features. Kim et al. [10] highlighted that Cas12 gRNA activity is determined actively by GC content, free energy, melting temperature along with position-specific dinucleotides and position-specific nucleotides. On the other hand, other studies focused mainly on the implementation of CRISPR/Cas12 gRNA efficiency [12]. Moreover, substantial domain expertise was needed for extracting essential features in the implementation of traditional machine learning approaches consumes a lot of time. The predication of CRISPR/Cas12 gRNA activity was efficiently handled by deep learning associated with the branch of Artificial Intelligence. The hierarchical feature representations was captured by multiple processing layers uses various types of deep learning models. The process of image classification and recognition, using architectures such as CNN-SVR was processed with the implementation of deep learning model [12, 13, 14]. The nonlinear approaches with the inclusion of multilayer perceptron neural networks (MLPNN) and functional-link neural networks (FLN) were compared with traditional linear model such as multiple regression model and partial least square (PLS) [15]. Across worldwide, existence of virus in clinical sample, progression of disease predication and modelling transmission dynamics was effectively captured by deep learning technique during the occurrence of SARS-CoV 2 pandemic [16, 17]. The predication of prediction of CRISPR gRNA on-target activity based on deep learning model was limited in compared with the implementation of CRISPR/Cas12 systems. The CRISPR/Cas12 gRNA activity was considered as critically important included with the integration of novel and robust computational model due to accurate, effective and safe predication of gene work. In related with CRISPR/Cas12 systems**,** only limited number of deep learning model was developed causes a major challenges in during implementation. The development of new computational method was considered as mandatory to anticipate CRISPR/Cas12 gRNA activity [19, 20, 21].

The main contribution of this study:

- The activity of RNA in the CRISPRCas12 system was predicated with the help of hybrid model helps to save time and money before laboratory implementation. The reliable estimation was assured by assigning an activity score based on indel frequency helps to guide the performance of RNA in a specific genome-editing experiment.
- The integration of gRNA sequence information with epigenetic data helps in the enhancement of predication performance based on the adoption of proposed hybrid deep learning framework.
- The using of larger data set helps to improve the performance of a model but lacks with limited size for the available data.
- In compared with other model, hybrid CNN–LSTM–SMOTE–RFE–XGBoost model shows higher performance improved up to 70% and 80% respectively over the model [11, 14, 22].
- The integration of CoV-2 gene using the CRISPR Cas12, the CNN LSTM SMOTE-RFE & XG Boost hybrid model's helps to identify foreseeing the actions of gRNAs. The implementation of CRISPR/Cas12 gRNA activity for COVID-19 detection was not published in the studies.

2. Sources and Procedures

2.1. Dataset and data pre-processing

Four Cell Line Independent Datasets was adopted as a dataset for this dissertation [23]. By using four well known human cell line dataset such as HCT116, HEK293T, HeLa, and HL60 was used in the study helps to guide RNA (gRNA) data collection. The experimentally validated measurements of gRNA effectiveness contained in the dataset were included in the study. The DeepCRISPR framework was integrated and employed with dataset proposed by Chuai et al. [18]. 23-nucleotide guide RNA sequence was contained in each dataset sample integrated with four features of epigenetic features includes binary and numerical labels representing cleavage efficiency. The ENCODE database with obtained by epigenetic features [24]. Moreover, DNase I hypersensitive sites sequencing (DNase-Seq) assay contain chromatin-opening information with the inclusion of CTCF binding information from the chromatin

immunoprecipitation (ChIP-Seq) and assay, H3K4me3 position information from the chromatin immunoprecipitation (ChIP-Seq) assay.

2.2. Method

The activity of RNA Indel frequency was forecasted by using hybrid CNN-LSTM & XG Boost model used in the research. To train the model, nearly 10% of testing set and 90% of training set was used. The optimization of convolutional layer depth and filter kernel length helps to predictive and guide RNA. The contrast model related to CNN-LSTM & XG model's Spearman Correlation performance metrics was chosen for the study. To evaluate CNN, CNN-LSTM & XG, C-SVR, and MLR Spearman Correlation various type of model was employed helps to deliver the best performance. The implementation of CNN-LSTM and XG helps to enhance the performance. The following figure (Fig.1) helps to explore the methodology of the study.

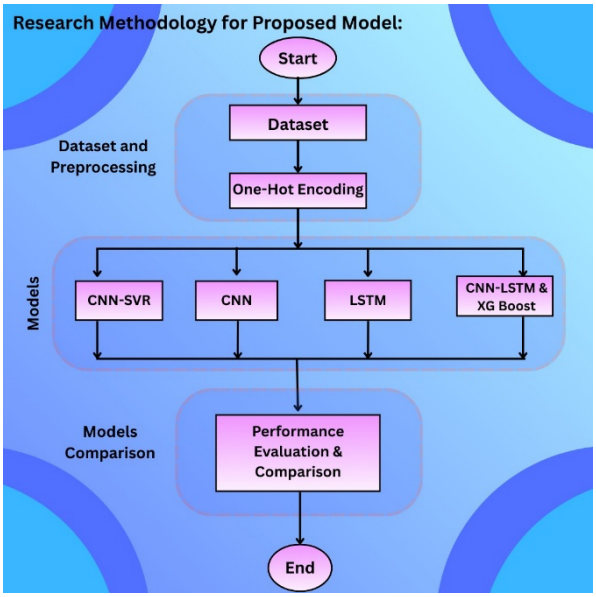

Fig.1. Techniques for estimating gRNA activity.

2.3. Deep learning-based development model

2.3.1. CNN model

The activity of guide RNA (gRNA) in CRISPR experiments was predicated with the help of CNN based deep learning model. Two stages of convolutional layers was used in CNN model. The matrix with dimension of 23 X 4 (gRNA sequence length and no of nucleotides) was considered as input for the model. The length of the gRNA sequence was denoted using the number 24. The four nucleotide types (A, C, G, T) was denoted using the number 4. To understand the important pattern of from the gRNA sequence implementation of CNN uses two stages of 1D convolution layers. The first convolution layer uses kernel size of 7, 128 filters integrated with 1D convolution layer helps to extract important local sequence pattern uses ReLU activation helps in the evolution of nonlinearity [23]. The next layer of the model receives the input from prior layer considered as maximum pooling layer with the size of two. The maximum pooling window was produced by each respective convolution layers with maximum values. The second convolution layer uses 64 filters with kernel size of 7 helps to refine the learned features. The second maximum pooling layer uses with pooling size of 2 helps to select the strongest feature from each convolution window. The flattened output was received from convolution layers passes using four dense layers such as 64,40,40 and 28 neurons was used. To prevent over fitting, and to assure regularization dropout rate of 0.3 is used. The process of regression was performed by using single neuron used in the output layer predicts continues frequency score with indel gRNA activity. The accuracy of the predication was evaluated by using Root Mean Squared Error (RMSE) and Mean Squared Error (MSE) to assess the CNN model were utilized as described in [24].

$$l_t = \tanh\left(x_t * k_t + b_t\right)$$

In the above equation the acronym l_t denotes output value. The activation function is denoted by using tanh. x_t was considered as input vector used. k_t as was used to denote the convolution kernel weight and b_t was used to represent convolution kernel bias.

2.3.2. Long Short Term Memory Model

In the year 1997, LSTM network model was developed [25]. The challenges caused by gradient disappearance and gradient growth occurred in RNN was addressed with the help of LSTM model [26, 27]. The implementation of LSTM model was used for the robot control, speech activity detection, sentiment analysis, text summarization, question answering with Chabot [28, 29]. Moreover, LSTM model helps in predicating the forecasting of stock market helps to provide numerous numbers of benefits [30, 31, 32]. The gradient issues occurred in the model similar to RNN addressed efficiently. Various kind of sequential data such as audio data, natural language and time serious was best suited to work efficiently with LSTM model. The challenges in LSTM model were caused due to existence of long-term dependencies between elements helps in understanding the predication [33, 34]. The use of memory cells was considered as major principle used in LSTM model. The longer text passages was analysed efficiently with the help of LSTM where the sentiment might changes over time. The flow of information in each memory cell was governed by using three critical gates helps to vanish the gradients problem. The flow of information such as let in, forget and output was manage efficiently by using three gates.

The following text explores the working formula and high-level explanation of LSTM

The new data stored in the memory cell was explored by It (Input Gate) controls.

it = sigmoid (Wi * [ht-1, xt] + bi)

The information deleted from the cell was determined by using Forget Gate (ft).

ft = sigmoid (Wf * [ht-1, xt] + bf)

The amount of new data stored in the memory cell was determined by Update Memory Cell (ct)

ct~ = tanh (Wc * [ht-1, xt] + bc)

The integration of new candidate memory cell state with old memory cell state was explored by Final Memory Cell (ct)

ct = it * ct~ + ft * ct-1

The determination of output information from hidden state of the memory cell was explored by Output Gate (ot)

ot = sigmoid (Wo * [ht-1, xt] + bo)

The predications was made by filtered version of the current memory cell state highlighted by Final Hidden State (ht)

ht = ot * tanh(ct)

2.3.3. Feature Representation Optimization

Synthetic Minority Oversampling Technique

The issues caused by the imbalanced data set was addressed by machine learning technique Synthetic Minority Over-sampling Technique (SMOTE). In compared with other technique in addressing the issues the SMOTE model was underrepresented. The workflow of SMOTE initiates by identification of minority class in the dataset with fewer samples compared with others. The minority classs was labelled. To generate the synthetic samples, each data point from minority class was selected. k nearest neighbours (commonly k = 5) are identified only within the minority class was identified for selected minority sample based on the implementation of distance metric such as Euclidean distance. To participate in synthetic sample generation, one of the k nearest neighbours was selected. By taking the weighted advantage of two samples, the process of interpolation was carried out by inserting some random fluctuation with the addition of synthetic samples to dataset helps to increase the balance of class distribution and to increase the number of minority class instances.

Recursive Feature Elimination

The performance of the model was improved by Recursive Feature Elimination (RFE) election strategy uses machine learning helps in the improvement of most significant features. Based on the relevance, essential features was ranked in the model. The deleting of least essential features, rating the features and training the model are the various kind of technique used for training the model. Until the desired number of characteristics was obtained, the technique was repeated. The process of over fitting was reduced by RFE helps in the enhancement of model interpretability. The removing of irrelevant or extra features helps in lowering computational costs increases the overall efficiency and efficiency.

2.3.4. XG-BOOST Model

XG Boost (Extreme Gradient Boosting) was built based on gradient boosting framework. The implementation of many simple methods was combined sequentially to produce final predication. During the process of boosting, objective of XG Boost helps to optimize a specific loss of function.

The two main component was used in the implementation of formula for the XG Boost model. The objective function and predication formula are the two major components used in XG Boost model.

1. Objective Function:

During the process of training, loss function minimization was represented as objective function in XG Boost model. The two major components considered in the objective function are regularization term and loss term. The existence of difference between actual target value and predicated value was highlighted by using loss term. The model complexity was adjusted to avoid over fitting explored by regularization term. The following was the general form of XG Boost

Objective Function = Loss (y_test, y_pred) + Regularization Term

2. Prediction Formula:

The predication from all weak learners was combined by using predication formula helps to produce final predication. The predication of individual trees are added to produce the weighted sum.

The predication formula for XG Boost was rewritten as y_pred = Σ (learning rate * tree weight * predict tree(x))

where

The final prediction for the input sample x was denoted by the acronym y_pred.

The weak leaner contribution to the final predication was identified by hyper parameter denoted by learning rate occurs between the value 0 and 1.

To specific tree position the weight assigned denoted by using tree weight.

Based on the input sample x single decision tree predication was denoted by predict tree(x).

By using step by step process the predication was improved in XG Boost model during training. The occurrence of error in previous tree was corrected in framing new trees. The contribution of each tree in new model was exhibited by learning rate control. The process of over fitting was avoided by using smaller learning model makes the model to organize better.

Many simple decision trees are combined using XBoost builds strong predication model. The weighted predication was contributed by each tree helps to produce the accurate result in compared with single tree. The objective function helps to guide training process minimizes the predication error [35]. The performance of the model was improved by using various techniques such as regularization and column subsampling helps to reduce over fitting making XGBoost fast, accurate and flexible.

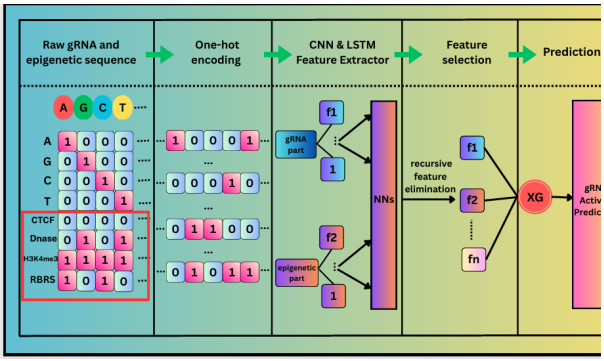

Fig.2. XG Boost model.

2.3.5. CLSTM SMOTE-RFE & XGBOOST model architecture

Figure 2 illustrates the steps to be followed to predict the cell line-specific activity of the guide RNA target with XG boost and CNN-LSTM.

CNN-LSTM was stacked with XG Boost and formed a hybrid predictive system of the gRNA activity of CRISPR/Cas12. The model is superior to those deep learning methods that are based on manually generated features. Based on the dataset, data have been separated into training and testing groups. The gRNA inputs as 23 nucleotide sequences, epigenetic data, and frequencies of indel insertions were transformed into a matrix and inputted into the CNN-LSTM model through one-hot encoding. In the training step, there was hyperparameter optimisation. Subsequently, the XG model itself was trained and measured with the help of features retrieved via CNN-LSTM. All these factors are optimized to achieve the best model performance, such as the maximum depth of each tree, number of trees, as well as the learning rate, by using XG training. Feature extraction is performed by CNN-LSTM model, where the regression score of gRNA activity is generated by XG Boost. This project was done in Tensor flow and deep learning library of Keras.

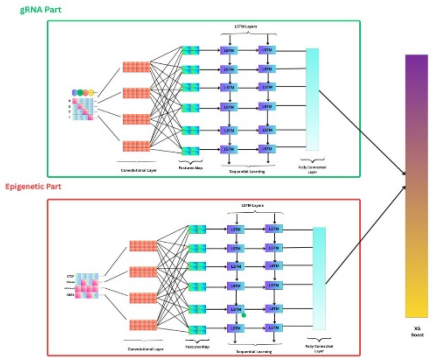

Fig.3. CLSTM SMOTE-RFE & XGBOOST model architecture.

Encoding Sequences

One-hot encoding methods were used for encoding gRNA sequences and epigenetic information. For this, it takes into account epigenetic information of the corresponding region, the nucleotide sequences of gRNA. In total, there are four base pairs: A, G, C and T. These are binary channels that can be utilized to encrypt the base information in a gRNA sequence of 1x23. When the corresponding place of each channel is 1, this denotes that the corresponding nucleotide is available; otherwise, it is 0. To be specific, T-channel nucleotide T was denoted by 1 whenever it appeared in a given position of the base pairs and 0 when it did not. Therefore, each gRNA was represented in 4x23 matrix, which refers to the length of gRNA sequence.

Similarly, the encoding of four bits of epigenetic data was made. Correlation with 23 locations was tabulated into a 4x23 matrix; where, 1 denotes that information is present, and 0 is not present. After this, the gRNA and epigenetic matrices were trained in the convolutional neural networks.

Establishment of CNN-LSTM and XG Models

CNN-LSTM and XG Boost have a sequential layer-by-layer framework governing them. This combined model plays a vital role that retrieves deep data characteristics of gRNA sequences and related epigenetic information of gRNA sequences. The CNNLSTM network shown in Figure 3 entails an epigenetic stream sub-network and a gRNA sub-network to derive the features of the gRNA sequence and epigenetic data. Each of the two sub-networks is composed of layers like, one flattening layer, two (1D) convolutional layers, two max-pooling layers, two LSTM layers, and one fully connected layer.

As an example, gRNA portion multiplies a 4x23 size input matrix of binary data. The first pass of 1D convolution is applied to obtain the characteristics of gRNA with 64 3x3 convolutional kernels. The

convolutional output activation function is then a rectified linear unit (ReLU). The max pooling layer reduces the number of parameters through the usage of filter with window size of two on the preceding layers. The other two convolutional layers use size 3 convolutional kernels of 128 and 256 size, respectively. After the initial layer of pooling, the structures of the max-pooling layer remain the same, followed by the max-pooling layers which are stacked to form LSTM layers, the output of which is flattened to create a single-dimensional vector. It has the following characteristics: four fully interconnected layers with the size of 256, 128, 64 and 32. The "concatenate" operator joins the properties of the four fully interconnected layer of both gRNA sequence and the epigenetic branches. The concatenated layers are inputted into the last fully connected layer of CNN-LSTM network. The CNN-LSTM network XG Boost model is fed with the last output layer of the network. The final output layer is made up of one neuron, which represents the projected score. Over-fitting in the models where the drop rate is 0.2 is prevented by dropout.

Pre-training of CNN-LSTM & XG

In the proposed architecture, CNN-LSTM & XG is a predictive algorithm that can predict gRNA activity. Before the model is trained, the CNN-LSTM pre-training and selection of parameters need to be done first. In the training stage, fivefold cross-validation method was used at random division of 90 percent of the samples of the dataset to be trained and 10 percent to be tested. The training set is arbitrarily split into 5 equal halves. The testing dataset was indicated to be a subset of each training dataset whereas the rest of the four portions were chosen as the training dataset. The use of cross-validation allowed all data sets to be used in the training process, which minimized the effects of overfitting and guaranteed CNN-LSTM&XG accuracy.

Feature Selection of SMOTE-RFE

In the calculation of the threshold, the median of y_train is used. The labels above the threshold are altered to 1 and those under the threshold are altered to 0. This renders continuous labels to be binary. SMOTE is switched on. The epigenetic (epi_train) and sequencing data (seq_train) are remodeled, where it layered horizontally to produce one array of features. SMOTE is utilized to equalize the binary dataset classes, giving x_train_resampled and y_train_binary_resampled. The data that has been resampled is further segmented into sequences and epigenetic data and then reinstated into their original forms. The resultant continuous data is put into an array called y_train_resampled.

XG Boost Model Development

The proposed techniques were developed in Python 3.10.12, Keras library 2.12.0, and Tensor flow version 3.2.2. Best training and testing were done with the Intel(R) Core (TM) i3-1125G4 @ 2.00GHz, and 8 GB RAM/GPU used to accelerate training and testing process.

The optimized features by the random forest are fed to XG Boost classifier [36]. The grid search has been used to change the hyperparameters of the proposed designs: the depth and number of trees, and the minimum weight of the leaf node. In order to optimize hyperparameters, the following numbers were maximized: Depth of six, number of trees 500 and minimum weight of the leaf node is one.

Fig.4. Activity Diagram Of CNN-LSTM SMOTE-RFE & XG Boost Model.

Evaluation of model performance & implementation

The ordinality of data required the calculation of the Spearman correlation coefficient value between the amount of predicted and actual indel frequency thus being used to assess the effectiveness of the models used. Spearman correlation is very effective in natural

finding a quantitative relationship between two variables [37].

Parameters of the Deep Learning Layers

During the deep learning section of the proposed model, CNN and LSTM architectures are combinedly tested. Figure 5 provides the parameters of these architectures with Keras API.

Layer (type)	Output Shape	Param #	Connected to
input_57 (InputLayer)	[(None, 23, 4)]	0	[]
input_58 (InputLayer)	[(None, 23, 4)]	0	[]
seq_conv_1 (Conv1D)	(None, 19, 256)	5376	['input_57[0][0]']
epi_conv_1 (Conv1D)	(None, 19, 256)	5376	['input_58[0][0]']
seq_activation1 (Activation)	(None, 19, 256)	0	['seq_conv_1[0][0]']
epi_activation_1 (Activation)	(None, 19, 256)	0	['epi_conv_1[0][0]']
seq_pooling_1 (AveragePooling1D)	(None, 9, 256)	0	['seq_activation1[0][0]']
epi_pooling_1 (AveragePooling1D)	(None, 9, 256)	0	['epi_activation_1[0][0]']
dropout_392 (Dropout)	(None, 9, 256)	0	['seq_pooling_1[0][0]']
dropout_399 (Dropout)	(None, 9, 256)	0	['epi_pooling_1[0][0]']
seq_conv_2 (Conv1D)	(None, 5, 256)	327936	['dropout_392[0][0]']
epi_conv_2 (Conv1D)	(None, 5, 256)	327936	['dropout_399[0][0]']
seq_activation_2 (Activation)	(None, 5, 256)	0	['seq_conv_2[0][0]']
epi_activation_2 (Activation)	(None, 5, 256)	0	['epi_conv_2[0][0]']
seq_pooling_2 (AveragePooling1D)	(None, 2, 256)	0	['seq_activation_2[0]
epi_pooling_2 (AveragePooling1D)	(None, 2, 256)	0	['epi_activation_2[0]
dropout_393 (Dropout)	(None, 2, 256)	0	['seq_pooling_2[0][0]
dropout_400 (Dropout)	(None, 2, 256)	0	['epi_pooling_2[0][0]
lstm_112 (LSTM)	(None, 2, 128)	197120	['dropout_393[0][0]']
lstm_114 (LSTM)	(None, 2, 64)	82176	['dropout_400[0][0]']
dropout_394 (Dropout)	(None, 2, 128)	0	['lstm_112[0][0]']
dropout_401 (Dropout)	(None, 2, 64)	0	['lstm_114[0][0]']
lstm_113 (LSTM)	(None, 2, 256)	394240	['dropout_394[0][0]']
lstm_115 (LSTM)	(None, 2, 256)	328704	['dropout_401[0][0]']
dropout_395 (Dropout)	(None, 2, 256)	0	['lstm_113[0][0]']
dropout_402 (Dropout)	(None, 2, 256)	0	['lstm_115[0][0]']
flatten_56 (Flatten)	(None, 512)	0	['dropout_395[0][0]
flatten_57 (Flatten)	(None, 512)	0	['dropout_402[0][0]
seq_dense_1 (Dense)	(None, 256)	131328	['flatten_56[0][0]'
epi_dense_1 (Dense)	(None, 256)	131328	['flatten_57[0][0]'
dropout_396 (Dropout)	(None, 256)	0	['seq_dense_1[0][0]
dropout_403 (Dropout)	(None, 256)	0	['epi_dense_1[0][0]
seq_dense_2 (Dense)	(None, 128)	32896	['dropout_396[0][0]
epi_dense_2 (Dense)	(None, 128)	32896	['dropout_403[0][0]
dropout_397 (Dropout)	(None, 128)	0	['seq_dense_2[0][0]
dropout_404 (Dropout)	(None, 128)	0	['epi_dense_2[0][0]
seq_dense_3 (Dense)	(None, 64)	8256	['dropout_397[0][0]
epi_dense_3 (Dense)	(None, 64)	8256	['dropout_404[0][0]
dropout_398 (Dropout)	(None, 64)	0	['seq_dense_3[0][0]
dropout_405 (Dropout)	(None, 64)	0	['epi_dense_3[0][0]
seq_dense_4 (Dense)	(None, 40)	2600	['dropout_398[0][0]']
epi_dense_4 (Dense)	(None, 40)	2600	['dropout_405[0][0]']
concatenate_28 (Concatenate)	(None, 80)	0	['seq_dense_4[0][0]', 'epi_dense_4[0][0]']
prediction (Dense)	(None, 1)	81	['concatenate_28[0][0]']

Total params: 2,019,105
Trainable params: 2,019,105
Non-trainable params: 0

3. Discussion of the findings

In order to predict the CRISPR/Cas12 gRNA activity, this study compared the classical linear model of MLR, CNN-SVR model and hybrid model CNN-LSTM & XG boost model. Although other studies applied ensemble learning to estimate the results, additional tasks such as the integration of various neural networks were found to be more advanced than individual models [38]. Only some studies applied convolution networks to obtain gRNA features to predict activities. In this study, a deep learning model CNN-LSTM and XG Boost was developed for improving the models of state-of-the-art. In the hybrid model, CNN-LSTM is used as a feature extractor to obtain the necessary features of the data and predict with the help of XG Boost. It was applied to predict the activity of guide RNA and to validate the performances of the model. The CRISPR/Cas12 system was applied to predict the guide RNA activity to detect COVID-19.

The CNN-LSTM & XG models was developed initially to predict the CRISPR/Cas12 guide RNA activity in terms of filter length, degree of filters and convolution, and the best model that was achieved was identified based on the minimum mean square error [39]. Moreover, this study compared the performance of different CNN-LSTM & XG model designs to select the CNN-LSTM & XG model with best performance through optimising 2 parameters, the first one being the number of convolution stages and the second one is the optimisation of the number of filter kernel size. With 5-fold cross-validation, 90 percent of the data was employed to train the models, 10 percent of the data employed to test the models, and Spearman correlation statistics to measure performance.

3.1. Comparison of MLR, CNN, CNN-SVR & hybrid CNN-LSTM&XG boost models

The Spearman correlation was applied as the basis of knowledge assessment to compare the traditional MLR model and AI models in order to identify which models would be the most effective to use in predicting gRNA activity. The linear pattern association between the input and the output was determined with the help of traditional MLR, and the nonlinear patterns in the date were identified with the help of the hybrid CNN-SVR models and the CNN to identify them. The CNN-LSTM and XG Boost model that was proposed in this current study combines the guide RNA activity prediction, allowing the CNN-LSTM and XG Boost regressor to be placed in an effective and robust non-linear relationship improving the predictability of guide RNA activity. The relationship between data split and model performance was demonstrated at different data splits. Table 1 and Table 2 have shown that optimal size of data to train and test were 80% and 20% train-test respectively. Based on the performance criteria, (i) Spearman correlation, MLR, and CNN have been found to perform best at 80 and 20 percent train-test splitting respectively; whereas, (ii) CNN-SVR was best at 90 and 10 percent train-test splitting.

Nevertheless, CNN-LSTM & XG worked best at 90 percent and 10 percent train test split. Moreover, the values of the spearman correlation of all the models decline when training data reduces from 70 to 40 percent.

Table 1. Output (Predicted activity score for gRNA) for different models

Model	HEK293T Cells	HCT116	HELA	HL60
MLR	0.1625482	0.161456	0.1623124	0.1654932
CNN	0.2062453	0.1952842	0.2046213	0.2051356
CNN-SVR	0.2274344	0.2056872	0.2145387	0.2024786
CNN-LSTM&XG BOOST	0.25936452	0.2732677	0.273958	0.2560974

Table 2. Spearman correlation for four different models with variable train-test splits.

Train-Test split	MLR	CNN	CNN-SVR	CNN-LSTM&XG
40–60	0.405	0.451	0.498	0.781
50–50	0.454	0.456	0.514	0.776
60–40	0.475	0.463	0.573	0.794
70–30	0.545	0.52	0.606	0.783
80–20	0.589	0.565	0.709	0.792
90–10	0.54	0.516	0.748	0.812

Table 3. Evaluation Metrics for four different cells

Evaluation Metrics	HEK293T	HCT116	HeLa	HL60
Accuracy	0.834	0.847	0.824	0.829
Precision	0.904	0.911	0.9	0.909
Recall	0.912	0.915	0.904	0.902
Specificity	0.155	0.1	0.109	0.088
F1-score	0.908	0.913	0.902	0.906
RMSE	0.287	0.282	0.213	0.276
MSE	0.082	0.079	0.081	0.072

Even though the MLR, CNN, and CNN based-SVR are all applicable to predicting gRNA activity, the hybrid CNN-LSTM&XG model is the most effective, as it is evident in Table 2. The CNN-SVR and CNN had the highest Spearman correlation of 0.748 and the lowest of 0.565 respectively of their train-test splits. MLR had 0.589 as the highest value of Spearman correlation; and the hybrid CNN-LSTM-XG boost model was a good predictor compared to CNN-SVR, which had a Spearman correlation of 0.81. These findings indicate that enhancement of the predictive performance of CNN model is possible as the hybrid model has the highest predictive performance.

Guide RNA activity predictive deep learning models are also available, primarily based on the CRISPR Cas9 system, but not on the CRISPR Cas12 systems. Consequently, additional computational support of the CRISPR Cas12 system is needed.

3.2. Training model visualisation

It is important to know the significance of each nucleotide in the guide RNA sequence in each position and epigenetic information. To achieve it, the gRNA sequences and the epigenetic kind of information of 23nt elongation were developed, and each of them contains only one of the four nucleotides that have previously been reported [40]. The remaining three nucleotides of this sequence were changed into 0. The current study's trained model CNN-LSTM&XG was presented with the sequences and detected the frequency of indel. The process was repeated with the four nucleotides and their frequencies of indel were determined.

To visualise the relevance of the guide RNA frequencies and the epigenetic data during each position, a heatmap is generated with the frequencies of the indel of the respective nucleotide at each position paired with the epigenetic data. When the nucleotide is darker red on the heatmap, such as the case with Fig. 6 it is more necessary there. To better understand the nucleotides dissimilarity between active and inactive gRNAs, the study applied the use of Kp Logo web application [41] to visualise them (Fig. 7).

Fig.5. False positive rate heatmap.

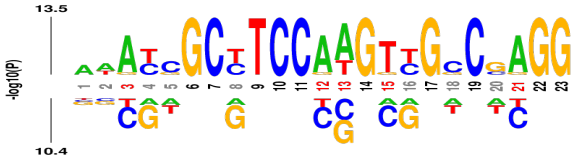

Fig.6. Visualization of Kp Logo web application

Activity of CRISPR/Cas9 gRNA Preference are influenced by the nucleotides present in active guide RNAs (top) and those present inactive guide RNAs (bottom). Active gRNAs have preference to adenine (A) in the (top) first position, whereas inactive gRNAs have preference to thyine (T) in the (bottom) first position. There is also an active preference of guanine (G) over cytosine (C) in the last position by active gRNAs. Generally, active guide RNAs show Adenine (A) preference in preference over the thymine (T) when translating the guide RNA.

The search results are narrowed down on the COV-2 virus gene to verify this study's trained model and check its efficiency by using the search query TTTN-N23 sequence. The genome sequencing was obtained in [42]. A Cas12-based system to detect SARS-CoV-2. Due to the similarity of the target location of Cas12 guide RNA to the TTTN sequence, the study identified all 23nt sequences following the occurrence of TTTN. The strategy was applied to assemble guide RNA sequences of the SARS-CoV-2 gene; the guide RNAs have the ability to bind to the matching sequences within the SARS-CoV-2 gene and then cut up the viral gene, producing signals on the existence of the COVID-19 virus [42]. Our model was fed these sequences and the indel frequencies of these sequences were calculated.

Based on the experiment, adenine or thymine in the first or final position constitutes the gRNA sequences and epigenetic information having a high rate of indel. Although guanine is more common in the seed of the gRNA with low rate of indel, it is heavily discouraged in position 1. This result confirms the effectiveness of our CNN-LSTM SMOTE-RFE &XG model, which will indicate the activity of the guide RNAs in advance to analyze the viral presence and can save time and money, as well as find an actual type of price detection of the virus. The darker blue the nucleotide is in the heatmap, the more essential the nucleotide is in that position. In order to determine the difference in nucleotide between active and inactive gRNA better, Kp Logo was utilized.

4. Conclusion

The gRNA activity of the CRISPR/CAS 12 system is challenging to predict accurately and can only be utilized effectively. Nonlinear models are better than linear to predict gRNA activity, and hybrid models provide more promising outcomes. The guide RNA sequence and epigenetic information were processed with CNN-LSTM SMOTE-RFE and an XG was conducted to produce the predictions of the guide RNA activity. The study also performed tests to ascertain the importance of a nucleotide within a guide RNA sequence and epigenetic data, and lastly, hybrid model's performance was tested against one-at-state-of-the-art models and predict the occurrence of indel in guide RNA sequences that will be used to identify the SARS-CoV-2 gene using the CRISPR/Cas12 system.

References

[1] F. J. Mojica, C. Díez-Villaseñor, E. Soria, and G. Juez, "MicroCorrespondence," *Molecular Microbiology*, vol. 36, no. 1, 2000.

[2] R. Barrangou, C. Fremaux, H. Deveau, M. Richards, P. Boyaval, S. Moineau, and P. Horvath, "CRISPR provides acquired resistance against viruses in prokaryotes," *Science*, vol. 315, no. 5819, pp. 1709–1712, 2007.

[3] T. Gaj, C. A. Gersbach, and C. F. Barbas, "ZFN, TALEN, and CRISPR/Cas-based methods for genome engineering," *Trends in Biotechnology*, vol. 31, no. 7, pp. 397–405, 2013.

[4] M. Jinek, K. Chylinski, I. Fonfara, M. Hauer, J. A. Doudna, and E. Charpentier, "A programmable dual-RNA–guided DNA endonuclease in adaptive bacterial immunity," *Science*, vol. 337, no. 6096, pp. 816–821, 2012.

[5] A. M. Chakrabarti, T. Henser-Brownhill, J. Monserrat, A. R. Poetsch, N. M. Luscombe, and P. Scaffidi, "Target-specific precision of CRISPR-mediated genome editing," *Molecular Cell*, vol. 73, no. 4, pp. 699–713, 2019.

[6] R. Barrangou and J. A. Doudna, "Applications of CRISPR technologies in research and beyond," *Nature Biotechnology*, vol. 34, no. 9, pp. 933–941, 2016.

[7] J. P. Broughton, X. Deng, G. Yu, C. L. Fasching, V. Servellita, J. Singh, and C. Y. Chiu, "CRISPR–Cas12-based detection of SARS-CoV-2," *Nature Biotechnology*, vol. 38, no. 7, pp. 870–874, 2020.

[8] B. K. Romanov, "Coronavirus disease COVID-2019," *Safety and Risk of Pharmacotherapy*, vol. 8, no. 1, pp. 3–8, 2020.

[9] World Health Organization, *Coronavirus Disease 2019 (COVID-19): Situation Report 123*, Geneva, Switzerland, 2020.

[10] H. K. Kim, M. Song, J. Lee, A. V. Menon, S. Jung, Y. M. Kang, and H. Kim, "In vivo high-throughput profiling of CRISPR–Cpf1 activity," *Nature Methods*, vol. 14, no. 2, pp. 153–159, 2017.

[11] H. Zhu and C. Liang, "CRISPR-DT: Designing gRNAs for the CRISPR-Cpf1 system with improved target efficiency and specificity," *Bioinformatics*, vol. 35, no. 16, pp. 2783–2789, 2019.

[12] F. J. Huang and Y. LeCun, "Large-scale learning with SVM and convolutional for generic object categorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2006, vol. 1, pp. 284–291.

[13] K. P. Oliveira-Esquerre, D. E. Seborg, R. E. Bruns, and M. Mori, "Application of steady-state and dynamic modeling for the prediction of the BOD of an aerated lagoon at a pulp and paper mill: Part I. Linear approaches," *Chemical Engineering Journal*, vol. 104, nos. 1–3, pp. 73–81, 2004.

[14] X. X. Niu and C. Y. Suen, "A novel hybrid CNN–SVM classifier for recognizing handwritten digits," *Pattern Recognition*, vol. 45, no. 4, pp. 1318–1325, 2012.

[15] K. P. Oliveira-Esquerre, D. E. Seborg, R. E. Bruns, and M. Mori, "Application of steady-state and dynamic modeling for the prediction of the BOD of an aerated lagoon at a pulp and paper mill: Part I. Linear approaches," *Chemical Engineering Journal*, vol. 104, nos. 1–3, pp. 73–81, 2004.

[16] D. S. Vijayakumar and M. Sneha, "Low cost COVID-19 preliminary diagnosis utilizing cough samples and keenly intellective deep learning approaches," *Alexandria Engineering Journal*, vol. 60, no. 1, pp. 549–557, 2021.

[17] J. Farooq and M. A. Bazaz, "A deep learning algorithm for modeling and forecasting of COVID-19 in five worst affected states of India," *Alexandria Engineering Journal*, vol. 60, no. 1, pp. 587–596, 2021.

[18] G. Chuai, H. Ma, J. Yan, M. Chen, N. Hong, D. Xue, and Q. Liu, "DeepCRISPR: Optimized CRISPR guide RNA design by deep learning," *Genome Biology*, vol. 19, no. 1, p. 80, 2018.

[19] L. Xue, B. Tang, W. Chen, and J. Luo, "Prediction of CRISPR sgRNA activity using a deep convolutional neural network," *J. Chem. Inf. Model.*, vol. 59, no. 1, pp. 615–624, 2019.

[20] H. K. Kim, Y. Kim, S. Lee, S. Min, J. Y. Bae, J. W. Choi, and H. H. Kim, "SpCas9 activity prediction by DeepSpCas9," *Science Advances*, vol. 5, no. 11, eaax9249, 2019.

[21] J. Luo, W. Chen, L. Xue, and B. Tang, "Prediction of activity and specificity of CRISPR-Cpf1 using convolutional deep learning neural networks," *BMC Bioinformatics*, vol. 20, no. 1, p. 332, 2019.

[22] B. Li, D. Ai, and X. Liu, "CNN-XG: A hybrid framework for sgRNA on-target prediction," *Biomolecules*, vol. 12, no. 3, p. 409, 2022.

[23] E. A. Feingold *et al.*, "The ENCODE (ENCyclopedia of DNA elements) project," *Science*, vol. 306, no. 5696, pp. 636–640, 2004.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] V. D. Ta, C. M. Liu, and D. A. Tadesse, "Portfolio optimization-based stock prediction using long short-term memory network," *Applied Sciences*, vol. 10, no. 2, p. 437, 2020.

[26] O. Zarrad, M. A. Hajjaji, and M. N. Mansouri, "Hardware implementation of hybrid wind-solar energy system for pumping water based on artificial neural network controller," *Studies in Informatics and Control*, vol. 28, no. 1, pp. 35–44, 2019.

[27] T. Šarić, G. Šimunović, Đ. Vukelić, K. Šimunović, and R. Lujić, "Estimation of CNC grinding process parameters using different neural networks," *Tehnički Vjesnik*, vol. 25, no. 6, pp. 1770–1775, 2018.

[28] N. Gupta and A. S. Jalal, "Integration of textual cues for fine-grained image captioning using deep CNN and LSTM," *Neural Computing and Applications*, vol. 32, no. 24, pp. 17899–17908, 2020.

[29] A. Yadav, C. K. Jha, and A. Sharan, "Optimizing LSTM for time series prediction in Indian stock market," *Procedia Computer Science*, vol. 167, pp. 2091–2100, 2020.

[30] H. Y. Kim and C. H. Won, "Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models," *Expert Systems with Applications*, vol. 103, pp. 25–37, 2018.

[31] N. C. Petersen, F. Rodrigues, and F. C. Pereira, "Multi-output bus travel time prediction with convolutional LSTM neural network," *Expert Systems with Applications*, vol. 120, pp. 426–435, 2019.

[32] Z. Jin, Y. Yang, and Y. Liu, "Stock closing price prediction based on sentiment analysis and LSTM," *Neural Computing and Applications*, vol. 32, no. 13, pp. 9713–9729, 2020.
[33] S. Borovkova and I. Tsiamas, "An ensemble of LSTM neural networks for high-frequency stock market classification," *Journal of Forecasting*, vol. 38, no. 6, pp. 600–619, 2019.
[34] T. Chen, "XGBoost: A scalable tree boosting system," Cornell Univ., Ithaca, NY, USA, 2016.
[35] A. W. Whitney, "A direct method of nonparametric measurement selection," *IEEE Trans. Computers*, vol. C-20, no. 9, pp. 1100–1103, 1971.
[36] M. M. Mukaka, "A guide to appropriate use of correlation coefficient in medical research," *Malawi Medical Journal*, vol. 24, no. 3, pp. 69–71, 2012.
[37] Z. S. I. Ameen, M. Ozsoz, A. S. Mubarak, F. Al Turjman, and S. Serte, "C-SVR Crispr: Prediction of CRISPR/Cas12 guideRNA activity using deep learning models," *Alexandria Engineering Journal*, vol. 60, no. 4, pp. 3501–3508, 2021.
[38] S. I. Z. Amee, A. S. Mubarak, A. Süleyman, and O. Mehmet, "Development of CNN model for prediction of CRISPR/Cas12 guide RNA activity," in *Proc. Int. Conf. Theory Appl. Soft Computing*, Cham, Switzerland: Springer, 2019, pp. 697–703.
[39] G. Zhang, Z. Dai, and X. Dai, "A novel hybrid CNN-SVR for CRISPR/Cas9 guide RNA activity prediction," *Frontiers in Genetics*, vol. 10, p. 1303, 2020.
[40] X. Wu and D. P. Bartel, "kpLogo: Positional k-mer analysis reveals hidden specificity in biological sequences," *Nucleic Acids Research*, vol. 45, no. W1, pp. W534–W538, 2017.
[41] J. Luo, W. Chen, L. Xue, and B. Tang, "Prediction of activity and specificity of CRISPR-Cpf1 using convolutional deep learning neural networks," *BMC Bioinformatics*, vol. 20, no. 1, p. 332, 2019.