# CRISPR guide RNA design by deep learning (DeepCRISPR)

A graduation project for the students:

Ahmed El nouby
Rawan Sayed
Reham Abd El Fatah
Sara Adel

Supervised by Dr. Ibrahim Youssef

Industrial Partner: Proteinea

# Acknowledgement

We would like to thank Proteinea which is an insect-based biotech start-up for their ultimate support that they have given to us and the AWS machine which they offered to train and test our model.

# Abstract

Clustered Regularly Interspaced Short Palindromic Repeats (CRISPR) is a revolutionary genetic editing system that can be used to modify the genome of any living organism. DeepCRISPR is a data-driven algorithm that fully automates the identification of sequence and epigenetic features that may affect sgRNA knockout efficacy. A major challenge for effective application of CRISPR systems is to accurately predict the single guide RNA (sgRNA) on-target knockout efficacy, which would allow for the optimum design of sgRNAs with high sensitivity and specificity. The Models we re-implemented are the on-target classifier, the on-target regressor and the auto encoder which will be used to fine-tune the previous models. Six different testing scenarios were carefully designed for complete and objective comparisons of DeepCRISPR with various ROC-AUC / Spearman Correlation values according to classification / regression models. Comparing against the previous study, our results from the 6 scenarios are almost the same but there is around 10% difference due to the gap between the data we achieved and the real data that they used. A common concern with deep models is that they can overfit the training data. Therefore, we enhanced our model by using dropout and attention techniques to reduce the overfitting problem and enhanced it by a percentage of 85%.

# List of Figures



# List of Tables

# Table of Contents

# 1. Introduction

## a. Biological Background

Every cell in our body has a copy of our genome, which consists of more than 20,000 genes and 3 billion letters of deoxyribonucleic acid (DNA). DNA is consisting of two strands twisted together into a double helix (**Figure 1**) and held together by a basic pairing rule in which Adenine (A) bases on one strand form a double hydrogen bond with Thymine (T) bases on the opposite strand, and Cytosine (C) bases on one strand form a triple hydrogen bond with Guanine (G) bases on the opposite strand.

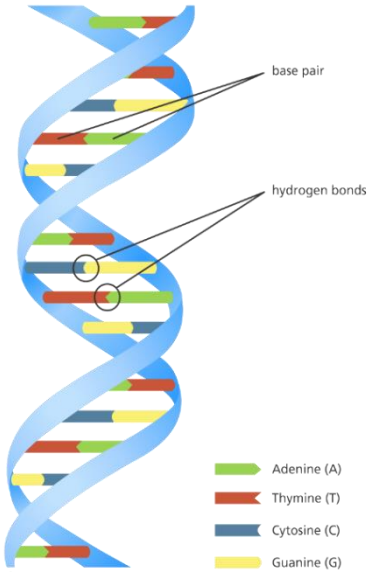


**Figure 1: DNA structure**

Our genes not only shape who we are as individuals and as a species, but they also have a significant impact on our health, and thanks to advances in DNA sequencing, researchers have discovered thousands of genes that influence our disease risk, but they still need ways to manipulate genes to understand how they function.

Clustered Regularly Interspaced Short Palindromic Repeats (CRISPR) is a ground-breaking genetic editing system that has been designed recently to enhance our ability to alter the genome of any living organism, although gene editing in living cells is difficult. [1]

The CRISPR approach is based on a natural mechanism that bacteria use to defend themselves against virus infection. When bacteria detect virus 'viral' DNA, it generates two forms of short ribonucleic acid (RNA) and these two RNAs form a complex with a protein called CAS9.

Guide RNA (gRNA) is one of these two RNAs and it consists of two components:

1. The trans-activating RNA (tracrRNA) which is responsible for Cas9 endonuclease activity
2. The CRISPR RNA (crRNA) binds to the target-specific DNA region

Then they joined with each other forming Single Guided Ribonucleic Acid (sgRNA).

sgRNA has been designed to locate and bind to a specific sequence in the DNA so it has a sequence (~20 nucleotides) that matches the invading virus to make sure that it will only bind to the target sequence and no other unwanted regions.

CAS9 is a nuclease which is a type of enzyme that follows the sgRNA to the desired location then it untwists the DNA and compares it to its target RNA; if the match is successful, the cas9 will cut the DNA with two tiny molecular scissors. When this occurs, the cell attempts to repair the cut, but the repair process is error-prone, leading to mutations that can disable the gene allowing the researcher to understand its function.[2]

Researchers have been studying the CRISPR approach for a few years and have discovered that by modifying the sgRNA to fit the target sequence, it can be designed to cut any DNA sequence at a specific location not only viral DNA. This can be achieved either in a test tube or within a living cell's nucleus.

Once the CRISPR be inside the nucleus, it will lock onto a short sequence called protospacer adjacent motif (PAM) as shown in **Figure 2**.

The PAM is a short DNA sequence (2-6 base pairs) that follows the DNA region targeted for cleavage by the CRISPR system and it is necessary for a Cas9 to make its cut and is generally found 3-4 nucleotides downstream from the cut site.

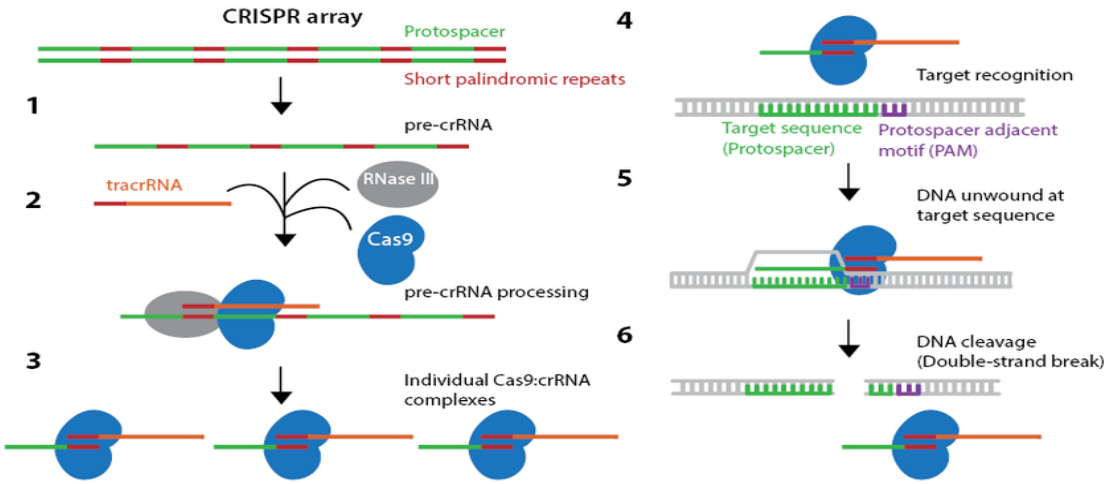


**Figure 2: How CRISPR works**

Although these mutations are random, the researchers sometimes need to be more precise. This can be achieved by replacing a mutated gene with a healthy copy that can be done by inserting another piece of DNA with the desired sequence. This DNA template will pair up with the cut ends after the CRISPR method has made a cut, recombining and replacing the original sequence with the new version.[3]

All of this can be achieved in cultured cells, like stem cells that can give rise to a variety of cell types, or in a fertilized egg, allowing for the development of transgenic animals with targeted mutations. Unlike previous methods, CRISPR can target several genes at once by modifying the crRNA sequences of the gRNA to suit the target location.

These methods are being improved rapidly and will have many applications in basic research, in drug development, agriculture, and, possibly, in the treatment of human patients with genetic diseases.

## b. Programs designed to predict CRISPR cutting specificity

There are two main methods to predict the specificity of CRISPR gRNAs.alignment-base method This method uses conventional algorithms That aligns the gRNAs to a genome to output the off-target sequences. Scoring-based method It takes the off targets that have been scored to select the most specific one by experiments. This method was mainly hypothesis driven which scores the off-targets based on the contribution of specific genome context factors to gRNA specificity and has the problem of outputting false positives. until, the era of machine learning came to create another sub category named learning-based models which uses deep learning to train a pre-scored data set that has the DNA sequence and the epigenetic features. Thus' allowing to create complex models that uses most of the information that we can get experimentally. These methods exhibit better performance than hypothesis driven methods.[1]

### C. DeepCRISPER

DeepCRISPR is our used approach that uses a well-built hybrid deep neural network for model training and prediction of on-target and off-target sites. Using a comprehensive set of genome-wide unlabelled sgRNAs, we used a deep unsupervised representation learning technique to automatically learn the underlying representation of sgRNAs.[4]

# 2. Gantt chart

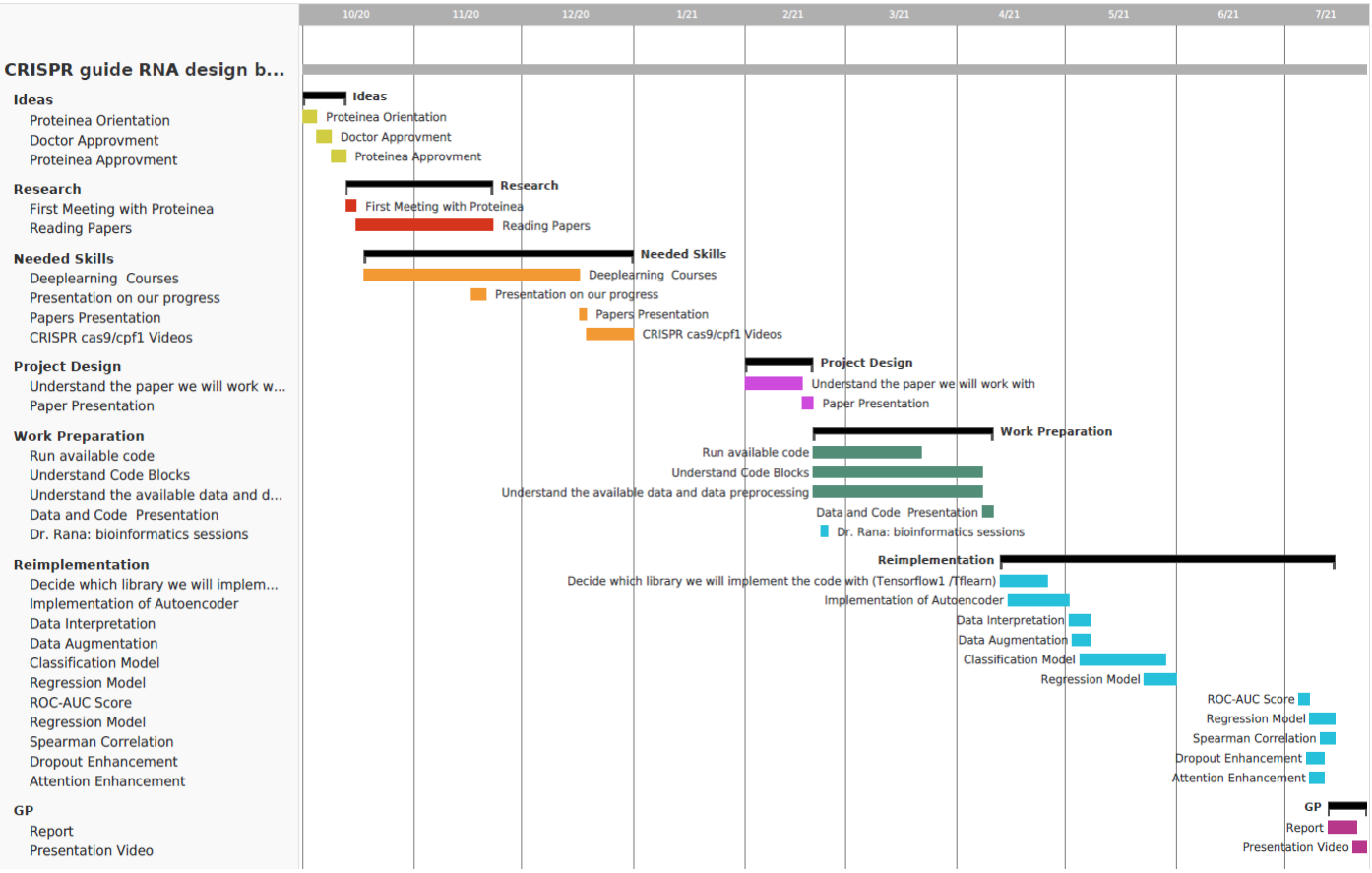


**Figure 3: Gantt Chart for our Project Plan**

# 3. Market Research

## Market Size

Gene Editing Market size is valued at 3.8 billion USD in 2019 and is expected to witness a 14.9% CAGR from 2020 to 2026. Data-driven advances in the field are anticipated to boost the market demand even further. It is expected that the market size of this gene-editing will reach 10.0 billion USD by 2026. The majority of this market is driven by recombinant protein production applications, whose market is evaluated at 2.3 billion USD, and has an 11.2% CAGR from 2020 to 2027. The projection of the recombinant protein production market in 2027 is 7.5 billion USD.[5]

## Competitors

While proteinea is the only Egyptian company interested in recombinant protein production in Eukaryotes, there are many companies that represent competitors to Proteinea internationally. Focusing on the AutoCRISPR project, some of the competitors are shown in **Table 1**. However, it is clear that AutoCRISPR still holds competitive edges over the current competitors.

**Table 1: Represents a comparison between AutoCRISPR software and other available softwares for CRISPR design**

| | Software as a Service | Non-model organism | Protein yield prediction |
|---|---|---|---|
| AutoCRISPR | ✔ | ✔ | ✔ |
| Synthego | ✘ | ✘ | ✘ |
| Inari Agriculture | ✘ | ✔ | ✘ |
| Pairwise Plants | ✘ | ✔ | ✘ |
| Caribou Biosciences | ✘ | ✔ | ✘ |
| Desktop Genetics | ✔ | ✘ | ✘ |

## Customers

AutoCRISPR has two main segments of customers. First, academic labs working on testing CRISPR experimentally requires a load of money and resources which might not be available to every lab especially in emerging markets. Proteinea is in initial talks with the academic labs interested in this product like the German institute, Fraunhofer the second segment is companies working on utilizing non-model organisms as recombinant protein hosts and need ready CRISPR systems to use. Proteinea is also in talks with interested companies like the Spanish company, Algenex.[5]

# 4. Material and Methods

### a. Data

A gene knockout `KO` is a genetic procedure that disables one of an organism's genes ("knocked out" of the organism). Knockouts are used to investigate gene function, most commonly by looking at the effects of gene loss. The difference between the knockout organism and normal individuals is used by researchers to derive conclusions.[6][7]

CRISPR-based gene knockout is widely implemented in various cell types and organisms. CAS9 follows the sgRNA to the desired location then it untwists the DNA and compares it to its target RNA; if the match is successful, i.e., the on-target, cleavage occurs upstream of a PAM however sometimes the cleavage occurs despite the mis-match i.e., off the target.[4][8]

We used a public dataset that contains on-target dataset and off-target dataset. We work on on-target dataset.[4][9]

#### i. On-target data sources

Our on-target dataset includes ~15,000 sgRNAs containing 1071 genes from four different cell lines with redundancy removed and validated known knockout efficacy.

The four different cell lines are **HCT116** that was isolated from male with colonic carcinoma, **HEK293T** that was taken from a kidney an aborted female fetus, **HELA** that was derived from cervical cancer cell, and finally **HL60** that was taken from a human with leukaemia.[10][11][12][13]

### ii. Epigenetic features

DeepCRISPR can be easily extended to other cell types if it similar to our used epigenetic features so we included it in our encoding schema.

These epigenetic features included **CTCF** binding information from the ChIP-Seq assay, **H3K4me3** position information from the ChIP-Seq assay, chromatin-opening information from the **DNase**-Seq assay, and DNA methylation information from the **RRBS** assay, obtained from **ENCODE** (The ENCyclopedia of DNA Elements: Project aims to identify all functional elements in the human genome sequence).[14]

## b. Data Pre-processing

### i. sgRNA encoding with genome and epigenetic features

The nucleotide sequence is represented by four channels in a DNA region, i.e., the A-channel, C-channel, G-channel, and T-channel, and each epigenetic feature is considered as one channel. As a result, each DNA region has an eight-channel representation (**Figure 4**).[4]

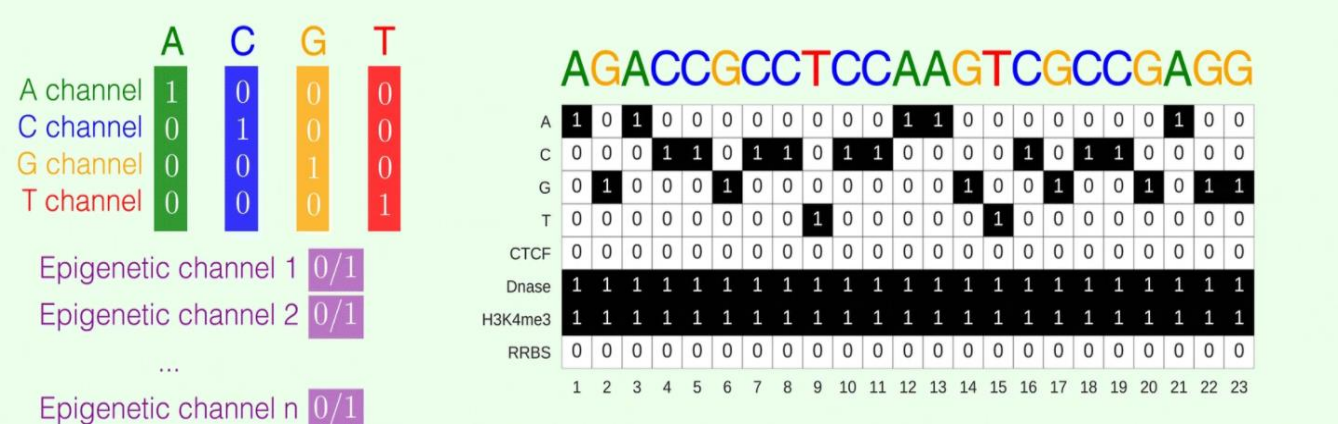


**Figure 4 : sgRNA encoding schema**

### ii. Autoencoder Data

We concatenated the available classification and regression data for the four cell lines to provide ~70,000 data samples to train our autoencoder because the downloaded public data weren`t complete as there weren't 0.68 billion data samples for training autoencoder as mentioned in the paper we worked on.[9]

### iii. Classification and Regression Model Data

Previously, sgRNA efficacy was converted to a binary value using a log-fold change of 1 as the cut-off for classification model and collaborative filtering-based data normalization method was adopted for regression model. **Table 2** provides both models data sources.[9]

Table 2 : sgRNA data for Regression and Classification Model

| Cell line | Classification Data | Regression Data |
|---|---|---|
| HCT116 | ~ 4,200 | ~ 4,200 |
| HEK293T | ~ 2,300 | ~ 4,500 |
| HELA | ~ 8,000 | ~ 8,000 |
| HL60 | ~ 2,000 | ~ 2,000 |
| Total | ~ 16,700 | ~ 18,900 |

### iv. Data Augmentation

To enhance our model performance, we needed an additional data which weren`t available so we did data augmentation by

1. Data Augmented I

   We concatenate the data of the four cell-lines to be in one file only.

2. Data Inverse

   We inverted the data augmented by swapping the start and end of our DNA sequences and epigenetic features separately for the four cell lines to gain more data without losing the meaning of the original dataset so it acts as a new dataset.

3. Data Augmented II

   We concatenate the data inverted `new dataset` to data augmented in the first step so we get a new file contains ~35,000 data for each model.

## c. Deep Learning Model

The Model we are trying to reimplement are the on-target classifier and the on target regressor and the auto encoder which will be used to fine-tune the previous models(**Figure 5**).[4], [9]

### i. Auto encoder Model

This is an unsupervised deep convolutionary denoising neural network (DCDNN)-based autoencoder. It consists of two groups of layers the encoder group and the decoder group. The encoder group consists of 5 layer of Convolution 2d and batch normalization and denoising layer. The decoder Group in the other hand have a nearly same structure but with the expectation of using deconvolution instead of a convolution.[4]

This model is the model that requires the most data. This model trains on the whole augmented dataset to optimize the encoder as much as we could So it can improve the other models when we load its encoder layers into them.

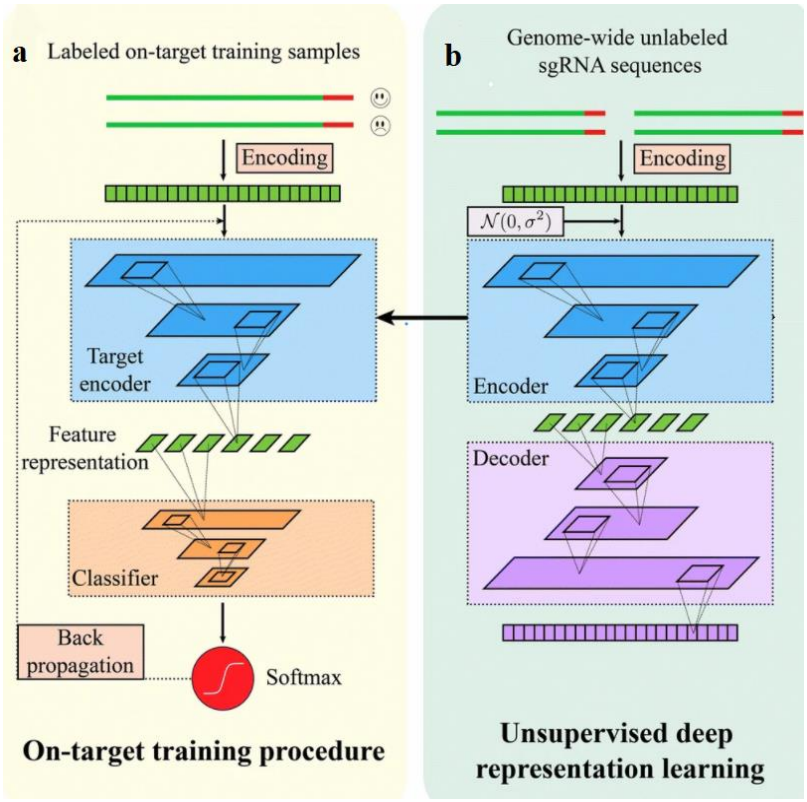


**Figure 5 : a) Training details of DeepCRISPR for sgRNA on-target efficacy prediction. The SoftMax and Identity functions correspond to classification and regression models, respectively. b) Unsupervised deep representation learning based on billions of genome-wide sgRNA sequences.**

## ii. Classifier and Regressor Models

These are the primary models are the classifier model and the regressor model. These models consist – like the autoencoder- from 2 Layer Groups: 1. The encoder layer 2. The classifier or regressor.

Both of these models have the same architecture for the encoder part as the encoder in the auto encoder, since they will load the Encoder layers from the auto encoder.

We used ROC-AUC value to be the metric of our classification model and used spearman correlation to be the metric of our regression model.[4]

### 1. The encoder group:

Its Primary goal is to encode the 23*8 DNA sequence format we got to a 1*256 sequence of numbers that contain important information about the DNA for the classifier or regressor to work on. These layers can be loaded from the auto encoder and over tuned on this model again. [4]

## 2. The Classifier or Regressor group:

This is the model's last layer that will take the 256-information outputted by the encoder layers and output the classification or regression output. These layers can differ but until now they have nearly identical layer configuration with expectation of the last layer.[4], [9]

**Table 3: Model Layers**

| Input Data of size: 1*23*8 | | | | | |
|---|---|---|---|---|---|
| Auto-Encoder model | | Classifier model | | Regressor model | |
| Encoder group | | Encoder group | | Encoder group | |
| layer | output Size | layer | output Size | layer | output Size |
| convolution | 1*23*32 | convolution | 1*23*32 | convolution | 1*23*32 |
| Batch normalization | | | | | |
| Adding normal distributed noise | | | | | |
| convolution | 1*12*64 | convolution | 1*12*64 | convolution | 1*12*64 |
| Batch normalization | | | | | |
| Adding normal distributed noise | | | | | |
| convolution | 1*12*64 | convolution | 1*12*64 | convolution | 1*12*64 |
| Batch normalization | | | | | |
| Adding normal distributed noise | | | | | |
| convolution | 1*6*256 | convolution | 1*6*256 | convolution | 1*6*256 |
| Batch normalization | | | | | |
| Adding normal distributed noise | | | | | |
| convolution | 1*6*256 | convolution | 1*6*256 | convolution | 1*6*256 |
| Batch normalization | | | | | |
| Adding normal distributed noise | | | | | |
| Decoder group | | Classifier group | | Regressor group | |
| De-convolution | 1*12*256 | convolution | 1*3*512 | convolution | 1*3*512 |
| Batch normalization | | | | | |
| De-convolution | 1*12*256 | convolution | 1*3*512 | convolution | 1*3*512 |
| Batch normalization | | | | | |
| De-convolution | 1*24*64 | convolution | 1*1*1024 | convolution | 1*1*1024 |
| Batch normalization | | SoftMax | 1*1*2 | squeeze | 1 |
| De-convolution | 1*24*64 | squeeze | 1 | | |
| Batch normalization | | | | | |
| De-convolution | 1*23*23 | | | | |
| Batch normalization | | | | | |
| De-convolution | 1*23*8 | | | | |

**Table 3**: This Table provide layers structures. on the left the layers structure for auto encoder. on the middle the layers structures for classifier. On the right the layers structures for regressor. There is a RELU activation layer before every convolution (expect the very first one).

## d. Model Enhancement

### i. Dropout

Dropout can be used to have a large number of different network architectures by randomly dropping out nodes during training. This offers a computational, cheap, remarkable and effective regularization method to reduce overfitting. We used a keep_prob = 0.7, the keep_prob is the percentage of nodes kept and dropped out which reduced the overfitting of our model with about 85%. But with slight decrease on the AUC-ROC value. We added a layer after every batch normalization and before the activation layer.[15]

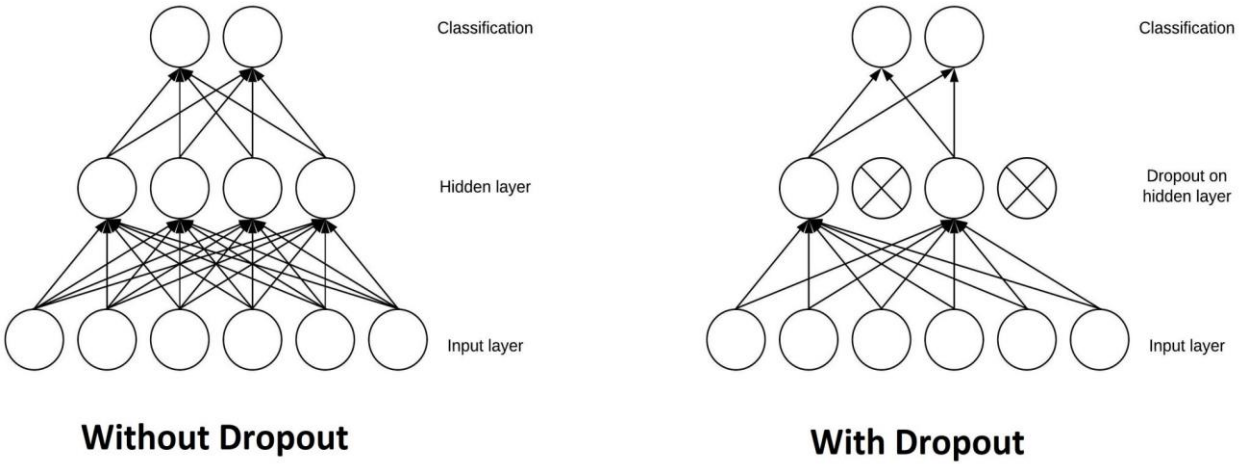


**Figure 6: with and without Dropout**

### ii. LSTM

It is an abbreviation for Long Short-Term Memory. LSTM can choose what to save in the memory and what to leave through a sigmoid function that is summed together at the end.

We tried to use LSTM to enhance the AUC-ROC accuracy of or model but we failed to implement it because our data is 4D data while the library of LSTM uses a 3D data and when we try to reshape our input data and the reshape the output data the AUC-ROC accuracy of our model decreases to 48%.[16]

### iii. Attention

Attention in deep learning is used as a vector of important weights to predict or infer an element, we estimate using the attention vector how strongly it is correlated to other elements and take the sum of their values weighted by the attention vector as the approximation of the target. It estimates the best keep_prob value after calculating it through training.

The attention mechanism we used reduced the overfitting in our model but doesn't enhance the AUC-ROC metric of the model. We added a layer after every batch normalization and before the activation layer.[17], [18]

### e. Coding Methods

We used python (3.6) and used tflearn library (0.5.0).we used tflearn because it is a high level API for tensor flow 1. They have the same graph structure and behaviour which is important for reimplementation. training was done using ADAM optimizer with learning rate 0.001 and beta1 of 0.9, beta2 of 0.999 and epsilon of 1e-8.[19]

As discussed, before we have 3 building blocks on out models:

#### i. Convolution

Convolution layers are good in extract different features (motifs) at different scales from the DNA than conventional scientific methods that's why every layer starts with a convolution. Each convolution layer was a two-dimensional convolution layer with a kernel shape of [1,3] with a different number of channels in every layer. We used the implementation used in tflearn.[20]

#### ii. Batch normalization

Batch normalization work on the idea of gaussian normalization. The normalization here does not occur once like normal normalization in pre-processing, here we normalize the mini batch trained in this layer to have a zero mean and unit variance. We use it after convolution directly because convolution have the property of changing the mean and variance of the input data. Using batch normalization enable us to care less about weights initialization and use higher learning rate. We used the implementation used in tflearn.[21]

#### iii. Denoising

It is the adding of a gaussian random noise to make the CNN model robust to noise in the test data and in the huge sgRNA dataset. We used the gaussian implementation used in tensorflow 1.

#### iv. Attention

As explained in the enhancement section.

Unfortunately, the tflearn library did not have a default implementation. We tried to use the attention from the Keras library but it works on a different frame work (tensor flow 2) so it did not work. We tried used the code in this paper [22] that combines both attention and convolution but the code was not complete and did not has some important features. Then we tried to get a RNN attention code and modify so it can accept the 4D input data.[18]

#### v. Dropout

As explained in the enhancement section. We used the implementation used in tflearn.

## 5. Results and Discussion

For comprehensive and objective comparisons of DeepCRISPR with varying ROC-AUC/Spearman Correlation scores, six separate testing scenarios were carefully created. These comparisons show that:

1. The deep learning models (without unsupervised pre-training) are superior to shallow learning models
2. The unsupervised pre-training strategy boosts model performance

3. The data augmentation further improves model performance and model robustness
4. DeepCRISPR generalized generally well in new cell types for sgRNA on-target knockout efficacy prediction

## 1. Testing Scenarios

For classification schema, we evaluated the performance using the values from the area under the receiver operating characteristic (ROC) curve (AUC).

For Regression Schema, we evaluated the performance using Spearman correlation method, which has previously been used in other investigations.[4]

### a. Testing scenario 1— Classification schema

In this test, we used ~15,000 sgRNA with known knockout efficacies from 4 cell lines. We split the data into two parts: 20% used as testing sets and 80% for model training and parameter tuning. The deep CNN-based classification model (denoted as "CNN" (**Figure 7**)) was trained and tested using independent test data for the four cell lines, without unsupervised pre-training `Autoencoder` or data augmentation.

The results showed that DeepCRISPR achieved an overall ROC-AUC of 0.71178 (**Figure 8**) compared against the previous study which is 0.796.

### b. Testing scenario 2— Classification schema

In this test, we built our model with unsupervised pre-training on ~35,000 labelled sgRNAs (denoted as "pt CNN"(**Figure 7**)) `while the paper trained the autoencoder on ~0.68 billion sgRNA`. The same training and testing data as for scenario 1 were used.

The results showed that DeepCRISPR achieved an overall ROC-AUC of 0.713 (**Figure 8**) compared against the previous study which is 0.836.

### c. Testing scenario 3— Classification schema

We used pre-training-based CNN plus data augmentation to build our final DeepCRISPR model (denoted as "pt + aug CNN" (**Figure 7**)). The training data was augmented `~70,000 sgRNA`, but the testing data remained the same as in testing scenarios 1 and 2.

The results showed that DeepCRISPR achieved an overall ROC-AUC of 0.70913 (**Figure 8**) compared against the previous study which is 0.857.

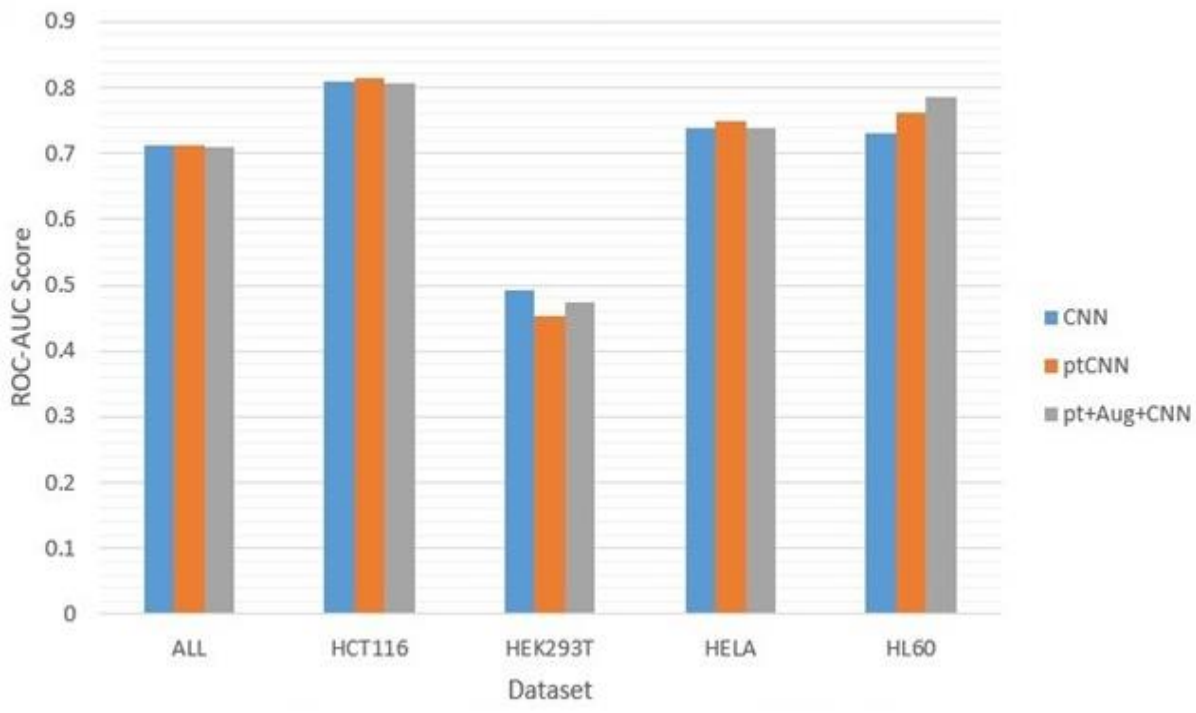


**Figure 7: Comparison of sgRNA on-target efficacy predictions in a classification schema for various datasets, i.e., hct116 cell line, hek293t cell line, hela cell line, hl60 cell line, and the overall testing dataset using ROC-AUC Score for 1st three scenarios.**

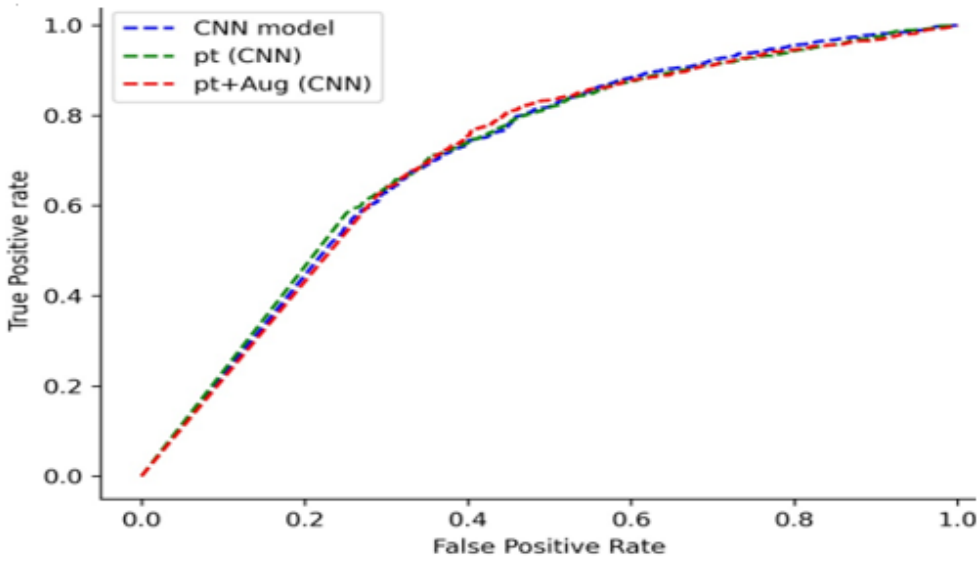


**Figure 8: Benchmarking sgRNA on-target efficacy predictions on all testing data with ROC curve for first three classification scenarios**

## d. Testing scenario 4 — Classification schema

We investigated DeepCRISPR's generalization property in new cell types in this scenario. Twenty percent of the data from each cell type was used as independent testing sets for the original ~15,000 sgRNAs with known knockout efficacies from four cell types. The remaining 80% of the data from various cell types was used to enhance the training data, which was identical to that used in testing scenario 3.

Then, using the training data combined from three cell types and testing on the leave one cell type out independent dataset, our model was trained four times, each time using the training data combined from three cell types and testing on the leave one cell type out independent dataset.

DeepCRISPR's performance on four cell types achieved an average ROC-AUC of 0.64166 (**Figure 9**) (**Figure 10**) in this scenario compared against the previous study which is 0.722.

**Table 4: Performance of Different cell lines in Scenario 4-Classification Schema**

| Cell Line | HCT116 | HEK293T | HELA | HL60 |
|---|---|---|---|---|
| **Testing scenario 4** | 0.78115 | 0.492385 | 0.736144 | 0.636315 |

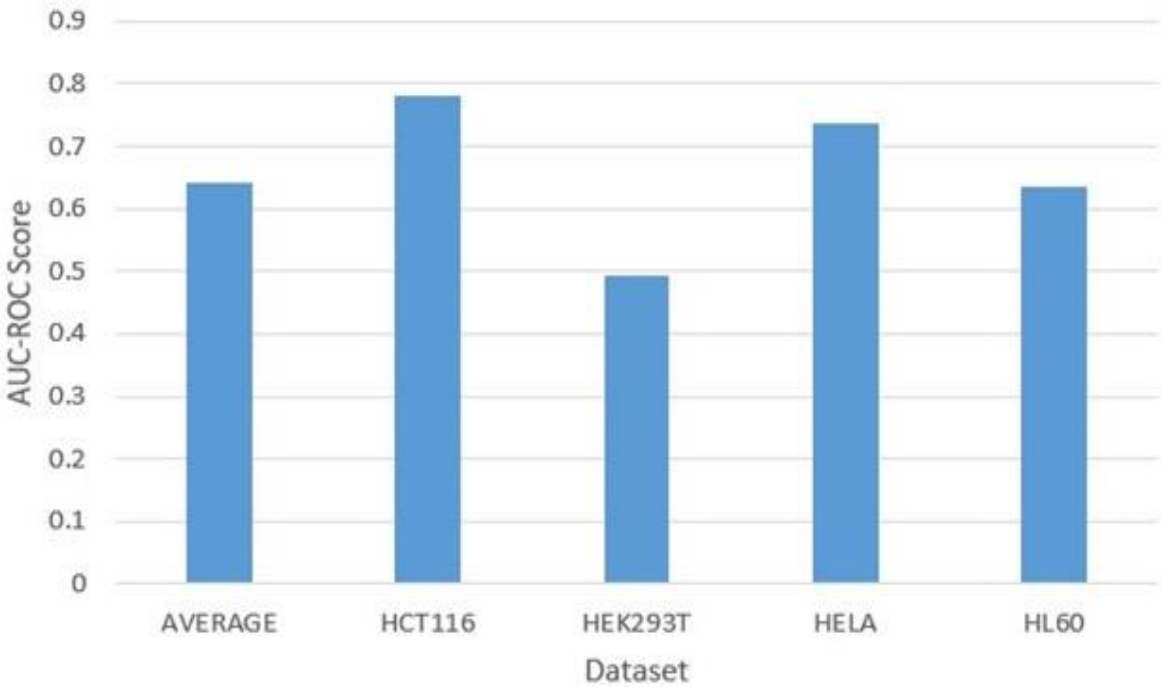


**Figure 9: Leave cell type out comparison of sgRNA on-target efficacy prediction in a classification schema**

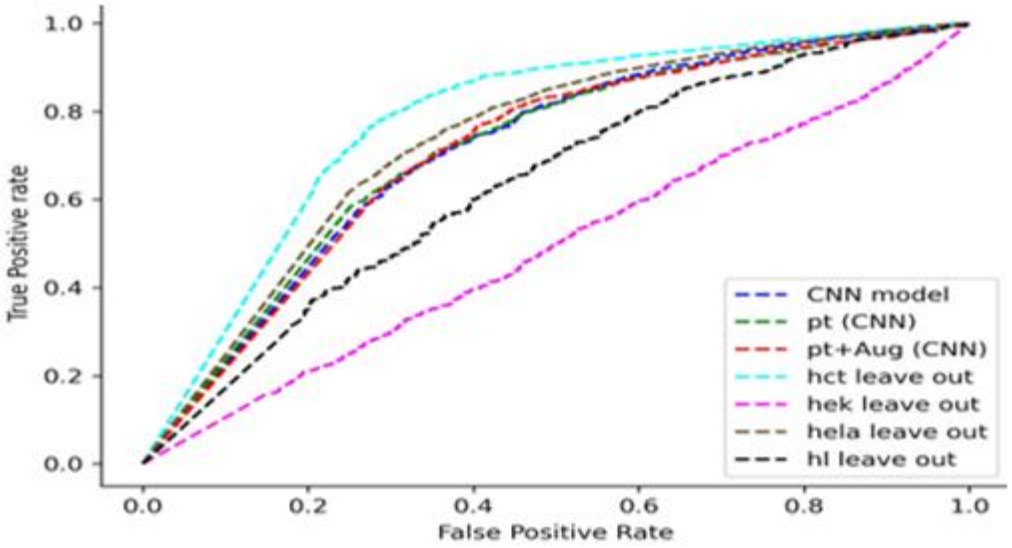


**Figure 10: Benchmarking sgRNA on-target efficacy predictions with ROC curve for all four classification scenarios**

## e. Testing scenario 1— Regression schema

In this test, we used the sgRNA knockout efficacies regression data to train DeepCRISPR in a regression way. The comparison was carried out in the same way as in testing scenario 3, with the exception that the model was trained using regression.

DeepCRISPR's performance on four cell types achieved a 4 different spearman correlation (**Table 5**) and a spearman correlation of the whole types was 0.44475 (**Figure 11**) compared against the previous study which is 0.6.

**Table 5: Performance of Different cell lines in Scenario 1-Regression Schema**

| Cell Line | HCT116 | HEK293T | HELA | HL60 |
|---|---|---|---|---|
| **Testing scenario 1** | 0.52313 | 0.54208 | 0.42448 | 0.33769 |

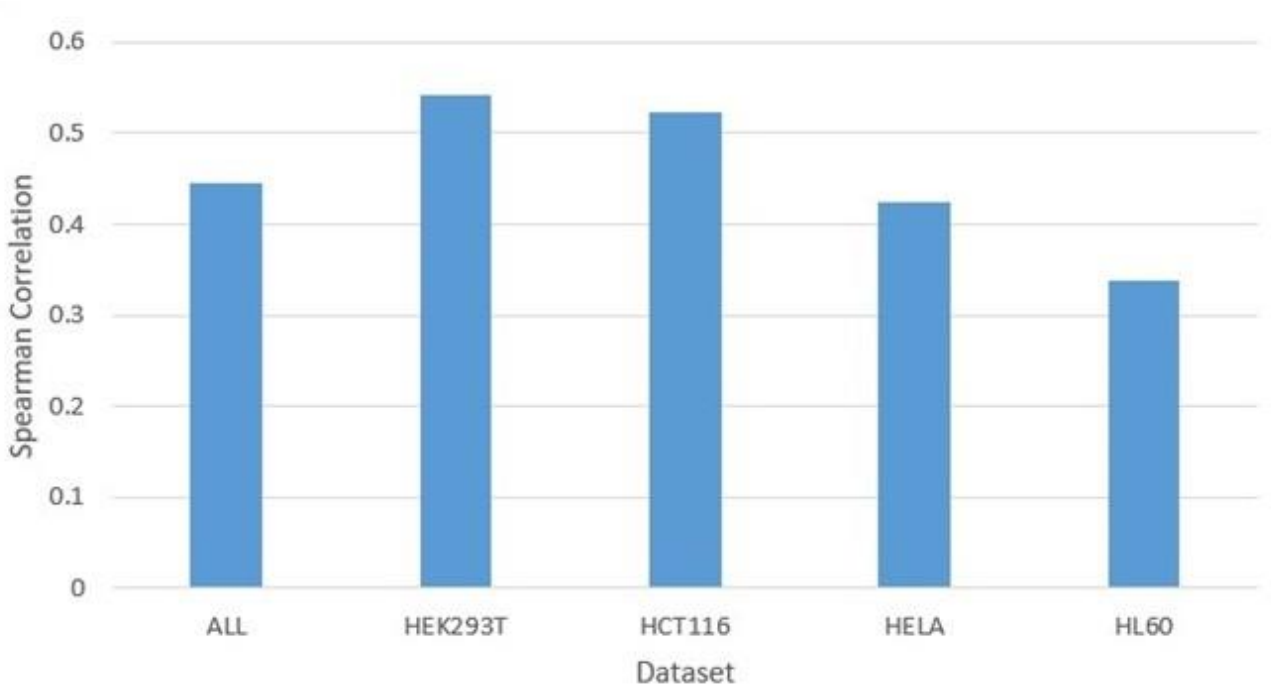


**Figure 11: Comparison of sgRNA on-target efficacy predictions in a regression schema for various datasets, i.e., hct116 cell line, hek293t cell line, hela cell line, hl60 cell line, and the overall testing dataset using Spearman Correlation**

## f. Testing scenario 2— Regression schema

We tested the regression-based DeepCRISPR in a leave one cell type out way to investigate its generalization ability in new cell types, similar to testing scenario 4.

DeepCRISPR's performance on four cell types achieved a 4 different spearman correlation (**Table 6**) and an average Spearman correlation of 0.24897 (**Figure 12**) compared against the previous study which is 0.4.

**Table 6: Performance of Different cell lines in Scenario 2-Regression Schema**

| Cell Line | HCT116 | HEK293T | HELA | HL60 |
|---|---|---|---|---|
| **Testing scenario 2** | 0.58414 | -0.040116 | 0.41214 | 0.15531 |

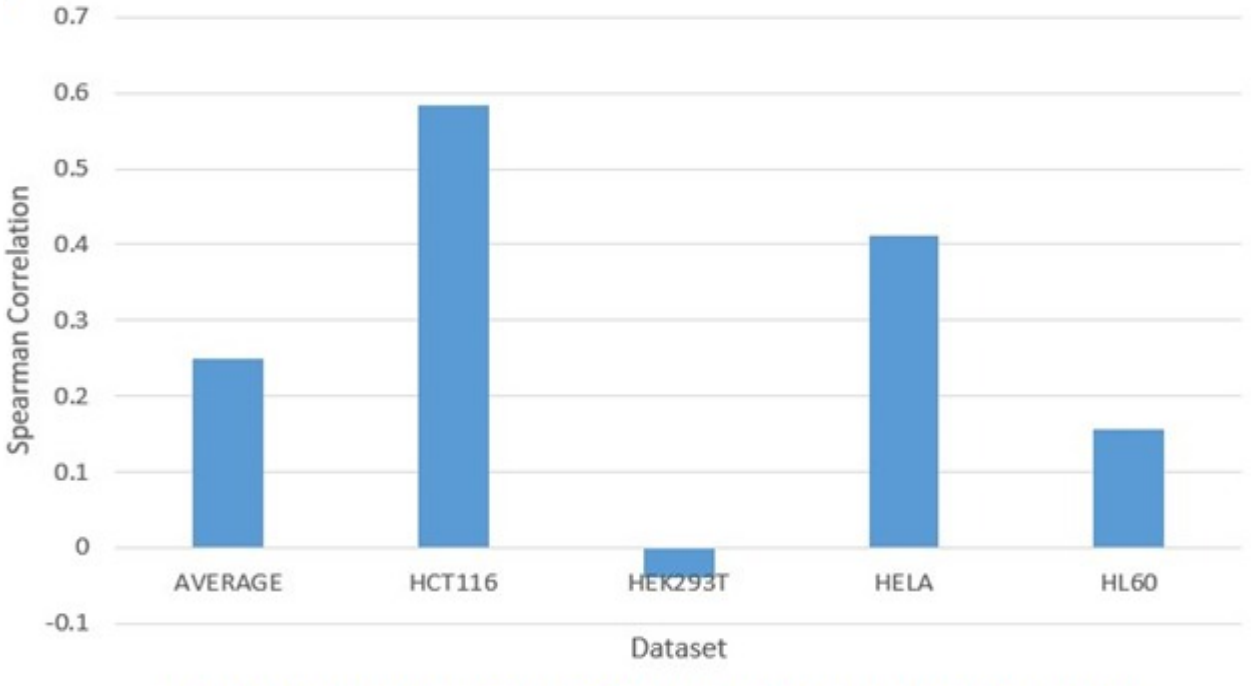


**Figure 12: Leave cell type out comparison of sgRNA on-target efficacy prediction in a regression schema.**

As a result of the public data not being totally available, there is a discrepancy of around 10% in all our results comparing to the original paper results.

## 2. Model Enhancement

### a. Dropout

We tried to use the dropout technique to enhance our model, as a result it decreases the overfitting but on the other side it reduces the accuracy of our model. So we decided not to use it because we need to achieve the accuracy in compare to the paper we are implementing.

As shown in (**Figure 13**) and (**Figure 14**) the accuracy decreases nearly by a percentage of 3%.

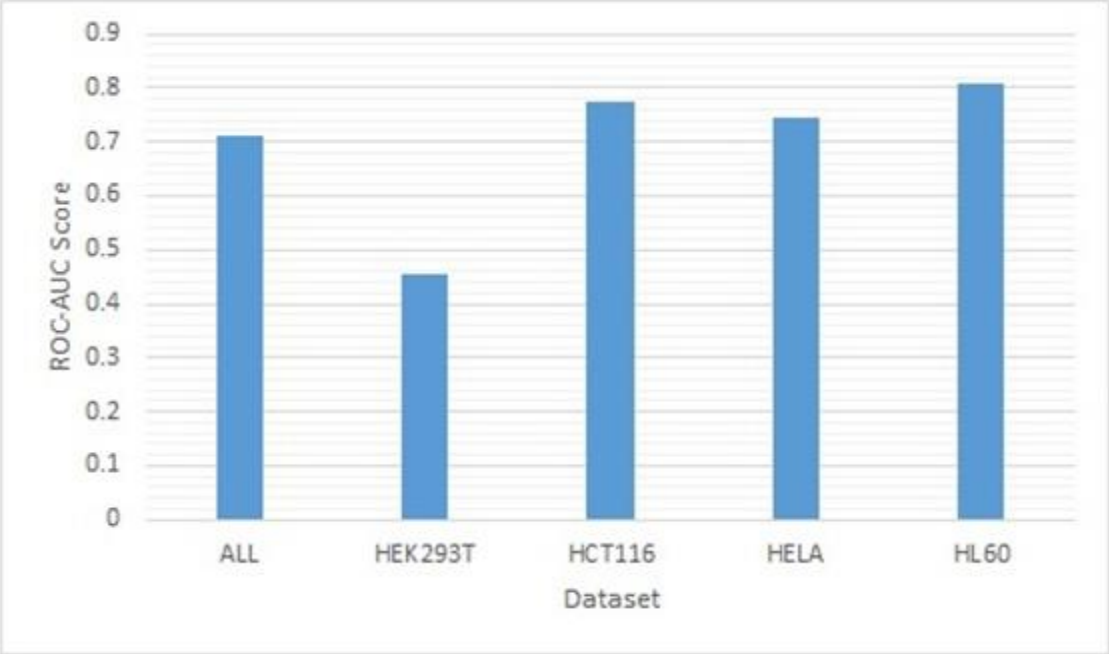


**Figure 13: Classification Model Enhancement by using Dropout Technique**


**Figure 14: Regression Model Enhancement by using Dropout Technique**

## b. Attention

We also tried to enhance our model using attention but we got the same results we got from dropout as the overfitting decreases but the accuracy also decreases by nearly 5%. (**Figure 15**) and (**Figure 16**)

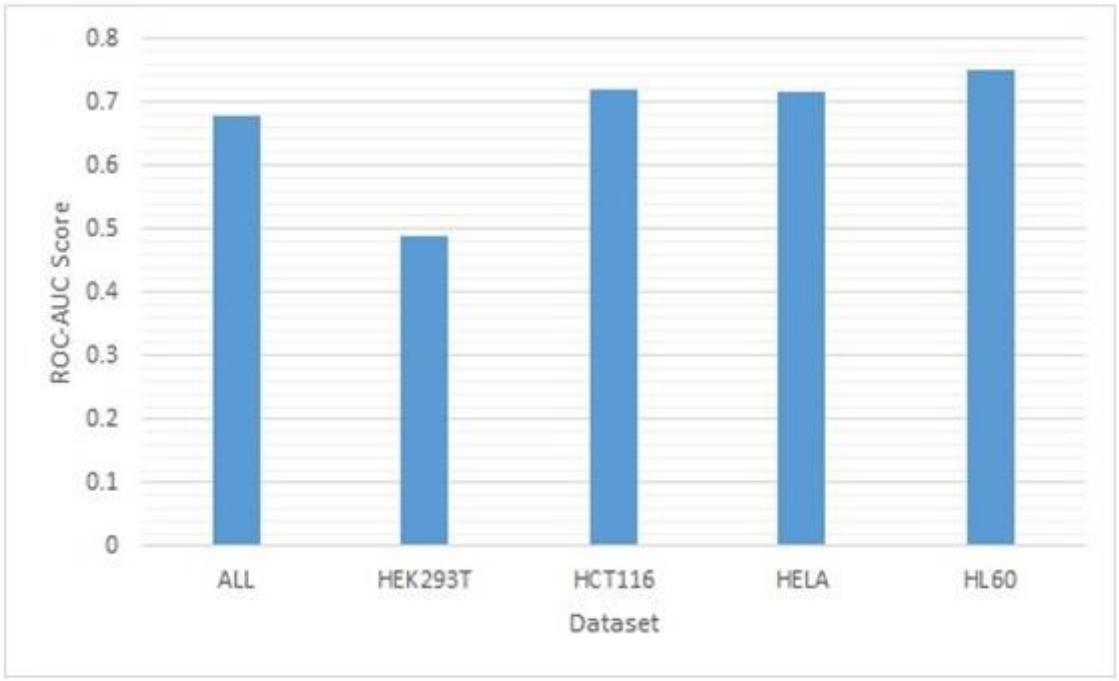


**Figure 15: Classification Model Enhancement by using Attention Technique**

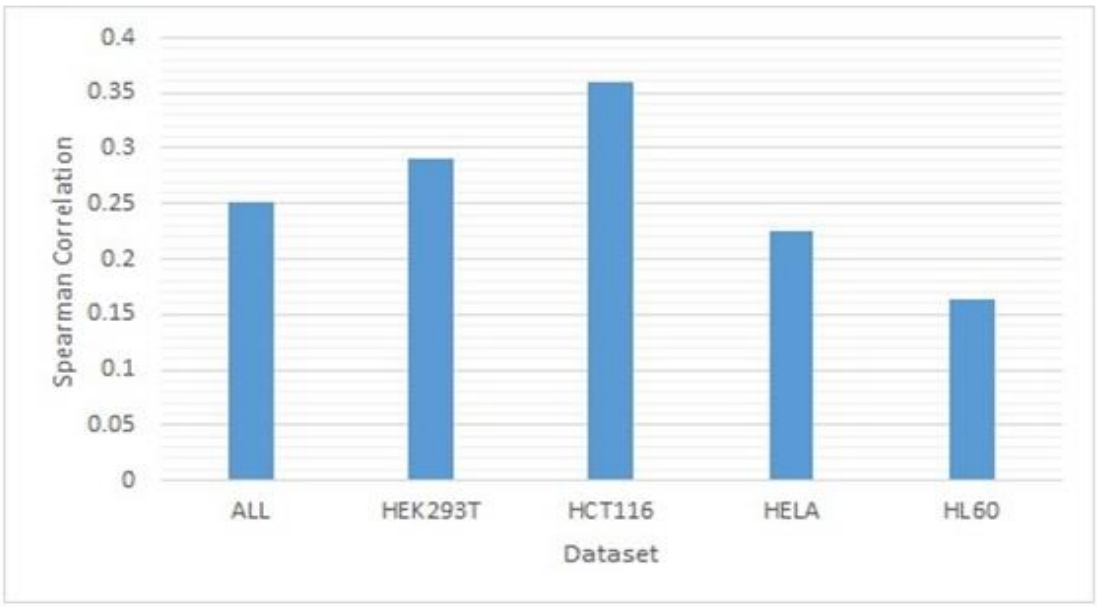


**Figure 16: Regression Model Enhancement by using Attention Technique**

# 6. Conclusion and Future work

DeepCRISPR is a computational model for predicting CRISPR sgRNA on-target knockout efficacy that is efficient and adaptable. Importantly. DeepCRISPR uses data to automate feature recognition for sgRNA design, allowing for easier interpretation and better CRISPR on-target design. our results show how crucial it is to have a large dataset and how it should be the norm to publish correct data publicly to improve such models and to improve our use of it.[4]

A number of future improvements can be done:

1. We have only created a CNN-based deep neural network model that is rather simple and concise. In the future, a lot more complex and modern deep learning architectures will be possible, and these models will be expected to improve our existing prediction performance.

2. The amount of available data for us was too small around 35,000, which provides a challenge for training a routine deep learning model. A common concern with deep models is that they can overfit the training data. Therefore; we need to increase the data available to get better training and therefore better results.

3. We have only worked on the on target data so in the future, the off target data can be used to achieve more accurate and precise results.

4. As a result of increasing the data in the future they can use the enhancement methods we have tried to use so that they can reduce the overfitting and therefore get a more precise model without the decrease of the accuracy of both classification and regression model.

5. The use of modern frameworks like pyTorch or Tensorflow 2 will make the community be able to use the full set of tools the community has to offer.

# 7. Availability of Data and Code

**Github source**

# 8. References


[1] G. Liu, Y. Zhang, and T. Zhang, "Computational approaches for effective CRISPR guide RNA design and evaluation," *Comput. Struct. Biotechnol. J.*, vol. 18, pp. 35–44, 2020, doi: 10.1016/j.csbj.2019.11.006.

[2] "What-Is-Crispr-Cas9 @ Www.Yourgenome.Org." [Online]. Available: https://www.yourgenome.org/facts/what-is-crispr-cas9.

[3] MedlinePlus Genetics, "Genomic Research Center," 2020, [Online]. Available: https://medlineplus.gov/genetics/understanding/genomicresearch/snp/.

[4] G. Chuai *et al.*, "DeepCRISPR: optimized CRISPR guide RNA design by deep learning," *Genome Biol.*, vol. 19, no. 1, p. 80, 2018, doi: 10.1186/s13059-018-1459-4.

[5] "index @ www.proteinea.com." [Online]. Available: https://www.proteinea.com/.

[6] "Gene-Knockout @ Www.Sciencedirect.Com." [Online]. Available: https://www.sciencedirect.com/topics/biochemistry-genetics-and-molecular-biology/gene-knockout.

[7] "4acf0d3a4b65f9a5912a5f679fa8523cd4aeeb2b @ www.ncbi.nlm.nih.gov." [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7018052/.

[8] "4875932 @ academic.oup.com." [Online]. Available: https://academic.oup.com/gigascience/article/7/3/giy011/4875932.

[9] "DeepCRISPR @ github.com." [Online]. Available: https://github.com/bm2-lab/DeepCRISPR.

[10] "b6ef87937d7c7545f732d8b564528cbfcbe318d8 @ imanislife.com." [Online]. Available: https://imanislife.com/collections/cell-lines/hct116-cells/.

[11] "hela-cell-line @ embryo.asu.edu." [Online]. Available: https://embryo.asu.edu/pages/hela-cell-line.

[12] P. H. England, "Cell line profile- SH-SY5Y," *Public Heal. Engl.*, vol. 2780, no. 84113001, pp. 1–2, 2016, [Online]. Available: https://www.phe-culturecollections.org.uk/media/114601/sh-sy5y-cell-line-profile.pdf.

[13] "cb_12022001 @ www.sigmaaldrich.com." [Online]. Available: https://www.sigmaaldrich.com/EG/en/product/sigma/cb_12022001.

[14] Stanford University, "Index @ Www.Encodeproject.Org." 2017, [Online]. Available: https://www.encodeproject.org/.

[15] "b78b7a9426f5a39d39d2ffcd62b5936fb82f729a @ machinelearningmastery.com." [Online]. Available: https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/.

[16] "1984908ab03cb89bf855af9f13278f5c1b8cae14 @ machinelearningmastery.com." [Online]. Available: https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/.

[17] "Attention-Attention @ Lilianweng.Github.Io." [Online]. Available: https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html#summary.

[18] "attention-in-rnns-321fbcd64f05 @ medium.datadriveninvestor.com." [Online]. Available: https://medium.datadriveninvestor.com/attention-in-rnns-321fbcd64f05.

[19] G. Chuai *et al.*, "DeepCRISPR: Supplementary notes of the detailed model architecture," pp. 5–7.

[20] "69a16178474053aafd847bf3b2c6ca9e3c142c6f @ machinelearningmastery.com." [Online]. Available: https://machinelearningmastery.com/padding-and-stride-for-convolutional-neural-networks/.

[21] F. Schilling, "The Effect of Batch Normalization on Deep Convolutional Neural Networks," p. 113, 2016, [Online]. Available: http://kth.diva-portal.org/smash/record.jsf?pid=diva2%3A955562&dswid=-5716.

[22] "1904 @ Arxiv.Org." [Online]. Available: https://arxiv.org/abs/1904.03516.