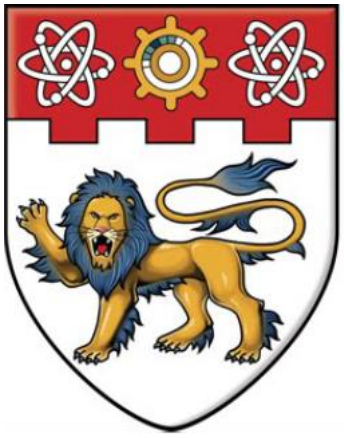

NANYANG TECHNOLOGICAL UNIVERSITY

SCHOOL OF BIOLOGICAL SCIENCES

BS6204 Deep Learning for Biomedical Science

Muhammad Irfan Bin Hajis

Olivia Loh Jia Hui

Cheung Fat Kei

02 November 2023

Introduction

Gene editing technologies allow researchers to modify the genome of an organism of interest. Genome-wide editing techniques can be interpreted as methods where DNA sequences are changed by deletions, mRNA processing, and post-transcriptional modifications to result in altered gene expression, leading to changes in the functional behaviour of proteins and act as valuable research outcomes for the researchers. Type VI CRISPR enzymes are programmable RNA-guided, RNA-targeting Cas proteins with nuclease activity. CRISPR Cas13 allows for target gene knockdown without changing the genome. Cas13 proteins are directed to their target RNAs by a single CRISPR RNA (crRNA). A single crRNA consists of direct repeat (DR) stem loop and a spacer sequence that mediates target recognition by RNA-RNA hybridization. Cas13 enzymes are known to exert some nonspecific collateral nuclease activity on activation. Cas13 enzymes also have drastically reduced off-target activity in cultured cells compared with RNA interference (RNAi). (Wessels et al., 2023) Cas13a has been shown to be useful particularly as a molecular diagnostics tool, including for Severe Acute Respiratory Syndrome Coronavirus-2 (SARS Cov-2) in the case of Specific High Sensitivity Enzymatic Reporter UnLOCKing (SHERLOCK). (Kellner et al., 2019)

We develop a Deep Learning model to predict for Cas13a efficacy in target sequences with an accuracy of 60%. There are currently no deep learning models developed to predict for effective and perfect matching Cas13a guide RNA (gRNA). Previous models have shown that a mismatch between the target sequence and Cas13a gRNA decreases the knockdown effectiveness (Metsky et al., 2022). In our report, we tested various Deep Learning Methods using various inputs to train a model to predict for Cas13a knockdown effectiveness.

Methods

Data Collection. We collected a total of 555 gRNAs across 4 knockdown experiments using engineered variants of LwaCas13a from Abudayyeh et al. (178 from Gaussia luciferase, 93 from Cypridinia luciferase, 93 from KRAS-1 and 93 from MALAT1). (Abudayyeh et al., 2017) Due to the experimental noise native to the data source methodology, a significant amount of the experimental replicates for a specific guide differed significantly in transcript expression, often by more than 25% (as observed in Figure2). We performed Grubbs' Test to detect any outlier, with a significance level of $\alpha = 0.05$. To select and remove outliers, an outlier is defined when a triplicate normalized expression (NE) score has a Coefficient of Variance > 0.1 and has a Z-score > 1 for the particular replicate. If two replicates have a z-score > 1, we select

replicates with the smaller difference for downstream analysis. We averaged each triplicate and negative-transformed the mean-normalized of each experiment (mean of each transcript - triplicate_mean). We defined poor performing gRNAs as 0 (negative mean-normalized < 0) and effective gRNAs as 1 (negative mean-normalized > 1). We tested inputs of various lengths, 0-10 upstream and downstream of the gRNA. We split the dataset into 75%-15%-20% for training-validation-test respectively.

Deep learning models. In this study, artificial neural network (ANN), convolutional neural network (CNN) and bi-directional Long Short-Term Memory (bi-LSTM) were implemented. ANN consists of dense neural network blocks which carry an array of weights and biases and form multiple "hidden layers" of extracting information (features) from input automatically before generating output. The structure elements of a CNN comprises three kinds of layers: convolution layers, pooling layers, and fully connected layers. In the convolution layers, weight vectors called filters are multiplied across the subregions of all the data. These enable CNN to identify the locally correlated patterns irrespective of their location in the data. The design works well for sequence data as in array input format. Bi-directional LSTM designed as a recurrent neural network which works well for sequence data such as language, RNA sequence and music, etc. With advancement from traditional LSTM, bi-directional LSTM consisted of two layers of LSTM as forwards and backwards directions. This enables the design to gather input information from both ends to enhance the richness of the data. This approach aligns with RNA sequence.

Core Model. The model comprises three layers of dense network forming three "hidden layers" with ReLu function as activation. Each dense network is followed by a dropout. The model concludes with an output layer with 2 units and a softmax activation function for binary classification. We defined the core model as the Multi Layer Perceptron (MLP). **Bi-directional Long Short-Term Memory (Bi-LSTM).** We added a layer that consists of a bi-directional LSTM layer followed by a dropout to the core model. **Convolutional Neural Network.** We added a 2D convolution layer followed by a batch normaliaztion layer and a 2D max-pooling layer to the core model. **Hyperparameters.** We used a learning rate of 0.0025, a batch size of 128, Adam optimizer and epoch size of between 80-120.

Loss Function and Metric. In our study, we wanted to maximize True Positive reported and minimize False Positive reported since a false effective gRNA will be costly to test for validation in wet lab experiments. We implemented a custom loss function and metric to assess

the performance of a binary classification model. The custom approach is designed to prioritize the correct identification of positive instances. We designed a "custom_loss" function to balance overall classification accuracy with the emphasis on minimizing FP. This loss function incorporates a weight factor (weight_fn) that allows for prioritization of TP recognition by assigning higher importance to minimizing FN. The custom loss is calculated as the sum of binary cross-entropy loss and weighted false negatives: custom_loss = bce + weight_fn * FN. We found a weight_fn of 10 sufficiently balances the number of FP and FN.

Results and Discussion

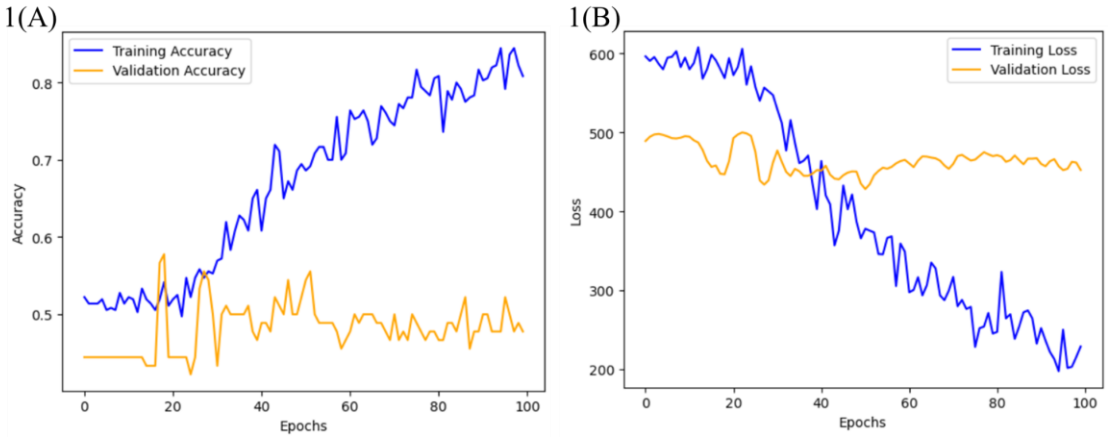

Figure 1. (a) Accuracy Plot (b) Loss Plot Generated during CNN training. Blue indicates the training set, Orange indicates the validation set used during model training

Convolutional Neural Network Result. For the CNN model, it can learn the patterns and variables in the training dataset. As shown in Figure 1(a), it is evident that training is in progress with significant changes in the line plot. The training starts at 0.47 accuracy in epoch 1 and steadily improves to reach an accuracy of 0.90. Subsequently, validation is conducted to assess the model's performance both during and after training. The validation set serves as a separate, unseen dataset to help evaluate how well the model generalises to new, unseen data. Unfortunately, the model does not yield the expected results as the validation loss remains stable and fluctuates around 450 (Figure 1(B)). It can be concluded that the model is experiencing overfitting, as the CNN model gives accurate predictions for the training data but not for new data. Overfitting occurs when the model cannot generalise and fits too closely to the training dataset. Overfitting in this study might be due to the small size of the training dataset (450 data points), which does not contain enough samples to accurately represent all

possible input data values. Increasing the number of samples in the dataset may overcome overfitting.

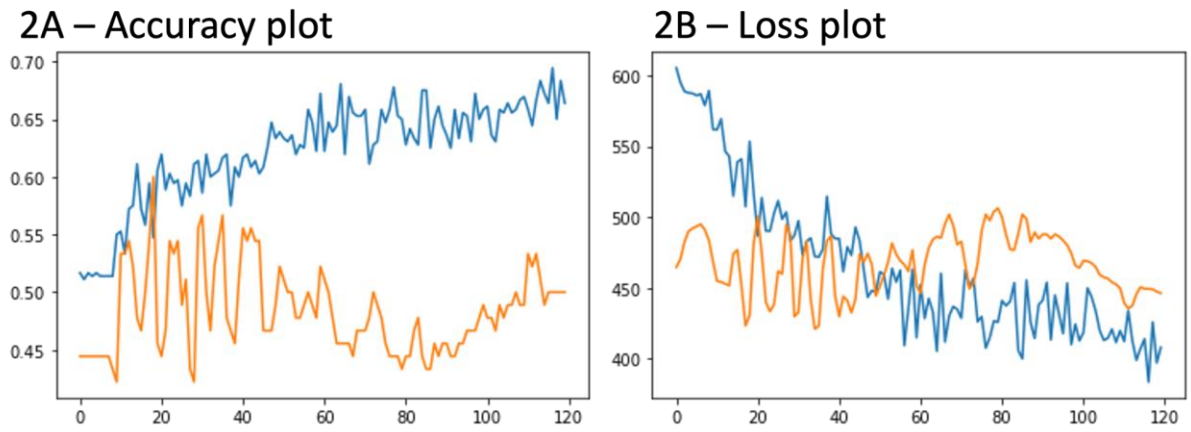

Figure 2. (a) Accuracy Plot (b) Loss Plot generated during MLP training. Blue indicates the training set and orange indicates the validation set used during model training

Multi Layer Perceptron Results. MLP model appeared to learn in the training data set but exhibited poorly for the validation set which only showed learning starting from the 80th epoch. This signified overfitting (Fig 2A). The loss plot (Figure 3B) also showed consistent results of overfitting. Improvement was found from the 80th epoch onwards with loss drop aligning with the training curve. Overall, ANN showed worse performance than the CNN model.

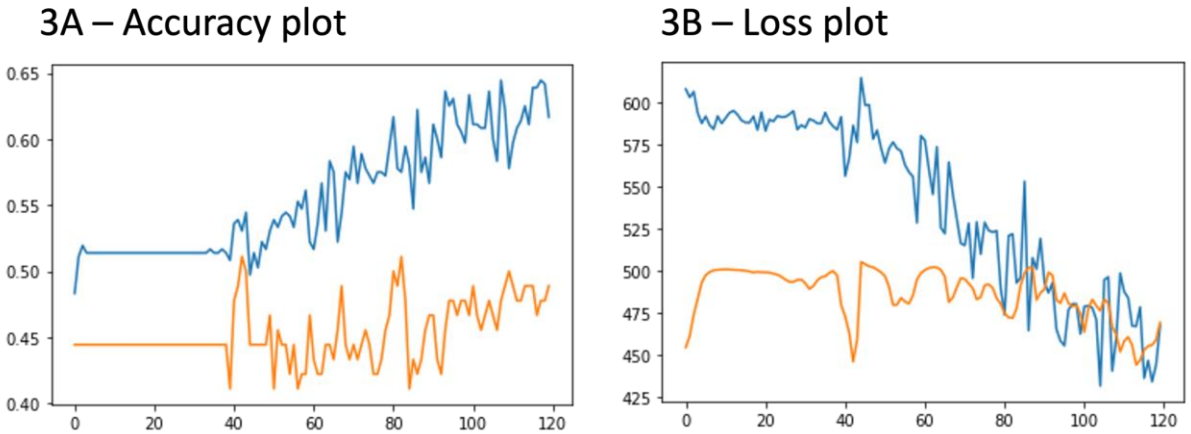

Figure 3. (a) Accuracy Plot (b) Loss Plot generated during bi-LSTM training. Blue indicates the training set and orange indicates the validation set used during model training

Bi-directional Long Short-Term Memory Results. bi-LSTM model showed that the model was learning the pattern in the training data set only after 40th epoch up to accuracy of 0.62 and exhibited poorly for the validation set which only showed learning starting from the 80th

epoch. This signified overfitting (Fig 3A). The loss plot (Figure 3B) also showed consistent results with the accuracy plot with improvement found from the 80th epoch onwards. Overall, bi-LSTM showed the worst performance compared to CNN and ANN model.

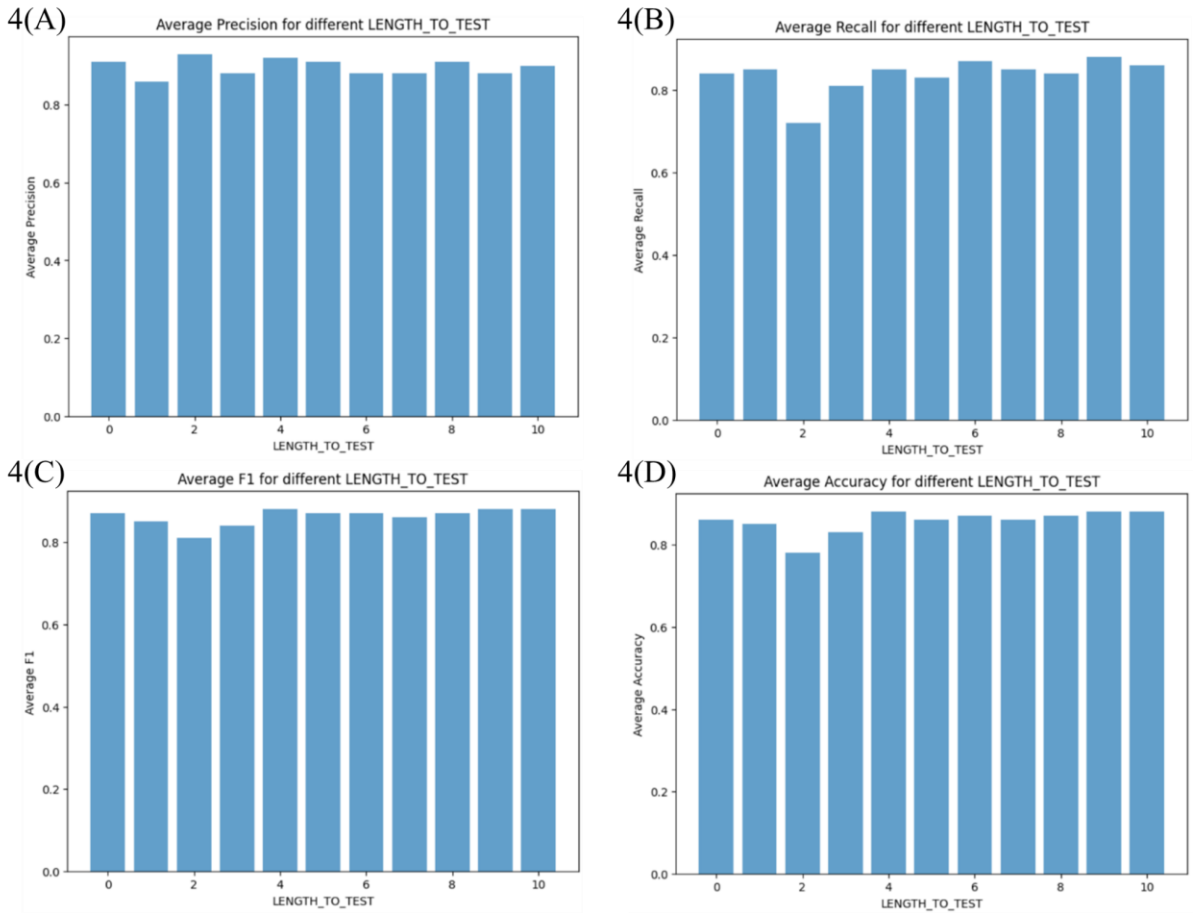

Figure 4. Average (n=3) (A) Precision Plot, (B) Recall Plot, (C) F1 score Plot, (D) Accuracy Plot for different input upstream and downstream lengths (0-10)

Various Input Lengths. We also tried the CNN model with different lengths to determine whether changes in the lengths would generate better results. In the initial model, we set the input lengths as 3, and we determined the accuracy of the predictions using evaluation metrics to measure classification performance, such as precision, recall, F1 score, and accuracy. Precision is a metric that gives the proportion of true positives to the total positives that the model predicts, suggesting that around 80% of the predicted positives are true positives. Recall focuses on how well the model can find all the positives. The F1 score is a measure that combines recall and precision. Since there is a trade-off between precision and recall, the F1 score measures how effectively our models make that trade-off. Accuracy shows how often the model is correct overall. From Figure 4, it can be seen that the input lengths do not substantially

affect the precision, recall, F1 score, and accuracy of the model. The average metrics are around 80% for all the evaluation metrics.

Improvement using Additional Inputs. We added two additional inputs, predicted crRNA folding of the gRNA, and unpaired probability of a position (within the gRNA sequence) with a window length of 50 (ie. at position *i* in the target sequence, unpaired probability for *i*-25 to *i*+25). The crRNA secondary structure and MFEs were derived using RNAfold [–-gquad] on the full-length gRNA (DR + guide) sequence. Direct Repeat (DR) Sequence, GATTTAGACTACCCCAAAAACGAAGGGGACTAAAAC, was appended to the 5' of the gRNA. Target RNA unpaired probability (accessibility) was calculated using RNAplfold. We extracted the unpaired probability of 50 bases within the gRNA target sequence. We also adjusted the Conv2d to accept a 2d layer of (34, 3) where each row represents an input from the sequence, unpaired probability and the predicted crRNA folding. We represented each nucleotide as {'A':1, 'C':2, 'G':3, 'T':4, 'N':0} ('N' because the gRNA maybe located at the 5' or 3' of the target sequence), and brackets (where the gRNA segment folds) as 1 and unfolded segments as 0. An example of a predicted crRNA folding is .....((((.....)))).(((.(((.....))).)))..... where we would only take .....))).)))..... as input (from index 37th to 65th).

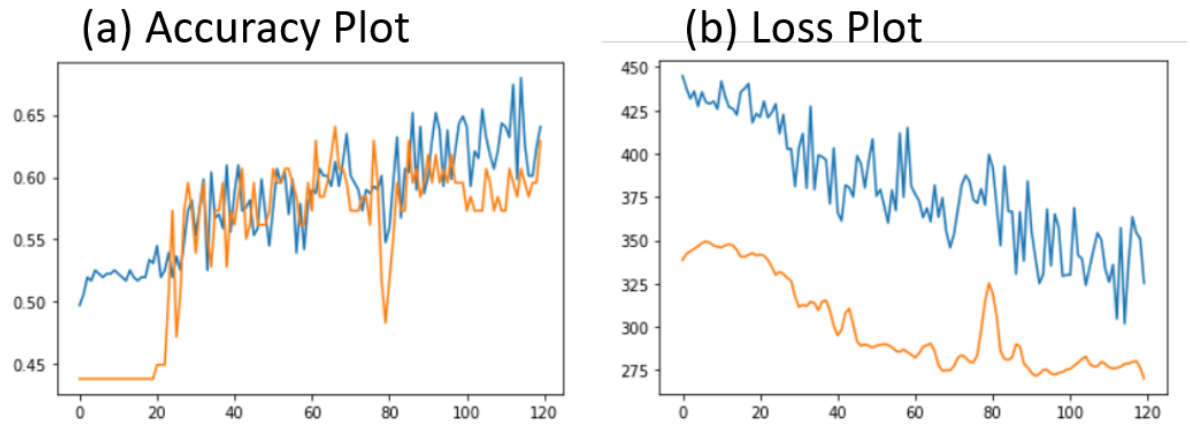

Figure 5. (a) Accuracy Plot (b) Loss Plot generated during CNN training with improved inputs. Blue indicates the training set and orange indicates the validation set used during model training

From Figure 5, we observe that accuracy for the training dataset (0.5078 to 0.640) and validation (0.4382 to 0.6292) is improving gradually while the loss (475.5428 to 325.2909) and validation loss (338.6130 to 270.2071) is decreasing steadily. We also observe that the training

accuracy and validation accuracy overlaps indicating that the training process is not overlapping. This indicates that overfitting is not occurring due to the added inputs (crRNA folding and target RNA unpaired probability). We tried to increase the epoch size from 120 to 150. However, the model tends to overfit towards the end (post 120 epochs).

	Initial Model	Improved Model
Accuracy	0.571429	**0.609524**
F1-score	0.516129	**0.559140**
Precision	0.533333	**0.577778**
Recall	0.500000	**0.541667**

Table 1. Metrics generated when the models were tested against the test set (n=105)

From the table above, we compared the results using the test data generated from both models (One hot encoding of the Target Sequence input vs Multi Layered Input) and there is a slight improvement across all metrics (accuracy, precision, recall and F1-score). The initial model generated a much lower score when the test set is tested against the training dataset. Although the testing data metrics performed worse, we are more confident of the model with multiple inputs as the training is not overfitting the training data set. The improved model is also 20% better than a random coin flip.

Conclusion

In our study, we have reinstituted the difficulty and the different challenges to build a model in the biological field using publicly available data. The challenge could possibly stem from the unaccounted noise such as global RNA folding and protein occupancy on the mRNA. mRNA have been shown to adopt unique structures in different environmental conditions. Another challenge includes insufficient data points (n=450) for training and the high variability in the raw data posed a challenge while normalising the results across the 5 mRNA knockdown experiments. Abbudayyeh used an arrayed screening method, which is more laborious, which may have led to such a high variability and fewer data points. We identified that a different normalisation approach could lead to different results (Not shown in the report).

We have tried various methods and inputs to build a model to predict for effective Cas13a gRNAs and shown that Convolution Neural Network works better over bi-LSTM and Multi Layer perceptron (MLP) with a nucleotide sequence in a one-hot encoding format. We have also shown that different lengths of upstream and downstream sequence did not make a lot of difference. We have also shown that sequence generally contains a lot of noise, possibly due to the one-hot encoding, which may consequently lead to an overfitted model. When we added predicted crRNA folding and unpaired probabilities as input, the model accuracy and precision rate improved to 61% and 58% respectively. Despite the drawbacks such as cost and effort, we believe that generating more data points may help develop a more accurate model, possibly using a pooled screen approach such as those reported in Wessels and Metsky.

Data Availability. The source code, raw data and results are available at https://github.com/micro-irfan/Cas13AgRNAtor/Deep_Learning under the GPLv3 licence.

Reference

Abudayyeh, O., Gootenberg, J., Essletzbichler, P. *et al.* RNA targeting with CRISPR–Cas13. *Nature* **550**, 280–284 (2017). https://doi.org/10.1038/nature24049

Kellner, M.J., Koob, J.G., Gootenberg, J.S. *et al.* SHERLOCK: nucleic acid detection with CRISPR nucleases. *Nat Protoc* **14**, 2986–3012 (2019). https://doi.org/10.1038/s41596-019-0210-2

Kim, H., Min, S., Song, M. *et al.* Deep learning improves prediction of CRISPR–Cpf1 guide RNA activity. *Nat Biotechnol* **36**, 239–241 (2018). https://doi.org/10.1038/nbt.4061

Metsky, H.C., Welch, N.L., Pillai, P.P. *et al.* Designing sensitive viral diagnostics with machine learning. *Nat Biotechnol* **40**, 1123–1131 (2022). https://doi.org/10.1038/s41587-022-01213-5

Wessels, HH., Stirn, A., Méndez-Mancilla, A. *et al.* Prediction of on-target and off-target activity of CRISPR–Cas13d guide RNAs using deep learning. *Nat Biotechnol* (2023). https://doi.org/10.1038/s41587-023-01830-8