NOTTINGHAM
TRENT UNIVERSITY

Department of Computing And Technology

Learning Models To Predict mRNA Base Degradation Rates

by

Dayeeta Das

in

2021

Supervised by

Dr Joao De Castro Cardoso Ferreira

Project report in part fulfilment

Of the requirements for the degree of

Bachelor of Science with Honours

in

Computer Science

Module Code: COMP30151
Module Name: Project 202021 Full Year
NTU ID: N0830182
Name: Dayeeta Das

I, Dayeeta Das confirm that I am the sole author of this report. I grant my approval to Nottingham Trent University for sharing this report with other individuals or institutions for the purpose of scholarly research.

I am fully aware of the University's rules of plagiarism and collusion and if I am found guilty of breaking any of these rules then that will be treated as an Academic Misconduct.

I understand that it is completely my responsibility to ensure that I submit the complete coursework within the deadline and any other submissions after the deadline will be disregarded.

I understand that these rules apply to me even on the event of system failure.

Dayeeta Das

Dayeeta Das

Abstract

The surge of the pandemic has caused the world to go into an emergency situation. Scientists and researchers are trying their best to design a vaccine which is going to work efficiently, act fast, be cost-effective and can be easily produced over a large scale. mRNA vaccines are leading the race as the potential solution to this situation. Successful clinical trials of some mRNA vaccines have created a buzz in the medical world. Unfortunately they have a major drawback of degrading spontaneously. This project aims to provide a collaborative framework enabling shareholders to design and create models for predicting the degradation rates of the mRNA molecules and design a web interface for organizing the results. The successful completion of this project would enable researchers to identify the part of the RNA from which is the most prone to degradation and in turn enable them to design an effective vaccine.

Acknowledgements

I would like to express my gratitude to my project supervisor, Dr. Joao De Castro Cardoso Ferreira for providing me with his able guidance and valuable feedback throughout the course of the project. I would also like to thank him for always being supportive and encouraging. I would like to thank my mother and all my friends who have helped me to walk through these uncertain times and have provided immense mental support for the successful completion of my dissertation.

Contents

List of Figures and Tables

List of Figures *Page No.*

List of Tables *Page No.*

(15553 words only)

Chapter 1 – Introduction

COVID-19 has caused havoc worldwide. This pandemic initiated by the SARS-CoV-2 virus has been spreading like wildfire. While people across the globe are still trying to transpire from the aftermath caused by the first wave of the pandemic, the second wave has come around. The World Health Organization (WHO) has reported a record 139,501,934 confirmed cases including 2,992,193 deaths globally as of 14th April 2021 (WHO, 2021). The numbers reflect on the brutality of the disease. The stakes are high. This has set the motivation for developing a vaccine which is not only going to be effective but is also going to act fast and can be easily produced on a large scale.

Researchers and scientists have concluded that mRNA vaccines can be the fastest remedy for this ailment (www.pbs.org, n.d.). However, they have also discovered a fly in the ointment. mRNA has a tendency to spontaneously degrade. A single cut can render the mRNA vaccine useless (kaggle.com, n.d.). However, scientists are yet to confirm which part of the mRNA is deemed to degrade the most (kaggle.com, n.d.). Scientists at the Stanford University have spear-headed the research work on designing effective mRNA vaccines (Conger, 2020). The research team at Stanford led by Dr. Rhiju Das have urged data scientists and programmers to come together and design rules and develop models for predicting the likely degradation rates at each base of an RNA molecule (Conger, 2020). The success of mRNA vaccines can turn out to be a revolution in winning the fight against COVID-19.

1.1. Aim

This project is concerned with the development of a collaborative framework enabling shareholders to design and create Machine Learning and Deep Learning Models that will predict the degradation rates of RNA bases based on five conditions/target columns (reactivity, deg_Mg_pH10, deg_Mg_50C, deg_pH10 and deg_50C). The description of the dataset used for creating these models will be provided in the later chapters. The project aims to use three different modelling techniques with two different evaluation functions for predicting the results. The project also aims to create an interface for organizing the results and data visualizations that were generated during the prediction of the bases.

1.2. MoSCoW Analysis

1.2.1. Must Haves

- The project must have a learning model that predicts the degradation of RNA bases.
- The model created in the project must have a high accuracy score when tested on the test dataset.

1.2.2. Should Haves

- The project should have a consistent coding style throughout depicting 'data cleaning' and 'data pruning'.
- The learning models should have graphs that depict the frequency and relationships of the different features present within the dataset.
- The project should have evidence of Feature Engineering.
- The project should have different ML and DL models for predicting the RNA degradation rates.
- The project should have an output log.

1.2.3. Could Haves

- The project could have an UI that would allow the programmers to browse through the different models created and use them for predicting RNA degradation rates and get an accuracy score.
- The project could have LSTM and GRU for prediction purpose.

1.2.4. Would Haves

- Converted version of the code in Julia.
- System for generating dataset from an RNA base structure created during the Eterna game.
- System for integrating the Eterna gaming platform with the learning model.
- UI that would enable the Eterna gamers to predict the degradation rates of RNA bases.

1.3. Objectives

The SMART objectives which were defined for meeting the requirements of this project has been summarised in the form of a table in the A.1 section of the Appendix.

1.4. Tasks and Deliverables

In order to meet the defined SMART objectives, certain tasks were identified throughout the project some of which had certain deliverables attacked with the. All of the objectives defined within the project were time-bound. Hence, each task was associated with a specific deadline for it's completion which led towards the successful completion of the entire project. The tasks and deliverables for this project has been summarised in form of a table in section A.2 of the Appendix and a Gantt Chart has been presented on section A.3 to represent the duration of each task throughout the course of the project.

1.5. Structure of the Report

This section provides information on how the entire report is structured.

Chapter 2: This chapter contains the review of different Literature which focussed on the current buzz of mRNA vaccines, their advantages and disadvantages and effects of using them in clinical trials. It also provides information on the different models created for predicting the degradation rate of the mRNA bases. This chapter also contains a feature comparison table for comparing and contrasting the aims and objectives and findings of the different research papers used for the Literature Review and identifies the research gaps in each of the papers.

Chapter 3: This chapter provides information on the research gaps identified from Chapter 2 and provides a proposed solution to overcome those limitations.

Chapter 4: This chapter gives a detailed account on the different resources and techniques that were used for implementing the different DL models – LightGBM, WaveNet and GRU + LSTM and also the web interface.

Chapter 5: Chapter 5 gives a detailed account of the different test plans that were used for detecting the success of each unit of the three models and the web interface created. It also provides contingency plans that provides information on how unexpected outcomes from tests could be dealt with.

Chapter 6: This chapter provides a summary of the entire project and concludes the report by reflecting on the successful completion of the project. It also gives an account on the improvements that can be used for developing the project in the future. This chapter covers all the LSEPI issues concerned with the project. It also provides a synoptic assessment reflecting on the lessons learned during the course of the project.

Chapter 2 – Context

Messenger RNA in short mRNA leads the race as the most effective vaccine under mass production for fighting the SARS-CoV-2 virus. However, this is not the first time that mRNA has created a buzz. mRNA has brought about a major innovation in the field of medicine over the past decade (Pardi et al., 2018). The major success of this vaccine caters to the fact that it works efficiently and can be produced easily over a large-scale (ServickDec. 16, 2020 and Pm, 2020). Moreover, mRNA is a non-infectious and non-integrating platform with no potential risk of infection or insertional mutagenesis (Pardi et al., 2018). The recent success of the two COVID-19 vaccines in clinical trials has marked the triumph for a previously unproven medical technology (ServickDec. 16, 2020 and Pm, 2020). In China, 5 vaccines are already under phase II clinical trials (Yang et al., 2020). Recently (Zhu et al., 2020), published the first inspiring clinical result of vaccine in human. mRNA vaccines differ from traditional DNA vaccines in a number of ways. Some of the most prominent features would be that mRNA vaccines do not need to enter the nucleus nor do they risk being integrated into the DNA, they are directly translated into protein antigens. Also, mRNA vaccines require only 1/1000 the dose of DNA vaccines and do not need any special delivery devices (Rachlin and Watson, 2017). Apart from the advantages , mRNA comes with some major drawbacks which needs to be addressed. Studies carried out by (Steele et al., 2020) has shown that a major limitation of mRNA is the inherent chemical instability of RNA. Phase III chemical trials carried out by Moderna, Pfizer-BioNTech vaccines reported promising efficiency rates (~95%) but these trials held unacceptable aftereffects as well (Crommelin et al., 2020). Researchers have observed that RNA molecules have the tendency to spontaneously degrade. A single cut can render mRNA vaccine useless. Under these conditions mRNA vaccines needs to be shipped under extreme refrigeration (kaggle.com, n.d.). This problem led Dr. Rhiju Das, a biochemist at the University of Stanford to launch the OpenVaccine Challenge on Kaggle which aims towards predicting the degradation rates of mRNA bases (Conger, 2020) which has motivated this project.

2.1. Evaluation of existing solutions and primary research

Analysis of models previously created (datasets and software used, technical analysis and results)

All the Deep Learning and Machine Learning models designed for predicting the mRNA degradation rates used the Eterna dataset provided on the Kaggle website (kaggle.com, n.d.). However, certain exceptions like (Coursera, n.d.) have used a smaller dataset for training the learning model.

All of the previously created models like that of(shadowburning, 2020; gagankarora, 2020; fernandoramacciotti, 2020; vbmokin, 2020) have used Python as the primary software and TensorFlow and Keras libraries for building the learning model. Also, the choice of environment has been either Jupyter Notebook (shadowburning, 2020) or Google Colaboratory (fernandoramacciotti, 2020). Some code shoed evidences of other software like ARNIE (Vandewiele, 2020) for developing Base Pair Probability Matrices (BPPS) for a particular RNA sequence. Other studies like that of (fernandoramacciotti, 2020) has also used BPPS. Advantage of creating a BPPS is that, a potential structure could be predicted from the sequence and the loop type could be inferred for each RNA base in a sequence (Vandewiele, 2020).

Studies like that of (Singhal, n.d.; shadowburning, 2020; gagankarora, 2020; vbmokin, 2020; ruko, 2020) and (anzhemeng, 2020) have used a combination of two RNNs- LSTM and GRU for building the learning model. While models like that of (fernandoramacciotti, 2020) and (omarvivas, 2020) have used LightGBM Regressor for building the prediction model. (Vandewiele, 2020) has used GNNs and RNNs for creating the learning model.

Models like that of (Singhal, n.d.) and (fernandoramacciotti, 2020) have used Root Mean Squared Error(RMSE) as the loss function while (Vandewiele, 2020) has used Reconstruction Loss/MAE as the loss function. Most of the studies like that of (shadowburning, 2020; gagankarora, 2020; vbmokin, 2020; ruko, 2020) and (anzhemeng, 2020) have used Mean Columnwise Root Mean Squared Error (MCRMSE) as the loss function. It is clearly evident from the sources that models like (shadowburning, 2020) and (ruko, 2020) which have used MCRMSE as the loss function has a higher accuracy than models like (Singhal, n.d.) and (fernandoramacciotti, 2020) which used RMSE as the loss function. Also, an evaluation table created by (Singhal, n.d.) showed that among the RNNs used for building the models, LSTM had the highest accuracy score. The predicted values from the LGBM Regressor models were a little higher than that of the original values and this is clear from the models created by (fernandoramacciotti, 2020) and (omarvivas, 2020).

2.2. Research gaps and recommendations

Although the creation of stable mRNA molecules remains difficult, datasets of sequences and corresponding information are becoming more popular and widely available. Through the use of deep learning architectures, reasonable predictions of structural features can be obtained, as demonstrated in this manuscript, with mean RMSE values ranging from 0.24 to 0.31. The usage of such techniques has the capacity to increase the speed and efficiency of mRNA vaccine discovery and has further implications in other related fields of research. In order to improve the performance of the model a simple binary classification system should be created to predict whether a RNA molecule is stable at the end of every model. This would minimize the False Negative Rate and finally, the model needs to be predicted on longer RNA sequences to observe how it impacts the accuracy of such methods (Singhal, n.d.).

2.3. Feature Comparison Table (on all the research papers used for the Literature Review and Case Studies)

In order to contrast on the objectives of the different research papers used for constructing the Literature Review and throw light on the findings, research gaps and recommendations of each paper, a feature comparison table has been constructed which has been placed in section B.1 of the Appendix.

2.4. Conclusion

This chapter has focussed on a thorough research of mRNA vaccines and have extracted key points relating to both the advantages and disadvantages of the vaccine and the risks it poses in the clinical trials. A thorough research has been carried out on the existing models that has been used for predicting the stability of the mRNA vaccines so far. The key limitations of the existing models are summarised here: **most of the predicting models either used LSTM or GRU, many models used accuracy metrics like RMSE that yielded poor accuracy score, there was a lack of user-interface and none of the results from the previous models were well organised**. Chapter 3 would be addressing these limitations and proposing solutions to overcome them.

Chapter 3 – New Ideas

It is clearly evident from the Literature review and research from the previous section that most of the prediction models have either used a combination of RNNs – LSTM and GRU or LightGBM Regressor for the prediction purpose. In this project three prediction models were created – One using LightGBM Regressor, second using WaveNet and a third using LSTM and GRU. The combination of these three models haven't been utilised before.

3.1. Discussion of the identified limitations with proposed solutions

It is evident LightGBM was utilised for building a model because it focuses on the accuracy of results, can easily handle large size of data and takes lower memory to run (Mandot, 2018). Most of the studies like that of (Singhal, n.d.; shadowburning, 2020) and (gagankarora, 2020) that used LSTMs and GRUs had an excellent accuracy score and the predicted results were almost similar to the original values. Hence, a second model has been built using LSTMs and GRUs.

None of the models discussed in the context section has utilised WaveNet for prediction. WaveNet is a powerful new predictive technique that uses multiple Deep Learning strategies from Computer Vision and Audio Signal Processing models and applies them to longitudinal (time-series) data. It was created by researchers at London-based Artificial Intelligence firm and currently powers Google Assistant voices (Balaban, 2019). A technique outlined in a paper in September 2016, showed that WaveNet was able to generate realistic sounding human-like voices by directly modelling waveforms using a neural network method trained with recordings of real speech. Tests with US English and Mandarin showed that the system outperformed Google's existing text-to-speech (TTS) systems (Wikipedia, 2021). However, WaveNet is not just confined to prediction of audio, it is now being utilised for a variety of purposes. The study carried out by (Liu et al., 2019) showed that WaveNet has been used to predict wave height and period from accelerometer data using convolutional neural network. This project has utilised WaveNet for the prediction of RNA base degradation which is a regression problem and is first of its kind.

None of the previously created models had any Graphical User Interface associated with them which. However, in this project a Web Application using Flask, HTML and CSS has been created that provides detailed information on the project – it’s motivation and dataset used. The website also provides a platform that has organized the data visualizations, screenshots of evaluation and results during from the three models created. The website also provides a link that would enable a viewer to play the Eterna Game. However, it should be kept in mind that the Web Application created is aimed to gratify the coders or someone who would like to gain insight into the working details of this project.

3.2. Evaluation process

The loss function used for checking the accuracy of the models were Mean Columnwise Root Mean Squared Error (MCRMSE)

$$\mathrm{MCRMSE} = \frac{1}{N_t} \sum_{j=1}^{N_t} \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_{ij} - \hat{y}_{ij})^2}$$

where is the number of scored ground truth target columns, and and are the actual and predicted values, respectively (kaggle.com, n.d.). The primary reason for choosing MCRMSE as the loss function caters to the fact that MCRMSE has been used in Kaggle’s OpenVaccine Challenge for scoring all the submissions (kaggle.com, n.d.). Moreover, models like that of (shadowburning, 2020), (gagankarora, 2020), (fernandoramacciotti, 2020), (vbmokin, 2020), (ruko, 2020) and (anzhemeng, 2020) that used MCRMSE had a better accuracy score in comparison to models like (Singhal, n.d.) and (fernandoramacciotti, 2020) which used RMSE as the loss function.

3.3. User Interface

The entire project is concerned with a lot of data visualizations and this is absolutely necessary to understand the dataset well. The Eterna dataset is a complex dataset having the information of 3029 RNA sequences with a length of 107. The dataset has 19 features each affecting the rate of degradation of an RNA sequence. Moreover, the process of prediction of this data is also very complicated. Most of the projects have utilised two or more RNNs with one or more accuracy metrics for predicting the degradation rates but none of them has organised the results chronologically or categorically. Hence, one of the main aims of this project is to create an interface that would provide an organised template for viewing all the graphs and charts created and also display the accuracy scores of the 3 different models created along with the predictions made by these models.

The project intends on constructing a Web Application using Flask, HTML and CSS and would also be scalable across different devices. The website would present a viewer with important information on COVID-19 and would have valuable links to different websites containing information relevant to the project. It should be kept in mind that this web application would primarily benefit a programmer, data scientist, a medical researcher or someone who is looking forward to extend their own mRNA base predicting models.

3.4. Design Diagrams

This section contains a flowchart depicting the process flow of the proposed models.

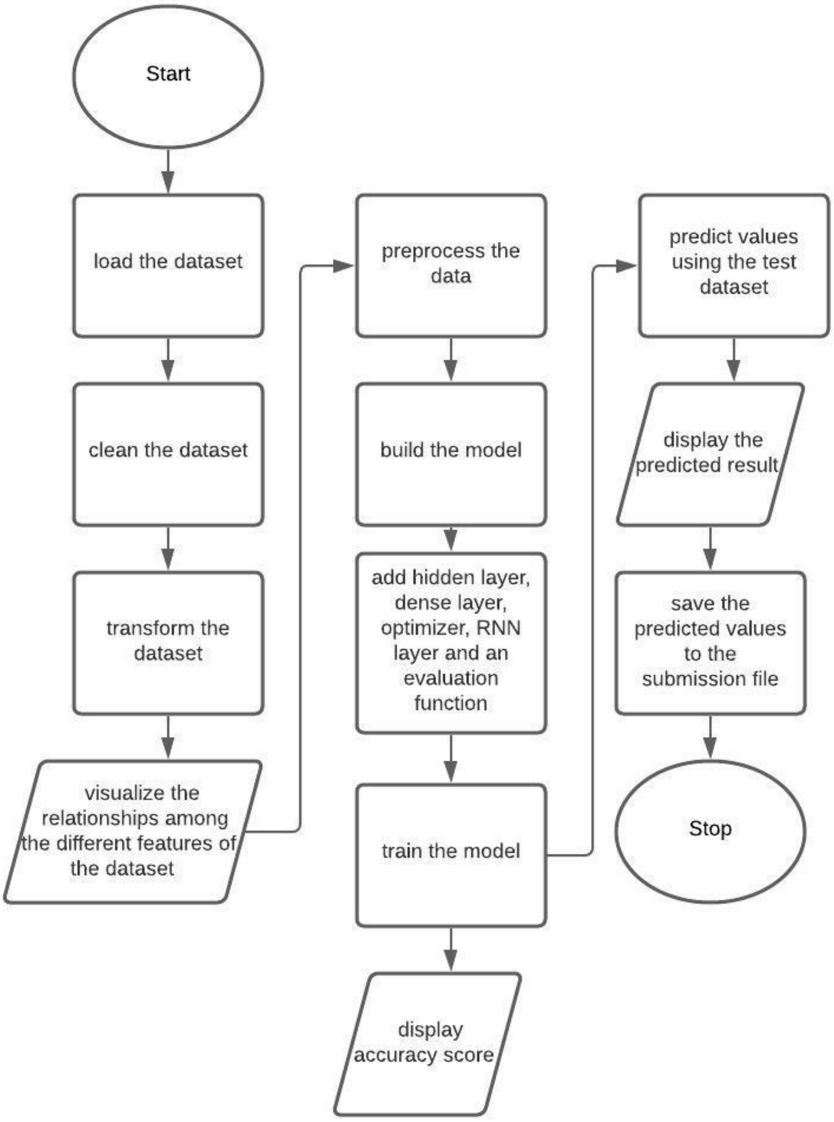

Start
load the dataset
clean the dataset
transform the dataset
visualize the relationships among the different features of the dataset
preprocess the data
build the model
add hidden layer, dense layer, optimizer, RNN layer and an evaluation function
train the model
display accuracy score
predict values using the test dataset
display the predicted result
save the predicted values to the submission file
Stop

Figure 1: Overview of the process flow of the proposed models.

3.5. Conclusion

This chapter was concerned with addressing the limitations highlighted in Chapter 2. Solutions to all the addressed limitations were proposed in this chapter. The addition of a new predicting model using Wave Net was besides LSTM and GRU was suggested. It was suggested that MCRMSE should be chosen as the accuracy metrics and the design for a web interface was also proposed. By building a solution using these new ideas, it would be possible to create a novel predictive model for mRNA degradation and present this research work in form of a web application to the programmers.

Chapter 4 – Implementation

This chapter is concerned with explaining the implementation of the proposed solution defined in Chapter 3 which is aimed at overcoming the loop holes identified in Chapter 2. This chapter gives detailed explanation of the methodology used for the project development, design diagrams showing the process flow and activity flow of the different models and web interface created, tools used for the development of the project and explanation of the entire implemented code. It also provides screenshots of the data visualizations and web interface created during implementation.

4.1. Methodology

The choice of methodology for the project development was scrum. The primary reason for choosing scrum as the development methodology was that the project could be disintegrated into 4 main parts – the development of a model using WaveNet, GRU+LSTM and LightGBM Regressor respectively and the development of the web based interface. Each of the 4 part of this project was concerned with a sprint which lasted for about 3 weeks. At the beginning of each sprint an initial plan would be made on how the respective part of the project was going to get completed, requirements were identified and finally an initial deadline was assigned which would be set at every two weeks from the start of a sprint. After the initial deadline, the work would be reviewed, necessary changes would be identified and finally, that part of the project would be tested. There was a three days interval between each of the sprints and there were four sprints in total throughout the course of the project.

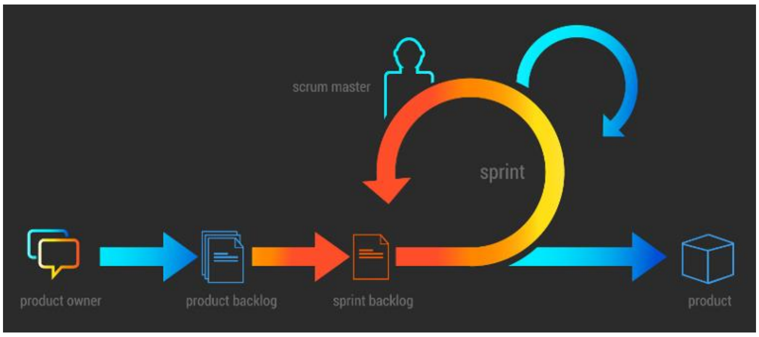


Figure 2: Scrum methodology (AIM Consulting Group, LLC, 2016).

4.2. Design Diagrams

4.2.1. Process Flow Diagram illustrating the process flow for the LightGBM Regressor model

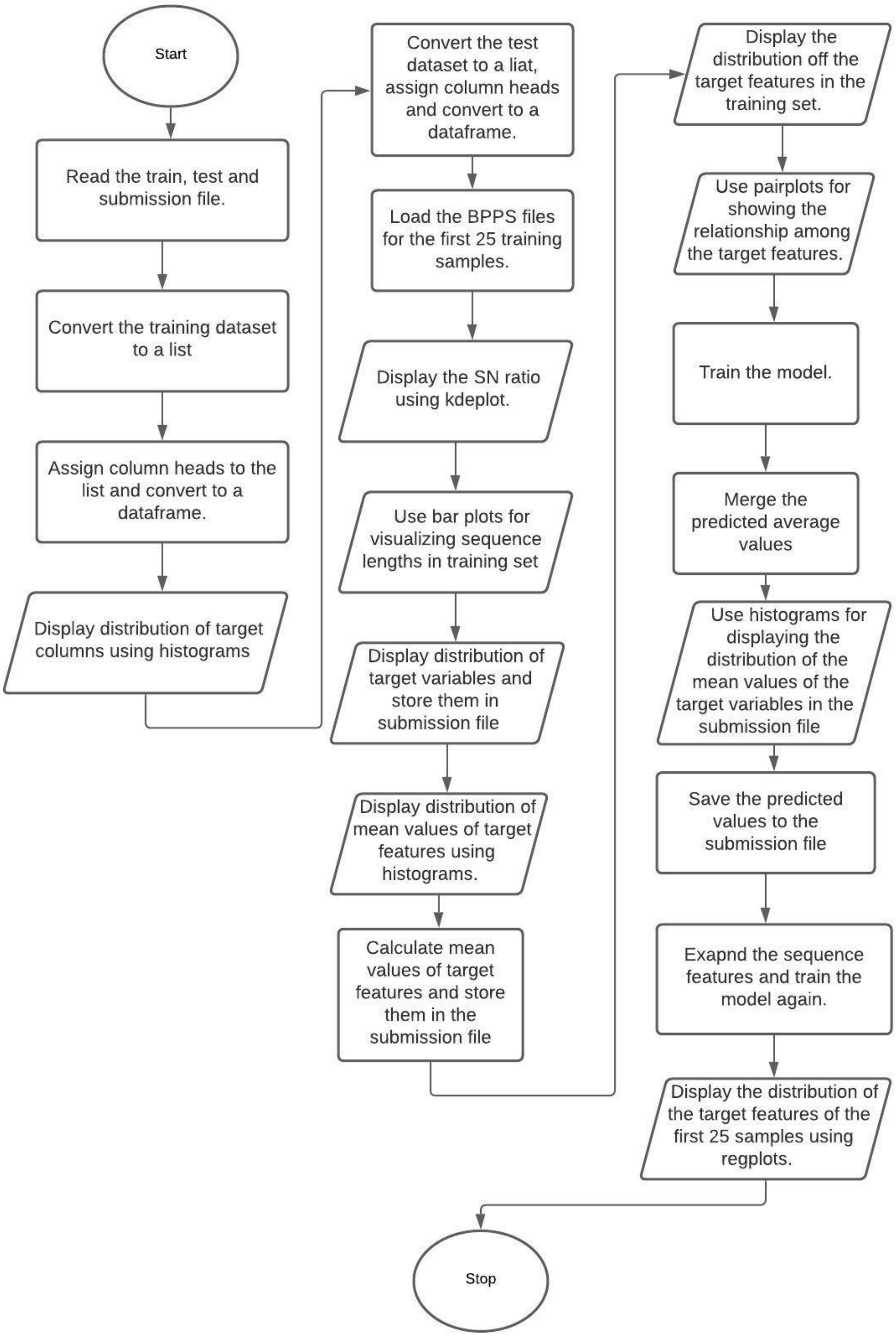

Start
Read the train, test and submission file.
Convert the training dataset to a list
Assign column heads to the list and convert to a dataframe.
Display distribution of target columns using histograms
Convert the test dataset to a liat, assign column heads and convert to a dataframe.
Load the BPPS files for the first 25 training samples.
Display the SN ratio using kdeplot.
Use bar plots for visualizing sequence lengths in training set
Display distribution of target variables and store them in submission file
Display distribution of mean values of target features using histograms.
Calculate mean values of target features and store them in the submission file
Display the distribution off the target features in the training set.
Use pairplots for showing the relationship among the target features.
Train the model.
Merge the predicted average values
Use histograms for displaying the distribution of the mean values of the target variables in the submission file
Save the predicted values to the submission file
Exapnd the sequence features and train the model again.
Display the distribution of the target features of the first 25 samples using regplots.
Stop

Figure 3: Process Flow Diagram illustrating the process flow for the LightGBM Regressor model

4.2.2. Process Flow Diagram illustrating the process flow of the WaveNet model

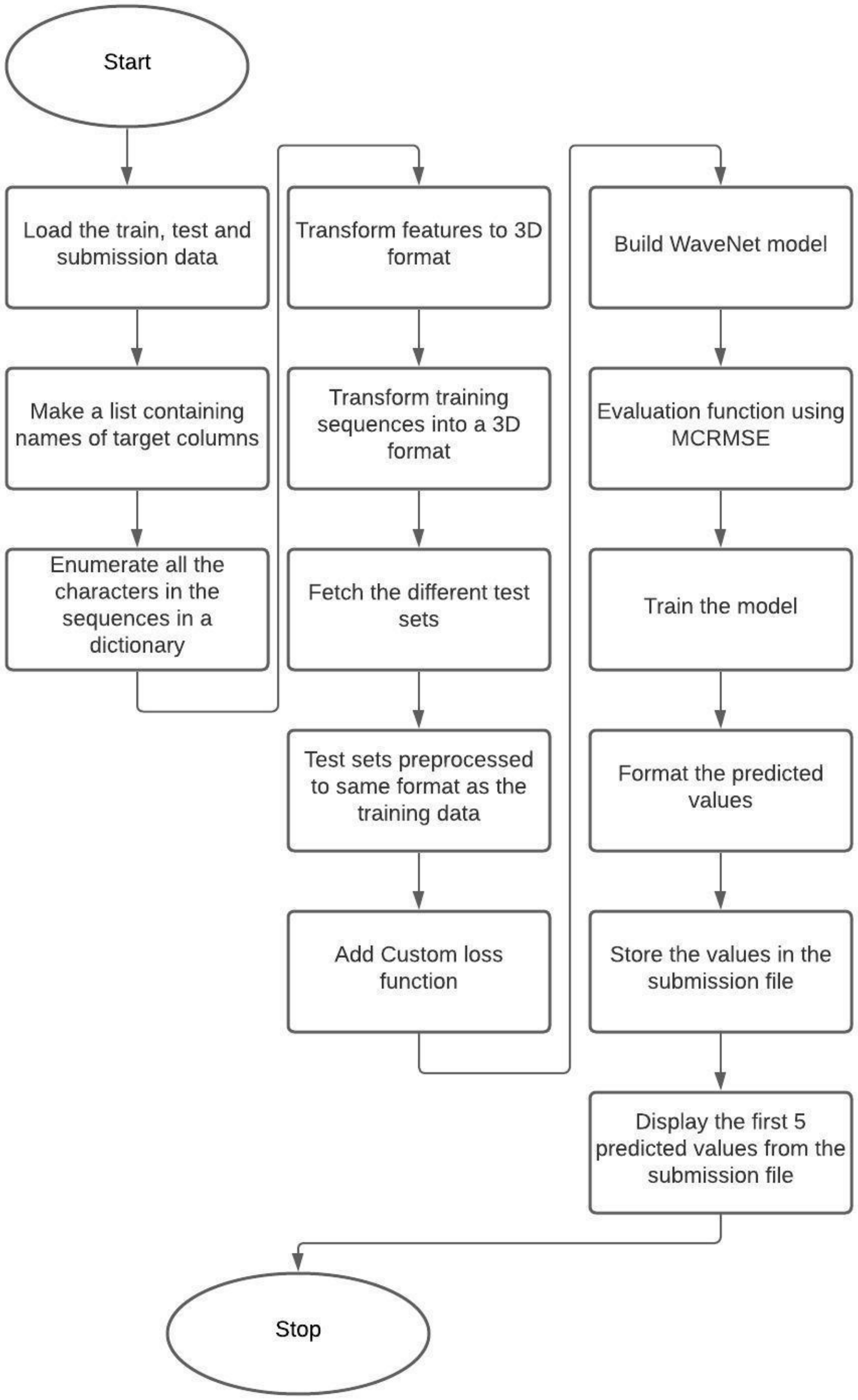

Start
Load the train, test and submission data
Make a list containing names of target columns
Enumerate all the characters in the sequences in a dictionary
Transform features to 3D format
Transform training sequences into a 3D format
Fetch the different test sets
Test sets preprocessed to same format as the training data
Add Custom loss function
Build WaveNet model
Evaluation function using MCRMSE
Train the model
Format the predicted values
Store the values in the submission file
Display the first 5 predicted values from the submission file
Stop

Figure 4: Process Flow Diagram illustrating the process flow of the WaveNet model

4.2.3. Process Flow Diagram illustrating the process flow of the GRU and LSTM model

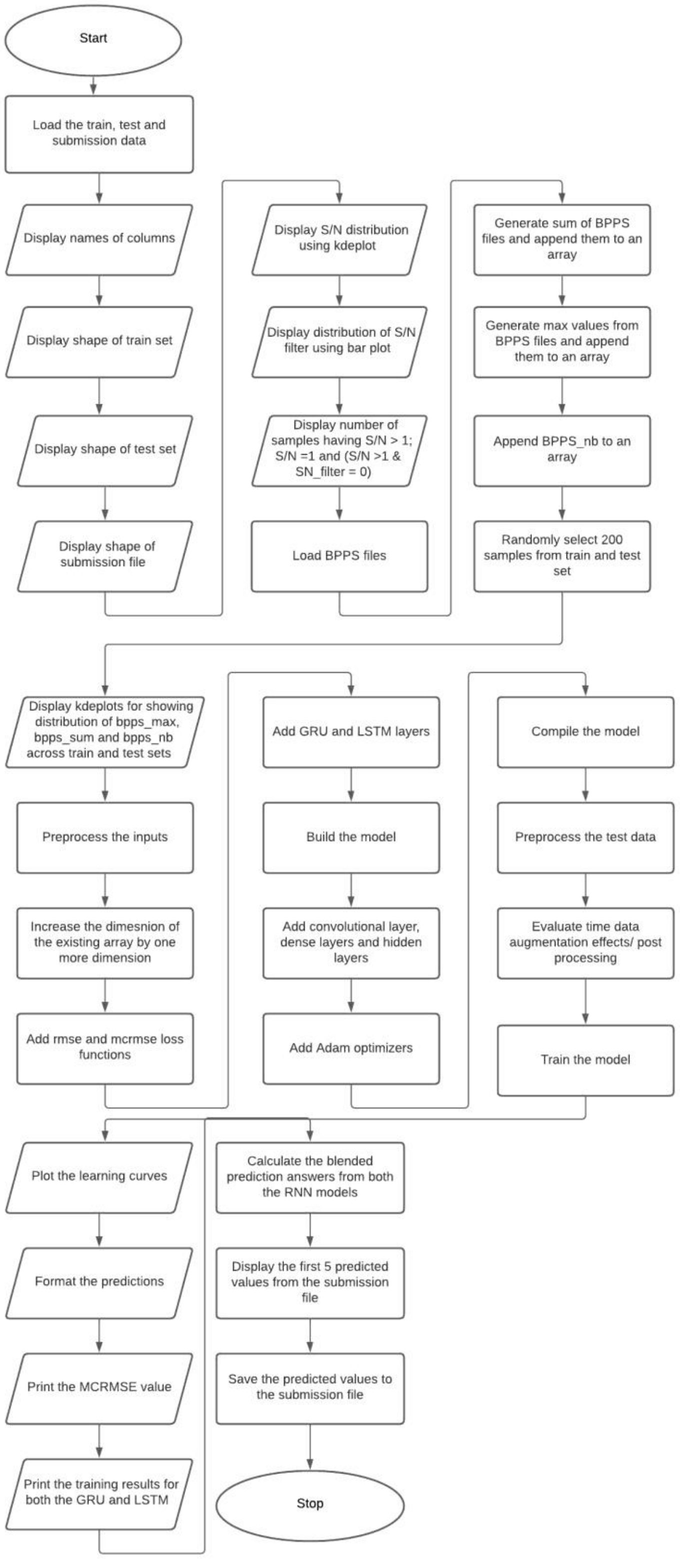

Start
Load the train, test and submission data
Display names of columns
Display shape of train set
Display shape of test set
Display shape of submission file
Display S/N distribution using kdeplot
Display distribution of S/N filter using bar plot
Display number of samples having S/N > 1; S/N =1 and (S/N >1 & SN_filter = 0)
Load BPPS files
Generate sum of BPPS files and append them to an array
Generate max values from BPPS files and append them to an array
Append BPPS_nb to an array
Randomly select 200 samples from train and test set
Display kdeplots for showing distribution of bpps_max, bpps_sum and bpps_nb across train and test sets
Preprocess the inputs
Increase the dimesnion of the existing array by one more dimension
Add rmse and mcrmse loss functions
Add GRU and LSTM layers
Build the model
Add convolutional layer, dense layers and hidden layers
Add Adam optimizers
Compile the model
Preprocess the test data
Evaluate time data augmentation effects/ post processing
Train the model
Plot the learning curves
Format the predictions
Print the MCRMSE value
Print the training results for both the GRU and LSTM
Calculate the blended prediction answers from both the RNN models
Display the first 5 predicted values from the submission file
Save the predicted values to the submission file
Stop

Figure 5: Process Flow Diagram illustrating the process flow of the GRU and LSTM model

4.2.4. Activity Flow Diagram illustrating the activities that can be carried out on the Web Interface

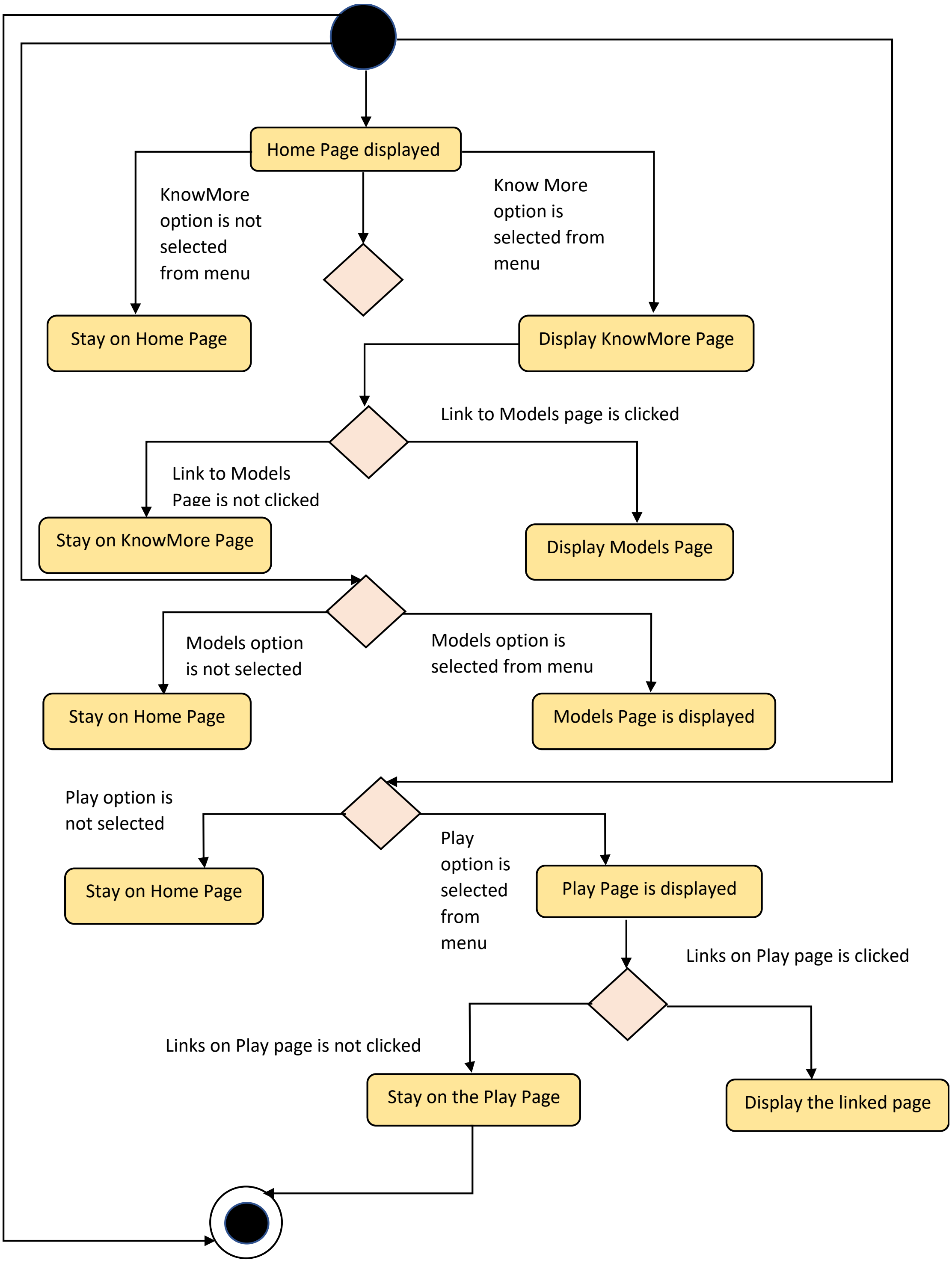

Home Page displayed
KnowMore option is not selected from menu
Know More option is selected from menu
Stay on Home Page
Display KnowMore Page
Link to Models page is clicked
Link to Models Page is not clicked
Stay on KnowMore Page
Display Models Page
Models option is not selected
Models option is selected from menu
Stay on Home Page
Models Page is displayed
Play option is not selected
Play option is selected from menu
Stay on Home Page
Play Page is displayed
Links on Play page is clicked
Links on Play page is not clicked
Stay on the Play Page
Display the linked page

Figure 6: Activity Flow Diagram illustrating the activities that can be carried out on the Web Interface

4.3. Tools Used

4.3.1. Choice of Environment

Google Colab/Colaboratory was chosen as the working environment. Here are the reasons why:-

- Allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.
- Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs.
- Colab is free to use (research.google.com, n.d.).

Figure 7: The logo of Google Colab (Digraje, 2019).

4.3.2. Deep Learning Methods Used

4.3.2.1. LightGBM Regressor

Tree-based framework uses gradient boosting algorithm, grows leafwise and chooses the leaf with the maximum delta loss to grow. Advantage of leaf-wise algorithm over level-wise algorithm is that it can reduce more loss. Here are the reasons why LGBM was used for building a model in the project:-

- Prefixed ‘Light’ because of its high speed.
- Can handle large size of data.
- Takes low memory to run.
- Focuses on accuracy of results.
- Supports GPU learning.
- Open-source library.

- Extends gradient boosting by adding a type of automatic feature selection while focusing on boosting examples with larger gradients (Mandot, 2018).

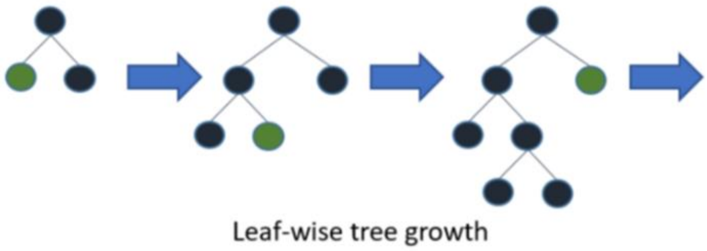


Figure 8: The picture above explains how LGBM works (Mandot, 2018).

4.3.2.2. LSTM

Consists of a cell, input gate, an output gate and a forget gate. The cell remembers the value over arbitrary time intervals and 3 gates regulate the incoming and outgoing information. Here are the reasons why it was chosen for building a model in the project:-

- Well suited for classification, processing and making predictions on time series data.
- Efficient in dealing with vanishing gradient problem.
- Relative insensitivity to gap length is an advantage of LSTMs over RNNs (Wikipedia Contributors, 2018).

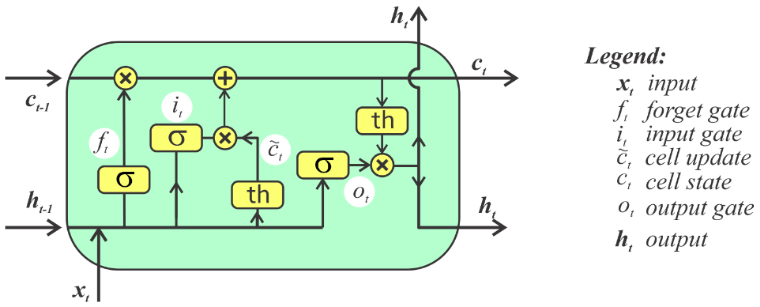


Figure 9: The picture above shows the structure of a LSTM unit (Predicting weather using LSTM, 2021).

4.3.2.3. GRU

Grus are a gating mechanism in RNNs introduced in 2014 by Kyunghyun Cho. It is similar to LSTM with having a forget gate but has fewer parameters and lacks an output gate. Here are the reasons why it was chosen for building a model in this project:-

- It controls the flow of information just like LSTM unit without having to use a memory unit.
- It is computationally efficient.
- It is less complex as compared to LSTM (Wikipedia Contributors, 2019).

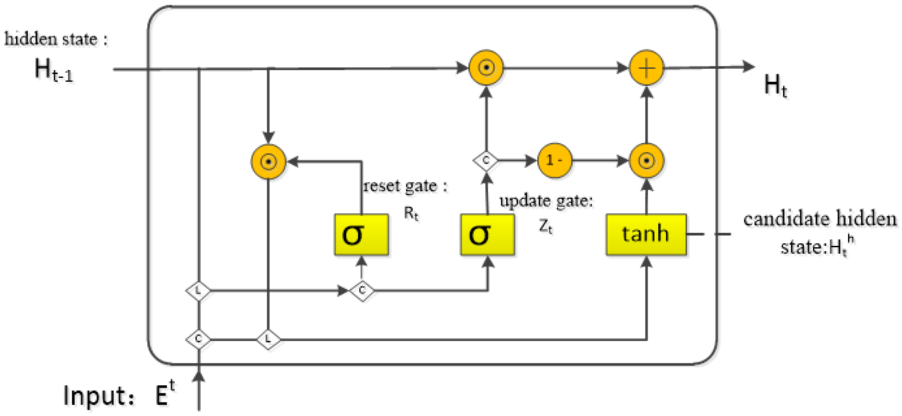


Figure 10: The picture above shows the structure of a GRU (2021).

4.3.2.4. WaveNet

It is a DNN for generating audio and was created by researchers at London-based AI firm named DeepMind (Wikipedia, 2021). Here are the reasons why it was chosen for building a model:-

- Works well on large set of data.
- Speed is similar to that of LGBM Regressor.
- Has rarely been used for predicting regression values like RNA base degradation.

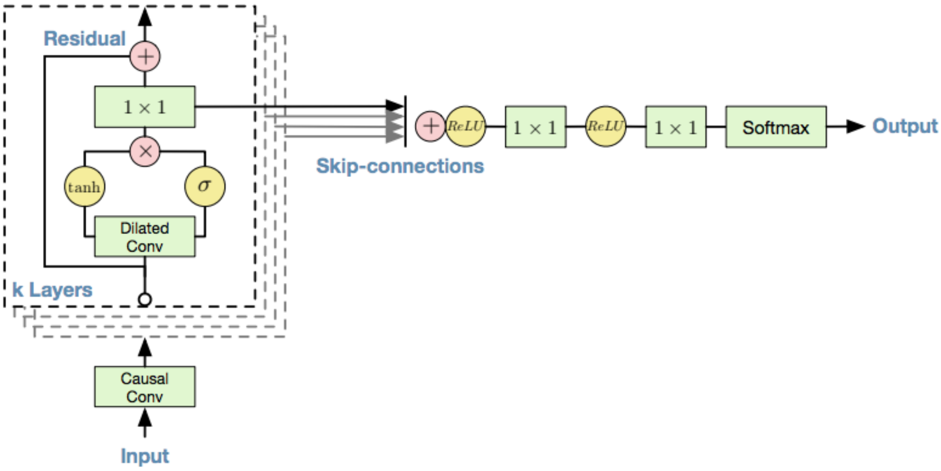


Figure 11: The picture above shows the structure of WaveNet (Understanding WaveNet architecture, 2021).

4.3.3. BPPS Matrix

The bpps are pre-calculated NumPy arrays for each RNA sequence. Biophysically speaking, this matrix gives the probability that each pair of nucleotides in the RNA forms a base pair (given a particular model of RNA folding). At the simplest level -- it's a symmetric square matrix with the same length as the sequence, so you can get N more features out of it, if you want them. Each column and each row should sum to one (up to rounding error), but more than one entry in each column/row will be nonzero -- usually somewhere between 1-5 entries.

In the project, a readily available N X N matrix was used. This was used to determine the likely structures of mRNA. bpps is base probability matrix, a matrix of the probability of each base dating with every other base in the mRNA. So, assuming the mRNA molecule consists of 10 characters: ACAUUGGGAU. Then the bpps is a 10 x 10 matrix. Taken the first base as 'A'. So the probability of the first base 'A' pairing up with the remaining 9 bases. A pairing with itself is 0 and the sum of the probabilities of the remaining 9 bpp will be the total probability of A being paired. So, **P(the probability of 'i' th base in the sequence seq being paired) = sum of probabilities of P(i,j) being paired) from j=0 to len(seq)-1**.

4.3.4. Libraries Used

A number of Python libraries has been utilised for successfully implementing the code. This includes some libraries from Python, libraries used for statistical data analysis as well as libraries from TensorFlow and keras backend. The names of the libraries along with th reason for using them in the project has been summarized in form of a table which is present in the C.1 section of the Appendix.

4.4. Choice of Language

Python was chosen as the language for creating the learning models. The reasons for choosing Python as the primary programming language for writing the code of this project are:-

- Easy to learn.
- Easily understandable.
- Has an extensive set of libraries used for Deep Learning and Machine Learning.
- It is platform independent and can be used on multiple platforms using Windows, Linux and macOS.

Figure 12: The logo of Python (Air, n.d.).

HTML was used for creating the structure of the web interface and CSS was used for creating the template which added style to the web pages.

Figure 13: The logo of HTML and CSS (www.deviantart.com, n.d.).

4.5. Description of the dataset

This section provides information about the dataset which was used for training and testing the different predictive models built during the project.

The dataset used in this project was made of collected data of 3029 RNA sequences. Experimental data of 5 target variables were presented for the first 68 RNA bases. This data was further divided into 2 parts having 2400 and 640 sequences respectively. The former was put into the training dataset and the latter was put into the test dataset.

The description of the different columns in the dataset has been summarised in a table which is presented in the C.2 section of the Appendix.

4.6. Description of Functionality (of the learning models)

This section provides an explanation of the code implemented. It is divided into four parts – LightGBM Regressor Model, WaveNet Model and LSTM + GRU model. The necessary header files needed for each section of the code is imported at the start. Headings and comments have been provided throughout the code for easy identification of the different models built. The names of the different identifiers has been provided in ***black bold and italics font*** while the names of the target features have been provided in ***blue bold and italic font*** for easy identification.

4.6.1. LightGBM Regressor

Loading the dataset: First the train, test and the sample submission files were loaded into the system. All these files were in the .json format. The shapes of the train and the test dataset were checked. The train dataset had 2400 rows with 19 different columns while the test dataset had 3634 rows with 7 different columns. The different column head names was printed to make the programmer accustomed with the dataset. The values of the first 3 rows of the train dataset was displayed followed by the column names in the test dataset.

Data cleaning and Data Transformation: A list named ***train_data[]*** was defined. The unique 'id' values were checked in the training dataset and the records having unique id's were retrieved. The length of the different columns in the training set was 68 (each column is a 1x68 vector). Hence, a loop with range 68 was run to retrieve all the values from these records and these values were appended in the list ***train_data[].*** The ***train_data[]*** list was converted to a DataFrame and the column names were set to be equivalent to that of the train dataset. The first 5 values of this newly created DataFrame was printed.

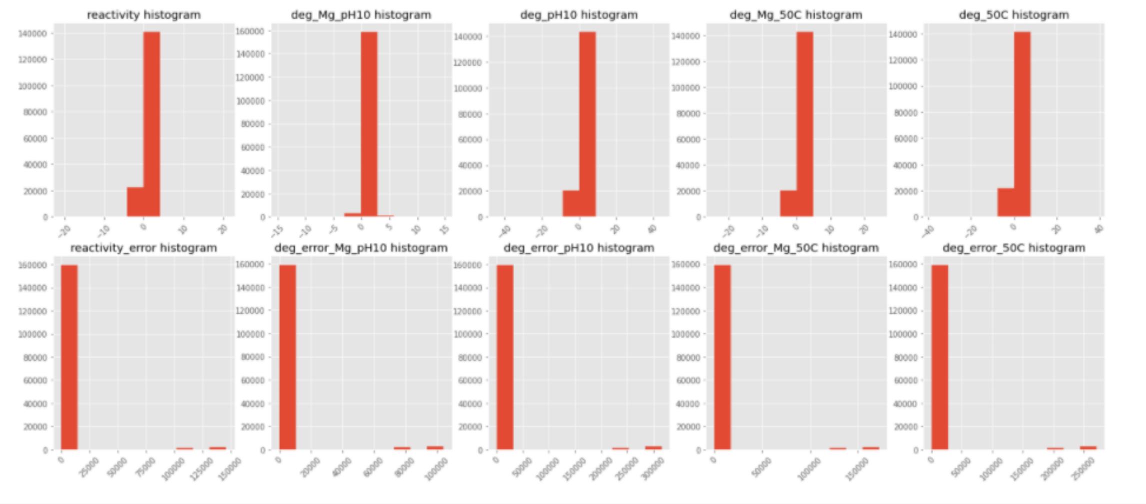


Figure 14: Histograms were used to show the frequency distribution of the different columns in the dataframe.

Similar to the training dataset, a list named ***test_data[]*** was defined. The records with the unique values were retrieved. The values under the 5 different columns **(*id, sequence, structure, seq_scored, predicted_loop_type*)** present in the records were extracted and then appended in the list ***test_data[].*** This list was converted to a DataFrame and the column names were set equivalent to the column heads of the test dataset. The first 5 rows in the test DataFrame were printed.

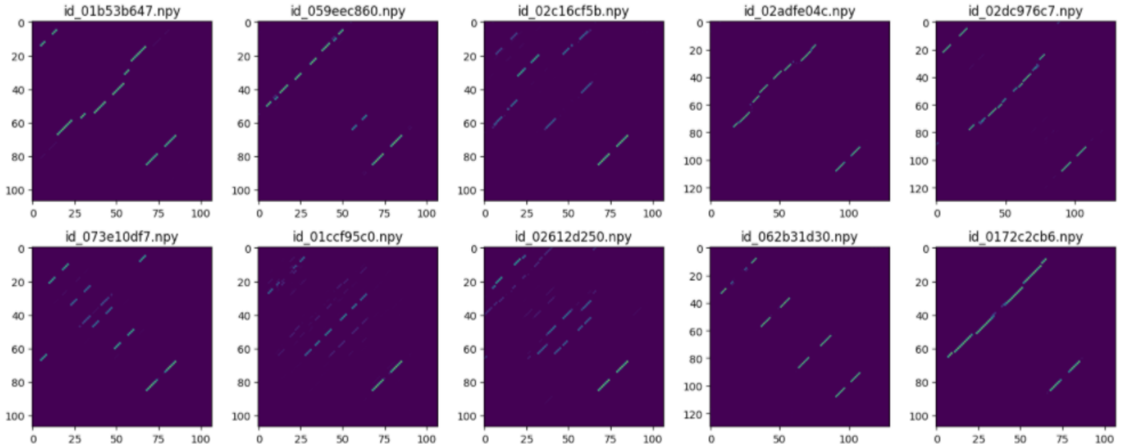


Figure 15: The bpps matrices for the first 25 sample RNA sequences were loaded and displayed.

The values of the ***Reactivity, deg_Mg_pH10, deg_Mg_50C, deg_pH10*** and ***deg_50C*** in the training DataFrame was rounded to 2 places off decimal.

Signal to Noise can be stated as mean(measurement value over 68 nts)/mean(statistical error in measurement value over 68 nts).

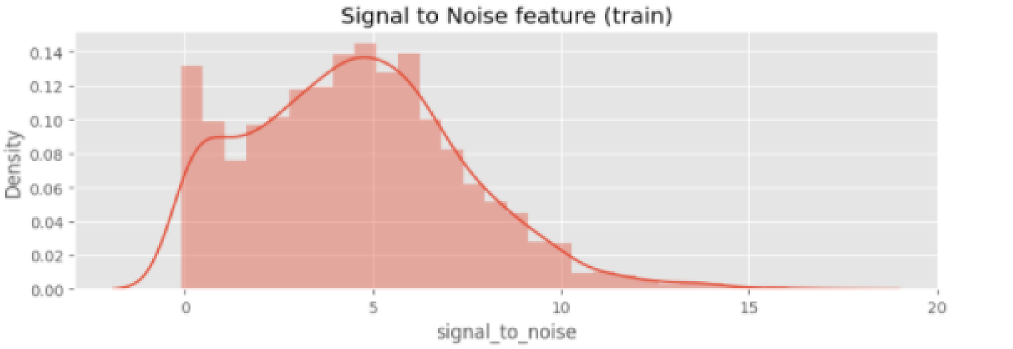


Figure 16: A distribution plot has was to show the distribution of the S/N filter in the training data.

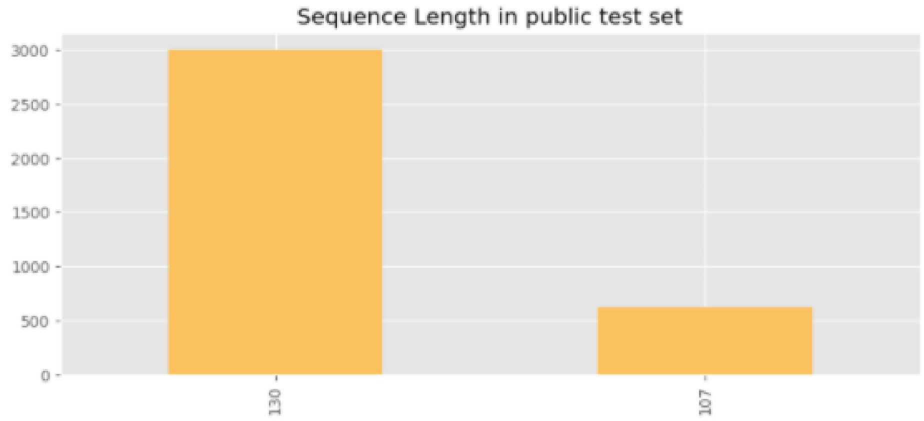


Figure 17: The distribution of the 2 types of sequence lengths of the RNA sequences (107 in the train set & public test set and 130 in the private test set) was visualized using bar plots.

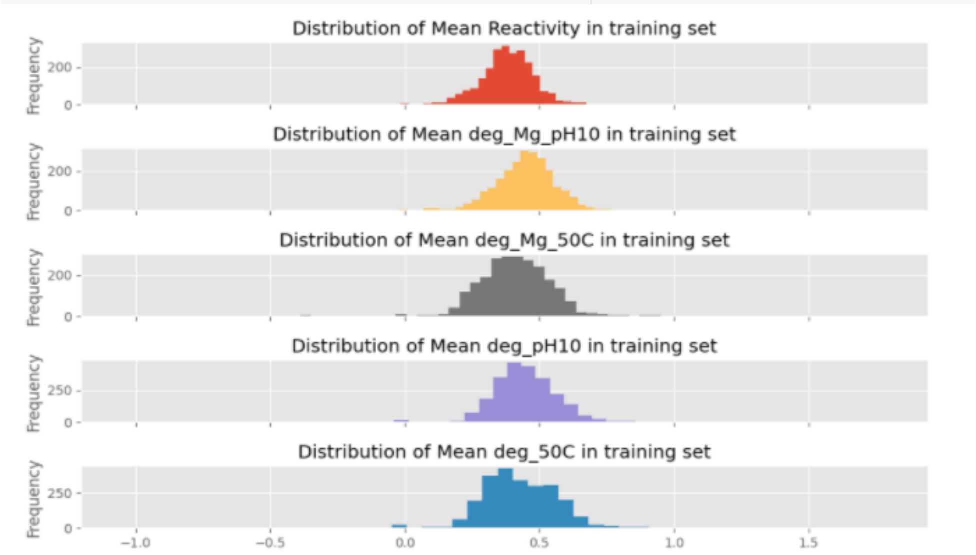


Figure 18: The mean values of each record of the 5 target columns (reactivity, deg_Mg_pH10, deg_Mg_50C, deg_pH10, deg_50C) was calculated and histograms were plotted to show their frequency distribution throughout the training set.

These mean values were assigned to the records in the submission.csv file.

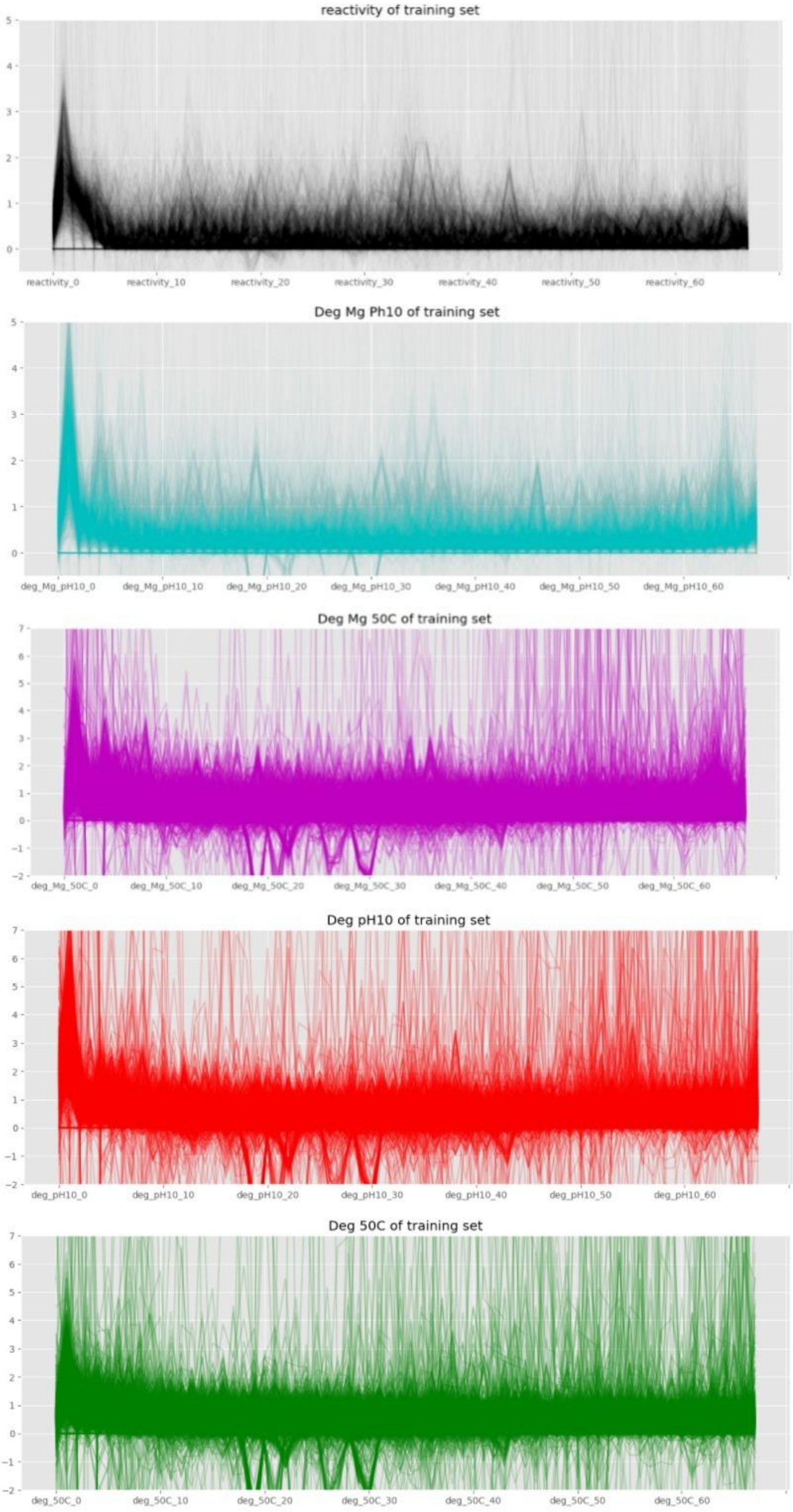


Figure 19: The distribution of the values of the 5 target columns without their errors was visualized using graphs.

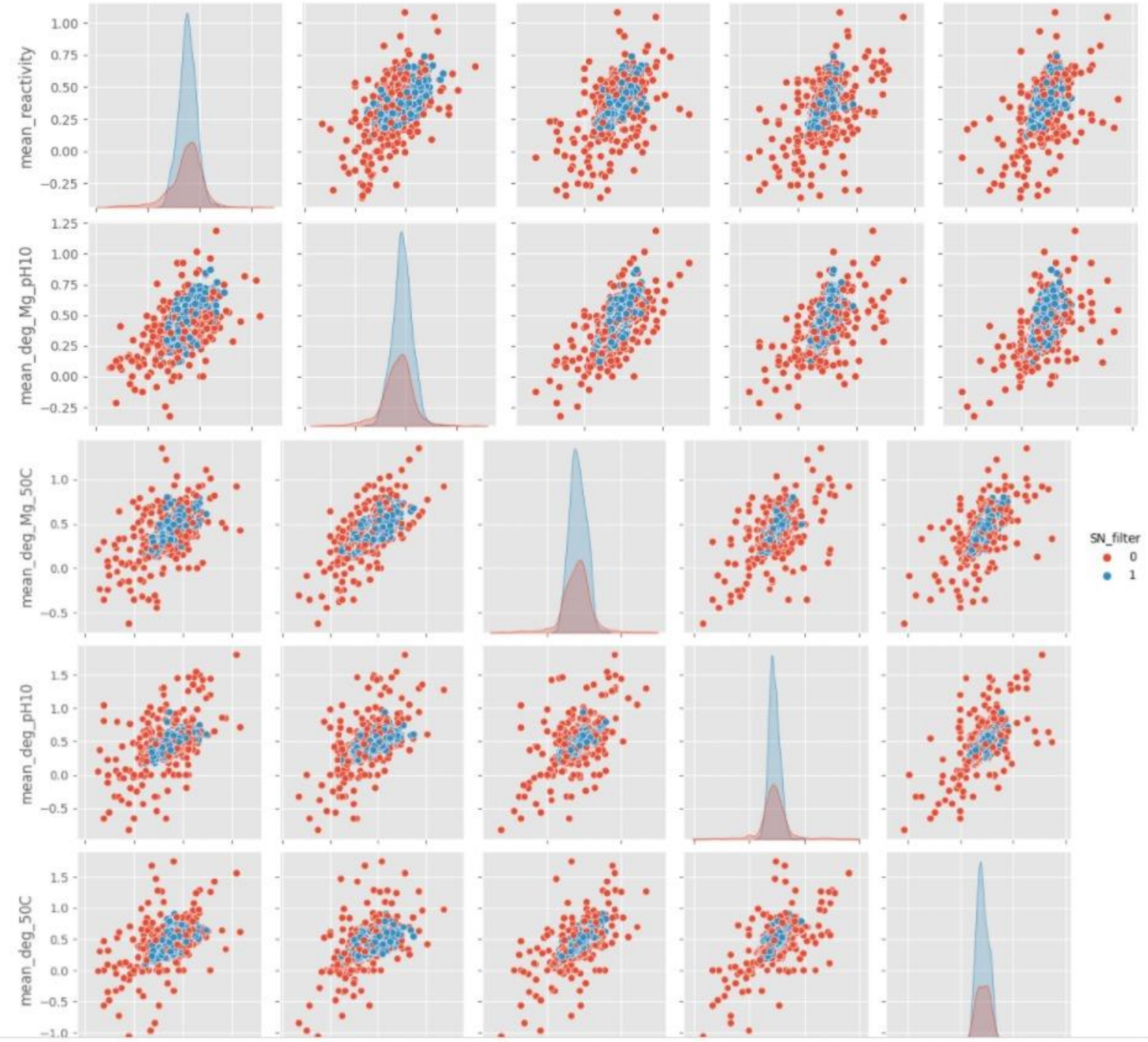
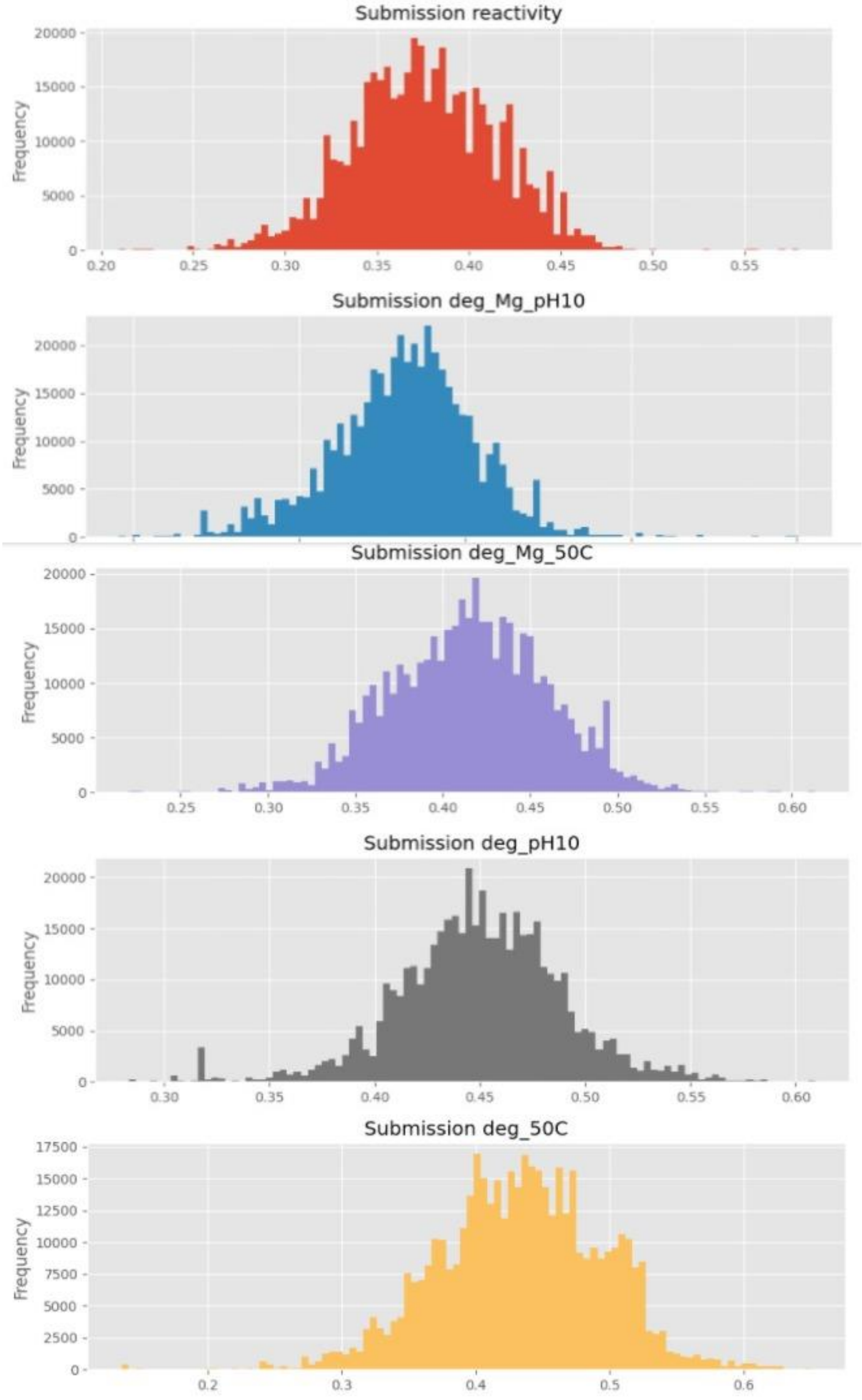


Figure 20: A pairplot was used to visualize the relationship among the mean values of the 5 target columns of the training set. The hue was set to SN_filter to map plot aspects to different colours.

Data Prediction: Since the sequence length is 107, a loop of range 107 was initiated, the ***sequence*** column in the train set and the test set was converted to a categorical type. The dataset was split into the train and test data and Light GBM Regressor was used for predicting the target values from the test set. The predicted average values was assigned to the respective columns in the submission.csv file and the column names were renamed accordingly (for example, ‘mean_reactivity_pred’ to ‘reactivity’).



Submission reactivity

Submission deg_Mg_pH10

Submission deg_Mg_50C

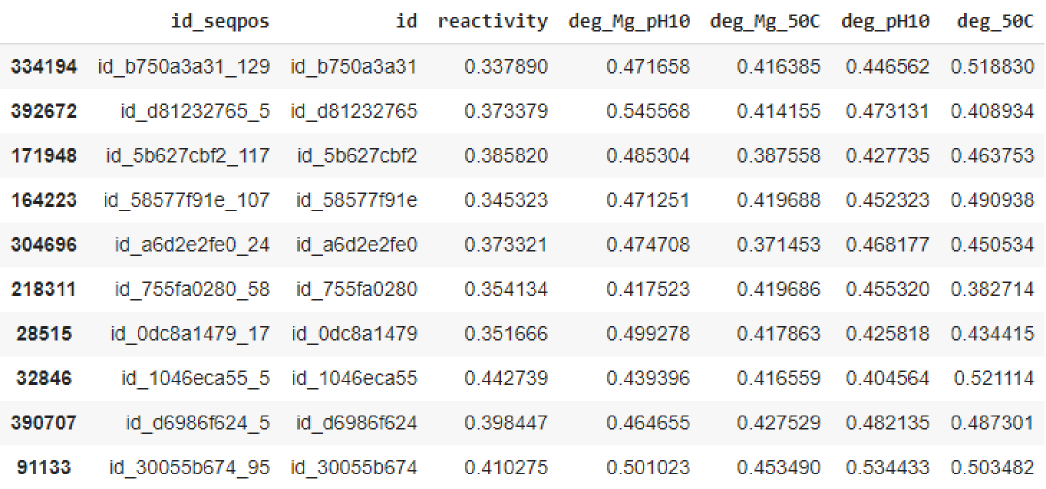
Submission deg_pH10

Submission deg_50C

Figure 21: The predicted target values in the submission.csv file were visualized using histograms. The first 10 rows of the file were displayed.

The sequence features were expanded and this time LGBM regressor was used to train the model on the sum of the sequence, structure and predicted loop type columns. The predicted values of the target columns were stored in a sample_submission.csv file. The values of each of the predicted target columns was visualized using histograms. Regplots were used to show the multicollinearity of the first 25 samples of the 5 target variables in the training set. Since, the replots are big in size, they have been shifted to C.3 of Appendix.

Accuracy score and predicted values: The KFold Cross Validation score at the end of training came out be 0.0176721.



	id_seqpos	id	reactivity	deg_Mg_pH10	deg_Mg_50C	deg_pH10	deg_50C
334194	id_b750a3a31_129	id_b750a3a31	0.337890	0.471658	0.416385	0.446562	0.518830
392672	id_d81232765_5	id_d81232765	0.373379	0.545568	0.414155	0.473131	0.408934
171948	id_5b627cbf2_117	id_5b627cbf2	0.385820	0.485304	0.387558	0.427735	0.463753
164223	id_58577f91e_107	id_58577f91e	0.345323	0.471251	0.419688	0.452323	0.490938
304696	id_a6d2e2fe0_24	id_a6d2e2fe0	0.373321	0.474708	0.371453	0.468177	0.450534
218311	id_755fa0280_58	id_755fa0280	0.354134	0.417523	0.419686	0.455320	0.382714
28515	id_0dc8a1479_17	id_0dc8a1479	0.351666	0.499278	0.417863	0.425818	0.434415
32846	id_1046eca55_5	id_1046eca55	0.442739	0.439396	0.416559	0.404564	0.521114
390707	id_d6986f624_5	id_d6986f624	0.398447	0.464655	0.427529	0.482135	0.487301
91133	id_30055b674_95	id_30055b674	0.410275	0.501023	0.453490	0.534433	0.503482

Figure 22: This table shows the predicted values for the first 10 rows from the LGBM Regressor Model.

4.6.2. Better LightGBM Regressor

Loading the dataset: 3 lists named ***SEQUENCE_COLS[], STRUCTURE_COLS[] and PRED_LOOP_TYPE_COLS[]*** were defined. A range of 130 was set because the sequence length was 130 in the private test set. In the loop, the values of the sequence, structure and the predicted_loop_type columns was stored in a dataframe. The records in the sequence and structure columns were appended in the ***SEQUENCE_COLS*** and the ***STRUCTURE_COLS***. ***expand_columns()*** was used to add columns to the dataframe passed as parameter for each of the sequences.

Data Processing and Data Transformation: The function ***parse_sample_submission()*** was used for splitting the id and sequence positions for ach record in sample submission. Function **get_train_long()** was used for this purpose: all the columns were padded with a constant value 107, a dataframe was created, appropriate column heads were added and the newly created dataframe was finally returned. Function ***get_test_long()*** was used for this purpose: columns ***id, seqpos, sequence, structure*** and ***predicted_loop_type*** were padded appropriately and the corresponding dataframe was finally returned. Function ***add_long_features()*** was used for this purpose: records with sequence less than or equal to 106 were filtered out, grouped by id column, merged with the dataframe passed as parameter. Check was done if merged key was unique in the filtered dataframe being merged. The process was done for ***sequence, structure*** and ***predicted_loop_type*** columns. These were further shifted using the sliding window technique.

Data training: LGBMRegressor was run with a learning rate of 1% and no. of boosted trees to fit as 100. The importance_type = 'gain' hyperparameter tuning allowed the result to contain total gains of splits which used the feature. A horizontal bar graph was plotted based on the relative importance of the dataframe columns. These graphs have been presented in the C.4 section of the Appendix Finally, histograms were plotted for all the 5 target columns based on training and test data. The histograms have been presented in the C.5 section of the Appendix.

Accuracy score: KFold Cross Validation was used for checking the accuracy. The accuracy score came out to be 0.510428.

4.6.3. WaveNet + GRU Model

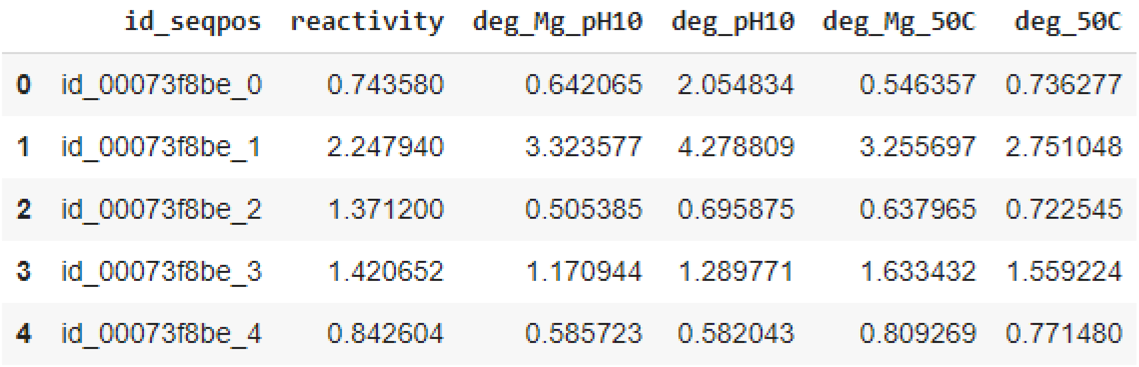
Train and Public Test dataset lengths: The sequence length of every feature in the training dataset and public test dataset was noted. ***sequence, structure*** and ***predicted_loop_type*** had a length of 107. ***reactivity, deg_Mg_pH10, deg_pH10, deg_Mg_50C*** and ***deg_50C*** had a length of 68.

Private Test dataset lengths: The sequence length of every feature in the private test dataset was noted. ***sequence, structure*** and ***predicted_loop_type*** had a length of 130. ***reactivity, deg_Mg_pH10, deg_pH10, deg_Mg_50C*** and ***deg_50C*** had a length of 91.

Data pre-processing: The data was pre-processed and the sequence was tokenized including the secondary structure and loop type. Different functions were used for this purpose. Function ***preprocess_inputs()*** transformed features to 3d format. Function ***cmcrmse()*** was the custom loss function. Function ***build_model()*** was used to build the wave net model. Default sequence length used was 107 with a dropout ratio of 10%. Adam optimizer was used. Loss and metric were also calculated.

Data training: Function ***Train_and_evaluate()*** was used for this purpose: Kfold cross validation was used. ReduceLROnPlateau from Keras was used for scheduling learning rate with patience hyperparameter set to 5. A bi-directional GRU model was trained having layers and dropout. Out-of-folds predictions were done. Function ***inference_format()*** formatted the predictions and wrote the submission to file submission.csv. All the information was used to train the model on degradations recorded by the researchers from OpenVaccine. The model was run on the public test set (shorter sequences) and the private test set (longer sequences), and the predictions were saved to the submission file.

Accuracy score and predicted values: The value of the Mean Columnwise Root Mean Squared Error at the end of the training came out to be 0.8407214440679358.



	id_seqpos	reactivity	deg_Mg_pH10	deg_pH10	deg_Mg_50C	deg_50C
0	id_00073f8be_0	0.743580	0.642065	2.054834	0.546357	0.736277
1	id_00073f8be_1	2.247940	3.323577	4.278809	3.255697	2.751048
2	id_00073f8be_2	1.371200	0.505385	0.695875	0.637965	0.722545
3	id_00073f8be_3	1.420652	1.170944	1.289771	1.633432	1.559224
4	id_00073f8be_4	0.842604	0.585723	0.582043	0.809269	0.771480

Figure 23: The figure above shows the predicted values for the first 5 rows from the WaveNet model.

4.6.4. GRU + LSTM Model

Data loading: The function ***seed_everything()*** was used to generate a random number everytime. The train set, test set and the sample submission files were loaded. The names of the columns in the train set was printed. The train set had 2400 rows and 19 columns and no missing values were found. The test set had 3634 rows with 7 columns and no missing values were found. The format of the sample submission file was checked.

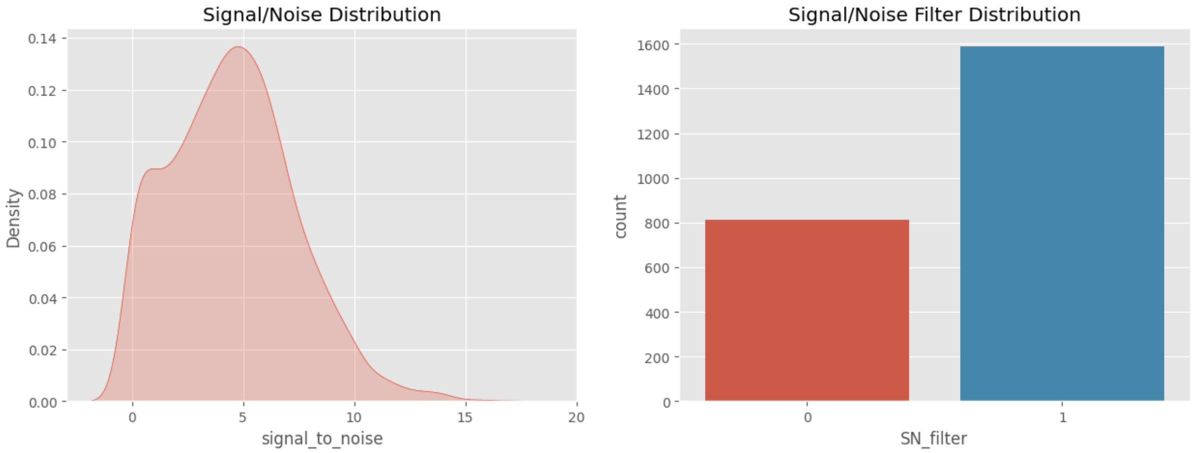
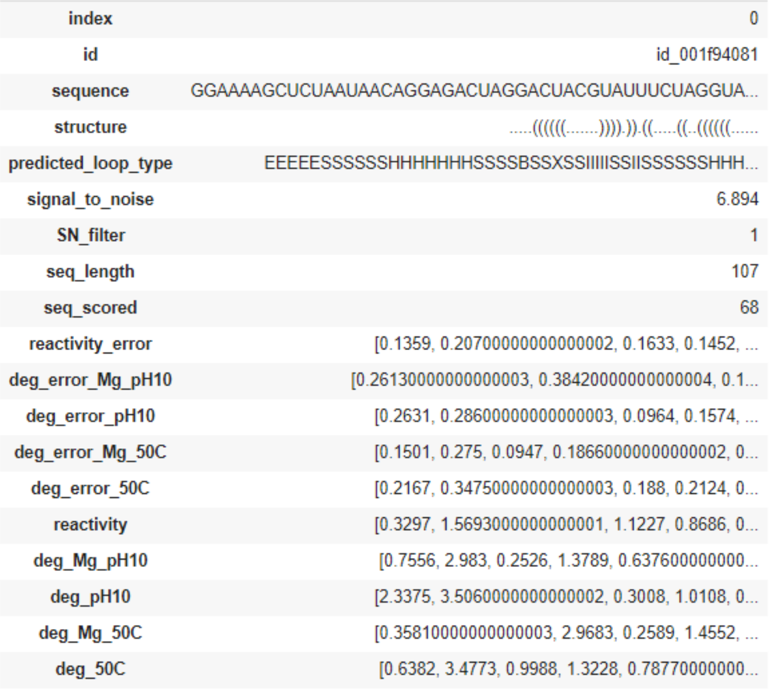


Figure 24: A KDE plot and a count plot was used to visualize the signal to noise distribution.

Data cleaning and transformation: The data that was presented was the sequence and the predicted structure and loop type of each base in the RNA. The feature reactivity measured the degradation at each base. The higher the reactivity the more likely the RNA is to degrade at that base.



index	0
id	id_001f94081
sequence	GGAAAAGCUCUAAUAACAGGAGACUAGGACUACGUAUUUCUAGGUA...
structure	.....((((((.......)))).)).((.....((..((((((......
predicted_loop_type	EEEEESSSSSSHHHHHHHSSSSBSSXSSIIIIISSIISSSSSSHHH...
signal_to_noise	6.894
SN_filter	1
seq_length	107
seq_scored	68
reactivity_error	[0.1359, 0.20700000000000002, 0.1633, 0.1452, ...
deg_error_Mg_pH10	[0.26130000000000003, 0.38420000000000004, 0.1...
deg_error_pH10	[0.2631, 0.28600000000000003, 0.0964, 0.1574, ...
deg_error_Mg_50C	[0.1501, 0.275, 0.0947, 0.18660000000000002, 0...
deg_error_50C	[0.2167, 0.34750000000000003, 0.188, 0.2124, 0...
reactivity	[0.3297, 1.5693000000000001, 1.1227, 0.8686, 0...
deg_Mg_pH10	[0.7556, 2.983, 0.2526, 1.3789, 0.637600000000...
deg_pH10	[2.3375, 3.5060000000000002, 0.3008, 1.0108, 0...
deg_Mg_50C	[0.35810000000000003, 2.9683, 0.2589, 1.4552, ...
deg_50C	[0.6382, 3.4773, 0.9988, 1.3228, 0.787700000000...

Figure 25: The train set's head was transposed and the values with their category names were displayed.

The SN_filter is the signal-to-noise filter capturing which RNA molecules passed the evaluation criteria defined by the Stanford researchers. Hence, the rows with SN_filter = 0 were dropped. There were some RNAs that had quite a large amount of noise which was filtered by the SN_filter. It was found that among the 2400 records in the training set, 2096 had S/N ratio > 1, 1589 had SN_filter equal to 1 and 509 samples had S/N ration greater than 1 and SN_filter equal to 0.

Data visualizations: The bpps files were loaded. A function named ***generate_bpps_sum()*** was created to generate sum of the values under each columns in the dataframe. These sums were then appended to an array which was returned. A function named ***generate_bpps_max()*** was created to generate row-wise max value in Numpy matrix from 'bpps' folder against each molecule id. The function ***generate_bpps_nb()*** was used to check the number of non-0 bpps files in the dataframe and calculate their mean value.

These values were appended to a list which was returned. The 3 list returned from these 3 functions were added to the training and the testing set. 200 random samples were selected from the train set and the test set and their bpps sum, max row values and mean values have were generated using the above 3 functions. The 3 values from the newly added columns in the dataframe has ere utilised for visualizing the distribution of the maximum value in the bpps files, the sum of the bpps files and the mean values in the bpps files. The images of these graphs have been presented in section C.6 of the Appendix.

Data preprocessing: A list named ***target_cols[]*** was defined for storing the column names of the target columns. The different characters used in the sequences were enumerated in a dictionary. The ***preprocess_inputs()*** function was used to preprocess the values of the input columns ***`sequence', 'structure'*** and ***'predicted_loop_types'*** based on the key-value pairs in the enumeration. The pre-processed values were converted to a list which in turn was converted to an array and stored in an array named ***base_features***. This array was then transposed. This array along with the ***bpps_sum*** and ***bpps_max*** columns in the dataframe (which were converted to a list) were appended in a list named ***mylist[]***. This list was returned.

Data training: The loss functions used here was the root mean squared error and the mean column-wise root mean squared. GRU and LSTM layers were used for building this model. **GRU layer:** Based on available runtime hardware and constraints, this layer chose different implementations (cuDNN-based or pure-TensorFlow) to maximize the performance and **LSTM layer**. The model was built using the different RNN layers.

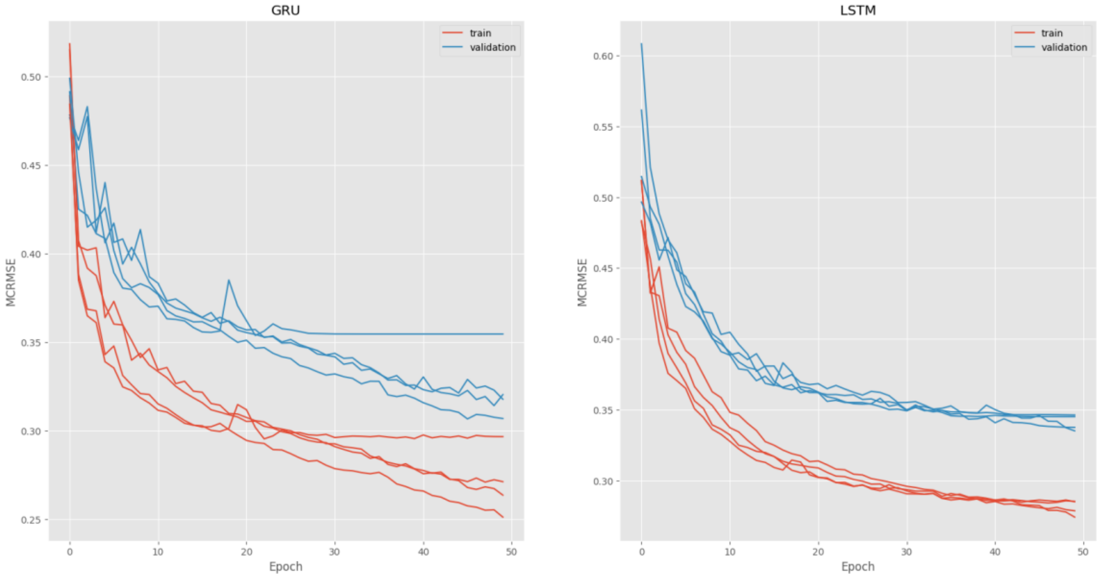


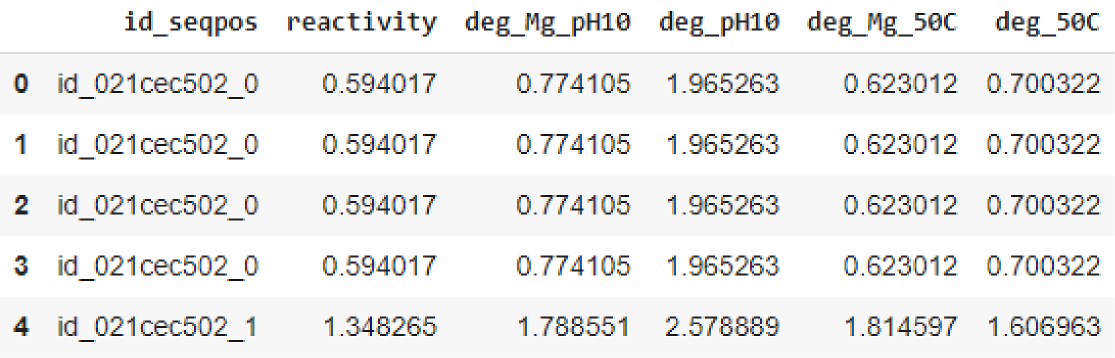
***Figure 26: 2 sets of learning curves were plotted to show the variation between the training loss and the validation loss from the GRU and LSTM layers*.**

It could observed from the predicted values that when the SN filter was not applied, then the values of the target columns were almost equal to the values in the training set. However, when the SN_filter was applied, then the values were lowered. The predicted values of the GRU and LSTM layers were stored in lists. Weights were assigned to the 2 layers (***gru_weight*** = 0.5 and ***lstm_weight*** = 0.5). Then the results from the 2 layers were blended and stored in a dataframe named ***blended_preds[].*** The formula for blending the result was as follows:

blended_preds[‘column_name’] = gru_weight * gru_preds[‘columns_name] + lstm_weight * lstm_preds[‘column_name’]

These blended results were stored in the submission.csv file.

Accuracy score and predicted values: The accuracy score at the end of training using GRU was as follows: loss: 0.2512 - val_loss: 0.3069 while the accuracy score at the end of training using LSTM was as follows: loss: 0.2851 - val_loss: 0.3451.



	id_seqpos	reactivity	deg_Mg_pH10	deg_pH10	deg_Mg_50C	deg_50C
0	id_021cec502_0	0.594017	0.774105	1.965263	0.623012	0.700322
1	id_021cec502_0	0.594017	0.774105	1.965263	0.623012	0.700322
2	id_021cec502_0	0.594017	0.774105	1.965263	0.623012	0.700322
3	id_021cec502_0	0.594017	0.774105	1.965263	0.623012	0.700322
4	id_021cec502_1	1.348265	1.788551	2.578889	1.814597	1.606963

Figure 27: Blended predicted values of the first 5 rows of the GRU + LSTM model

4.7. Description of Functionality (of the Web Interface Created)

This section provides a brief explanation on the web interface created for organizing the graphs, predictions and accuracy score retrieved after the implementation of the 3 models created in the project. The Web interface has been created using Flask and consists of four pages – Home , KnowMore , Models and Play. The description of the four pages has been provided below.

4.7.1. Home Page

The Home page displays the Welcome message to the viewer. The entire page has been divided into three sections – the header section which contains a menu on the right side of the header, a body section which displays a background image and a footer section. The menu can be used for surfing to the other pages on the website. A screenshot of the Home page has been provided below.

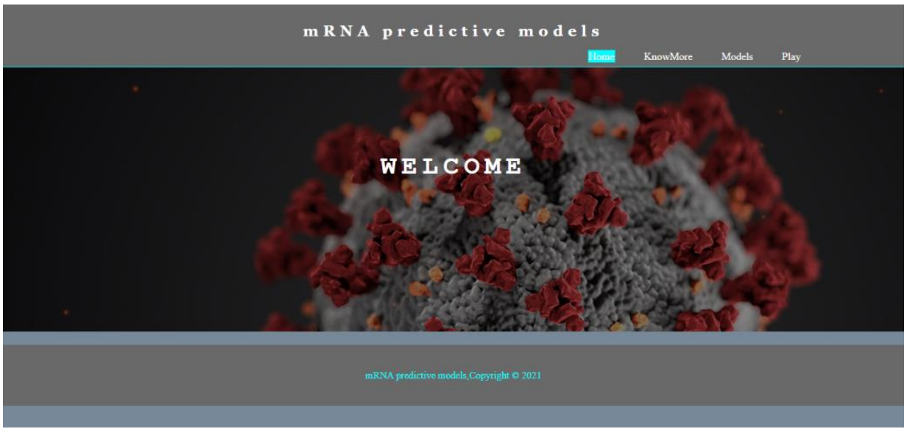
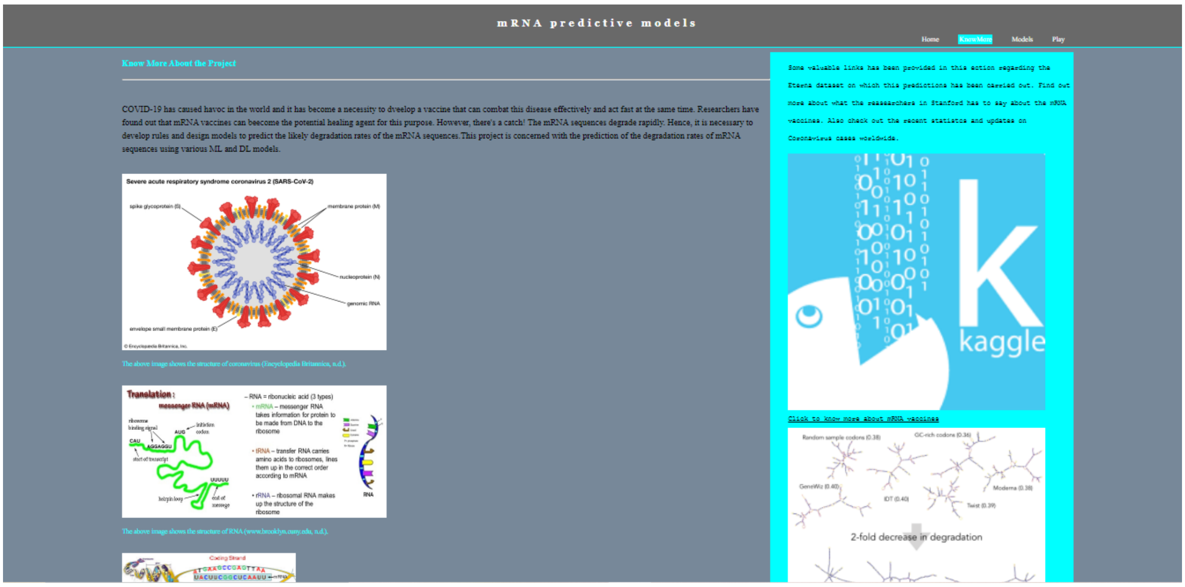


Figure 28: Home Page

4.7.2. KnowMore Page

The KnowMore Page contains detailed information on the project and project dataset. This page can also be divided into three sections – the header from which the menu on the right can be accessed, the body contains all the information and pictures related to the project and it has a footer. This page also has an aside which contains important information related to the pandemic and also contains links to other websites. This page also contains an interlink to the Models page located before the footer.



mRNA predictive models

Home KnowMore Models Play

Know More About the Project

COVID-19 has caused havoc in the world and it has become a necessity to dveelop a vaccine that can combat this disease effectively and act fast at the same time. Researchers have found out that mRNA vaccines can beecome the potential healing agent for this purpose. However, there's a catch! The mRNA sequences degrade rapidly. Hence, it is necessary to develop rules and design models to predict the likely degradation rates of the mRNA sequences.This project is concerned with the prediction of the degradation rates of mRNA sequences using various ML and DL models.

Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2)
spike glycoprotein (S)
membrane protein (M)
nucleoprotein (N)
genomic RNA
envelope small membrane protein (E)
© Encyclopaedia Britannica, Inc.

The above image shows the structure of coronavirus (Encyclopedia Britannica, n.d.).

Translation :
messenger RNA (mRNA)
ribosome binding signal
AUG
initiation codon
CAU
AGGAGGU
start of transcript
UUUUU
hairpin loop
end of message
– RNA = ribonucleic acid (3 types)
• mRNA – messenger RNA takes information for protein to be made from DNA to the ribosome
• tRNA – transfer RNA carries amino acids to ribosomes, lines them up in the correct order according to mRNA
• rRNA – ribosomal RNA makes up the structure of the ribosome
RNA

The above image shows the structure of RNA (www.brooklyn.cuny.edu, n.d.).

Some valuable links has been provided in this ection regarding the Eterna dataset on which this predictions has been carried out. Find out more about what the reasearchers in Stanford has to say about the mRNA vaccines. Also check out the recent statistcs and updates on Coronavirus cases worldwide.

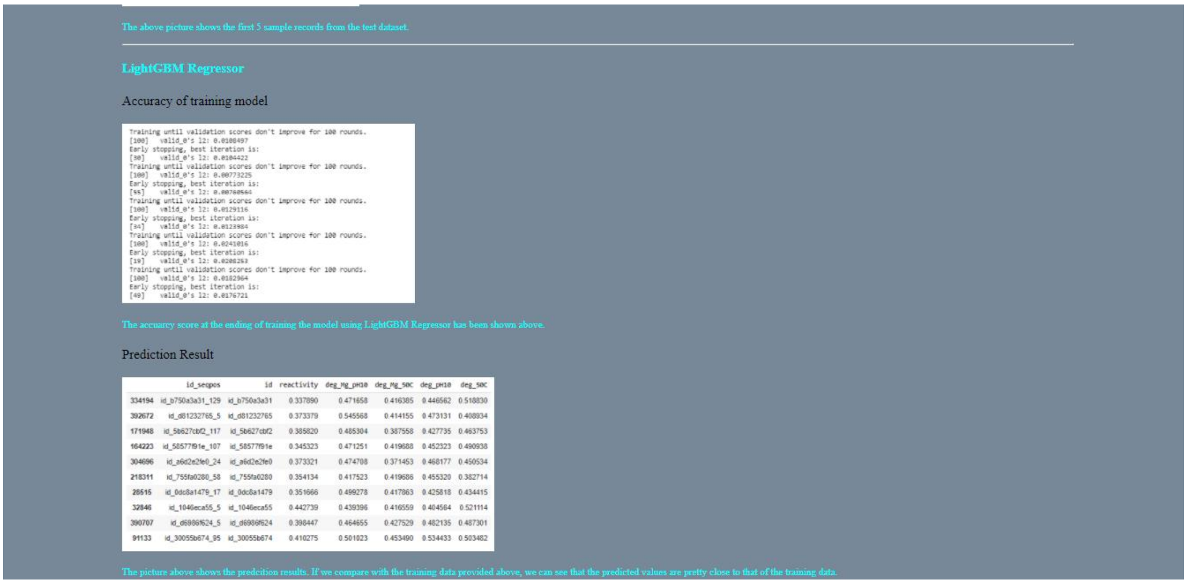
Click to know more about mRNA vaccines

Figure 29: KnowMore Page

4.7.3. Models Page

The Models Page contains screenshots of the different data visualizations like graphs that were produced during the course of building the predictive models, it also contains screenshots of all the epochs and value losses and accuracies at the end of training each model. Just like the Home page and the KnowMore page this also has 3 parts – a header section which can be used for accessing the menu, a body and a footer section.

A screenshot of the Models Page has been provided below.



The above picture shows the first 5 sample records from the test dataset.

LightGBM Regressor

Accuracy of training model

```
Training until validation scores don't improve for 100 rounds.
[100]   valid_0's l2: 0.0108497
Early stopping, best iteration is:
[30]    valid_0's l2: 0.0104422
Training until validation scores don't improve for 100 rounds.
[100]   valid_0's l2: 0.00773225
Early stopping, best iteration is:
[55]    valid_0's l2: 0.00760564
Training until validation scores don't improve for 100 rounds.
[100]   valid_0's l2: 0.0129116
Early stopping, best iteration is:
[34]    valid_0's l2: 0.0123984
Training until validation scores don't improve for 100 rounds.
[100]   valid_0's l2: 0.0241016
Early stopping, best iteration is:
[19]    valid_0's l2: 0.0208253
Training until validation scores don't improve for 100 rounds.
[100]   valid_0's l2: 0.0182964
Early stopping, best iteration is:
[49]    valid_0's l2: 0.0176721
```

The accuarcy score at the ending of training the model using LightGBM Regressor has been shown above.

Prediction Result

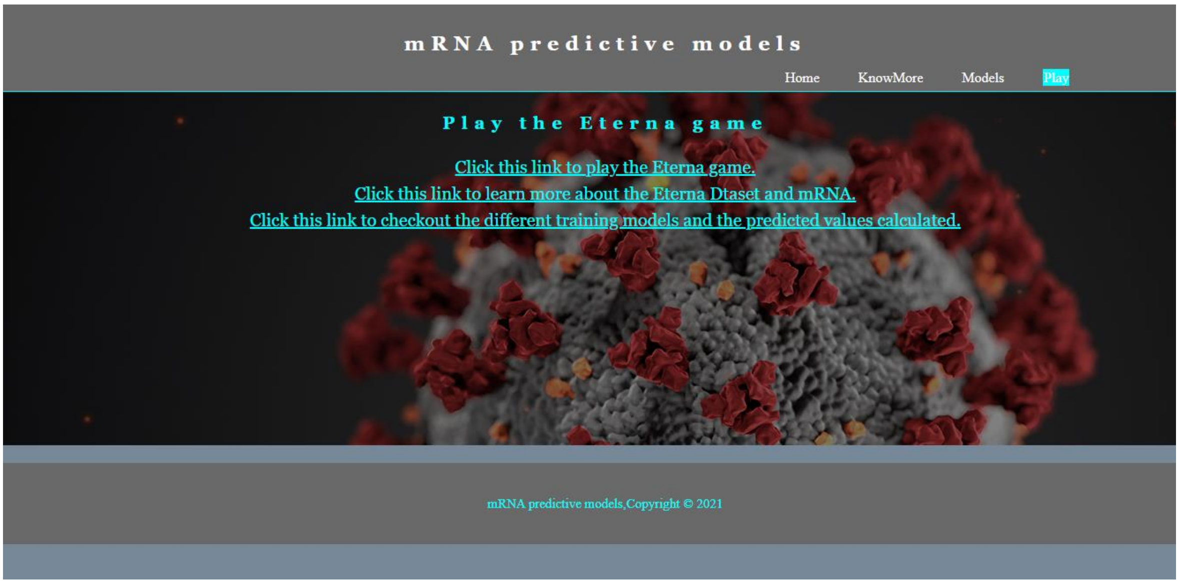
	id_seqpos	id	reactivity	deg_Mg_pH10	deg_Mg_50C	deg_pH10	deg_50C
334194	id_b750a3a31_129	id_b750a3a31	0.337890	0.471658	0.416385	0.446562	0.518830
392672	id_d81232765_5	id_d81232765	0.373379	0.545568	0.414155	0.473131	0.408934
171948	id_5b627cbf2_117	id_5b627cbf2	0.385820	0.485304	0.387558	0.427735	0.463753
164223	id_58577f91e_107	id_58577f91e	0.345323	0.471251	0.419688	0.452323	0.490938
304696	id_a6d2e2fe0_24	id_a6d2e2fe0	0.373321	0.474708	0.371453	0.468177	0.450534
218311	id_755fa0280_58	id_755fa0280	0.354134	0.417523	0.419686	0.455320	0.382714
28515	id_0dc8a1479_17	id_0dc8a1479	0.351666	0.499278	0.417863	0.425818	0.434415
32846	id_1046eca55_5	id_1046eca55	0.442739	0.439396	0.416559	0.404564	0.521114
390707	id_d6986f624_5	id_d6986f624	0.398447	0.464655	0.427529	0.482135	0.487301
91133	id_30055b674_95	id_30055b674	0.410275	0.501023	0.453490	0.534433	0.503482

The picture above shows the predcition results. If we compare with the training data provided above, we can see that the predicted values are pretty close to that of the training data.

Figure 30: A section of the Models Page

4.7.4. Play Page

The Play page contains links to the Eterna game and interlinks to the KnowMore page and the Models page. It has 3 parts – a header from which the menu can be accessed, a body and a footer.



mRNA predictive models

Play the Eterna game

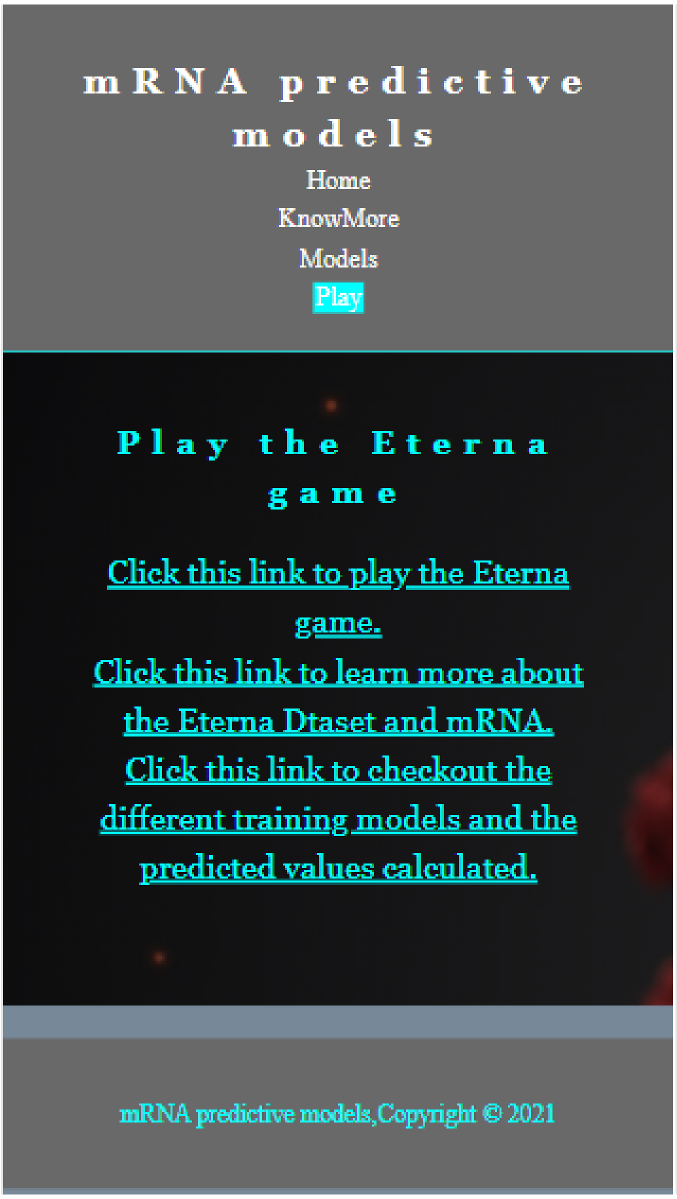
Click this link to play the Eterna game.
Click this link to learn more about the Eterna Dtaset and mRNA.
Click this link to checkout the different training models and the predicted values calculated.

mRNA predictive models,Copyright © 2021

Figure 31: Play Page

4.7.5. Other features

A CSS style sheet was used for styling the website and creating the templates for the website. A hover feature was added to the website so that everytime the mouse is hovered on the menu options, the option gets highlighted. The website created is responsive across various devices. Screenshot has been provided below to show the responsiveness of the website.



mRNA predictive models

Home
KnowMore
Models
Play

Play the Eterna game

Click this link to play the Eterna game.
Click this link to learn more about the Eterna Dtaset and mRNA.
Click this link to checkout the different training models and the predicted values calculated.

mRNA predictive models,Copyright © 2021

Figure 32: Play page as it would appear on a mobile device

4.8. Contingency Plans

Table 1: Contingency Plan in case of failures of while implementing the Learning Model

Test ID	Test Description	Risks	Mitigation Plan
1.	Load the datasets by reading them through Pandas.	Files does not get loaded.	1.1. Check if the drive is mounted. 1.2. Try uploading the files on Colab during runtime and reading them.
2.	Plot different graphs for data visualizations.	Graphs does not get displayed.	2.1. Use %matplotlib notebook. This will lead to interactive plots embedded within the system. 2.2. Use %matplotlib inline. This would lead to static images of the plot embedded into the notebook. 2.3. If working on Jupyter then re-install matplotlib from source using setup.py and execute the get_backend() function.
3.	Save the predicted results to a .csv file.	Results does not get saved.	3.1. Mount the drive again. 3.2. Export the dataframe as .csv with Pandas. Read and put it in a variable. Write the result of this reading in another file and rename it. 3.3. Export the file to the local machine. Remove '/content' from the directory path. Check the current working directory by running the pwd command.
4.	Describe the dataset thoroughly using graphs for showing the distribution of different features present in the dataset and the relationships among the different features.	Dataset not described or visualized properly.	4.1. In the early stage of the project create graphs taking into account all the features present in the entire dataset. 4.2. Describe each feature including the ones having null values.
5.	Clean the dataset.	Dataset ending up with a lot of noise.	5.1. Carefully examine the dataset before training.

			5.2. Make the training dataset larger 5.3. Resample with different ratios to get rid of imbalanced data.
6.	Implement Feature Engineering on the cleaned dataset.	Insufficient implementation of Feature Engineering done on the dataset.	6.1. Identify the indicator and interaction features. 6.2. Perform feature representation. 6.3. Bring in external data.
7.	Generalize the model well.	Chances of dataset getting overfitted/ underfitted.	7.1. Use cross validation methods. 7.2. Train learning algorithm iteratively. 7.3. Use regularization techniques. 7.4. Use ensemble methods like bagging and boosting.
8.	Train the model and end up with a good accuracy score.	Risk of the model not converging to the global maxima.	8.1. Use bigger batch sizes that will create smoother updates by updating the learning rate decay.

Table 2: Contingency Plan in case of Failures while implanting the Web Interface

Test ID	Test Description	Risks	Mitigation Plan
1.	The website gets loaded with the style templates.	The website does not get loaded when run on Colab.	1.1. Upload the static folder containing the style .css file and the template file on the drive during run time. 1.2. Upload the template and static folder beforehand on Drive and mount the drive. Read the files in the folders using the file path.
2.	Check if all the web pages load with the style sheets.	The web pages load but only the Home page retains its style.	2.1. Check that the style sheet is linked to all the web pages. 2.2. If the style sheet is placed in a different folder, make sure to

			put the folder's name in the directory path.
3.	Check that all the images on the web pages are in perfect orientation.	Images on the web pages are not perfectly oriented and scroll bars appear.	3.1. Try to resize the images with the necessary widths and heights to fit them within that page.
4.	Check that all the links on the web pages are working fine.	The links on the web pages do not work.	4.1. Check that the URL for the links placed in the href are placed within double quotes. 4.2. Check that the entered URL is correct and actually exists.

4.9. Conclusion

This chapter gave details on the implementation procedures and designs that were used for the successful completion of the project. The solution was designed keeping in mind the limitations identified in Chapter 2 and the new ideas generated in Chapter 3. The created solution comprised of the successful implementation of three learning models for predicting mRNA degradation rates and a web interface for organizing all the results and predictions obtained from the models.

Chapter 5 – Results and Discussion

This chapter is concerned with the evaluation of the implemented functionality which had been described thoroughly in Chapter 4. This Chapter will provide detailed test plan and contingency plans for testing every single unit of the three models created – The LightGBM Regressor model, the WaveNet Model and the LSTM + GRU model and also the Web Interface. The testing plans are based on the strategy which deconstructs from each of the SMART objectives. The results of the tests will be summarized at the end of this chapter.

5.1. Test Plan for testing the functionality of the different models.

Table 3: LightGBM Regressor Model

Test ID	Test Description	Expected Outcome	Actual Outcome
1.	Load the train and test .json files and the submission .csv file by reading through Pandas.	Files get loaded.	As expected.
2.	Transform the train dataset into a dataframe.	Dataset transformed into a dataframe.	As expected.
3.	Plot histograms for showing the distribution of all the features present in the training set.	Histograms get displayed.	As expected.
4.	Transform the test dataset into a dataframe.	Dataset transformed into dataframe.	As expected.
5.	Load BPPS files by using os.listdir function.	BPPS files gets loaded.	As expected.
6.	Plot kdeplots for showing the distribution of Signal to Noise ratio throughout train set.	Kdeplots get displayed.	As expected.
7.	Plot bar plots for showing the distribution of sequence length throughout test set.	Histograms get displayed.	As expected.
8.	Plot histograms for displaying the mean values of the target features in the training set.	Histograms get displayed.	As expected.
9.	Save the mean values to the submission.csv file.	Data gets saved.	As expected.
10.	Display plots for showing the distribution of the target features in the training set.	Graphs get displayed.	As expected.
11.	Plot pairplots for showing the relationships among the mean values of the different features.	Pair plots displayed.	As expected.

12.	Build and train the model.	Models gets trained with a high accuracy score.	As expected.
13.	Save results to the submission .csv file.	Results gets saved.	As expected.
14.	Plot regplots for checking the multi collinearity present in the training sample.	Regplots get displayed.	As expected.

Table 4: WaveNet Model

Test ID	Test Description	Expected Outcome	Actual Outcome
1.	Load the train, test .json files and the submission .csv file.	Files get loaded.	As expected.
2.	Preprocess and transform the dataset to a dataframe.	Dataset gets transformed.	As expected.
3.	Transform training and testing features to a 3D matrix.	Training and testing features get transformed.	As expected.
4.	Build the model, add the custom loss function, add hidden layers, dense layers, optimizers and the WaveNet layer. Add evaluation metrics and train the model.	Model gets trained with a high accuracy score.	As expected.
5.	Transform the results to the correct format and save the predicted results to the submission .csv file.	Results saved successfully.	As expected.

Table 5: GRU + LSTM Model

Test ID	Test Description	Expected Outcome	Actual Outcome
1.	Load the train and test .json files and the submission .csv file.	Files get loaded.	As expected.
2.	Plot kdeplots and bar plots for showing the Signal/Noise distribution and the Signal/Noise filter distribution respectively.	Graphs get displayed.	As expected.
3.	Load BPPS files and display them.	Files get displayed.	As expected.
4.	Plot kdeplots for showing the distribution of bpps_max, bpps_sum and bpps_nb.	Files get loaded and displayed.	As expected.
5.	Preprocess the data.	Data gets pre-processed.	As expected.
6.	Build the models, add hidden and dense layers and train using GRU and LSTM layers.	Models get trained with a high accuracy score.	As expected.
7.	Plot learning curves for showing the training and validation losses.	Graphs get displayed.	As expected.

8.	Calculate the blended results from the GRU and LSTM layers and save the results to thee submission .csv file.	Results get saved.	As expected.

5.2. *Table 6: Test Plan for testing the different components of the Web Interface created*.

Test ID	Test Description	Expected Outcome	Actual Outcome
1.	The flask app is run on Google Colab.	The Welcome page gets loaded.	As expected.
2.	The menu is used for switching to the KnowMore page.	The KnowMore page gets loaded.	As expected.
3.	The links on the aside section of the KnowMore page are used for switching to other websites.	The links open up other web pages.	As expected.
4.	The link at the bottom of the KnowMore page is used for switching to the Models page.	The Models page gets loaded.	As expected.
5.	The menu is used for switching to the Models page.	The Models page opens up.	As expected.
6.	The menu is used for switching to the Play page.	The Play page gets loaded.	As expected.
7.	The links on the Play page are used for switching to the different websites.	The links open up different web pages.	As expected.

These tests checked the performance of every single unit of the code and the website produced. It can be noticed from the test plan that all the components of the project successfully passed the tests. This chapter also provided contingency plans which would serve as reference in case a component failed to pass a test or yielded an alternate outcome rather than the expected. The tests performed showed that the models and website produced at the end of the project were successful in meeting the functional and non-functional requirements of the project.

5.3. Conclusion

This chapter thoroughly described the tests that were used for determining the success of the implemented solution. The test plans were designed keeping in mind the SMART objectives that were defined for achieving the aim of the project. To draw a conclusion from the test plans stated above, it can be inferred that all the objectives defined at the start of the project have been successfully completed and no errors have been found during the execution of the code.

Chapter 6 – Conclusion and Future Work

6.1. Conclusion

The current situation of COVID-19 seems to worsen with every passing day. Globally, new COVID-19 cases rose for the eighth consecutive week, with over 5.2 million new cases reported in the last week. The number of new deaths increased for the fifth consecutive week, increasing by 8% compared to last week, with over 83 000 new deaths reported. While all regions except the European Region reported an increase in incident cases in the last week, the largest increase continues to be reported by the South-East Asia Region, largely driven by India, followed by the Western Pacific Region (www.who.int, n.d.). There is a huge demand for an effective vaccine which can be replicated over a large scale and which would also be cost effective. This project has tried to take an approach to help the medical scientists and researchers in the creation of a successful mRNA vaccine. This project has generated a novel solution which can be taken up by the stakeholders (data scientists at Stanford) for designing and creating models that will enable them to mitigate the problem of the degradation of mRNA bases. The solution produced has three models built using different Deep Learning methods that aim to produce minimal values during the prediction of the target features. This project has also produced a website that has organized the results, predictions and data visualizations generated from the different models – this would serve as a valuable account to the biochemists and data scientists who intend on improving on the existing models and produce a better solution.

6.2. Future Work

The current solution is confined to the creation of the models that check the rate of degradation of mRNA bases and produce minimal values for the target features. However, none of the models check if the predicted values of the target features would result in the formation of a stable RNA molecule. Although the creation of stable mRNA molecules remains difficult, datasets of sequences and corresponding information are becoming more popular and widely available. Through the use of deep learning architectures, reasonable predictions of structural features can be obtained, as demonstrated in this manuscript. The usage of such techniques has the capacity to increase the speed and efficiency of mRNA vaccine discovery and has further implications in other related fields of research. In order to improve the performance of the model a simple binary classification system should be created to predict whether a RNA molecule is stable at the end of every model. This would minimize the False Negative Rate and finally, the model needs to be predicted on longer RNA sequences to observe how it impacts the accuracy of such methods (Singhal, n.d.). Another major improvement that could be executed in the future would be to combine the three Deep Learning methods used in this project, that is, LightGBM Regressor, WaveNet, GRU and LSTM within a single model and blend the results of each of these models to produce a single concrete predicted dataset.

In terms of the web interface, it could be improved by adding a functionality that would enable a user to upload a dataset containing mRNA sequences and select a model for generating real time predicted results from the data. Another major improvement could be added by adding a functionality that would generate real time data sequences from the mRNA sequence produced on the Eterna game and enable the user to select a model for predicting the results of the model. Accounts can be created on the website in the future that would enable other data scientists to add on their solutions and results on the site and discuss about future improvements.

6.3. LSEPI

This project was created with due care and diligence, keeping in mind the client or company’s best interests at all times. The project was built with personal and collective responsibility for while maintaining discretion and ethical standards (Bcs.org, 2019).

The successful completion of this project does have a significant social impact. It might lead to the production of vaccines that can be supplied to the masses to cure COVID-19. However, it is important to note that the models generated at the end of this project would not be deployed directly for the production of the mRNA vaccines. It is important to note that the models created in this project was just concerned with checking the degradation rates of the mRNA molecules and not with checking their stability. Also, they were not concerned with the generation of an mRNA vaccine.

The models devised in this project would have to undergo thorough checking by the stakeholders (data scientists at Sandford) before it is termed as the perfect solution to the problem statement. This model along with other prediction models devised by data scientists would be tested on a 2nd generation of RNA sequences devised by Eterna players. The models would be scrutinized thoroughly by the members of the Eterna community led by Professor Rhiju Das and his team at Stanford's Medical School. The team might end up taking the best bits from each model to devise another model that would give them the best solution for their problem. Moreover, the stability of the predicted values of these models would be checked by the scientists.

This project aimed to help the medical team at Stanford to accelerate their research in developing ways to stabilize the RNA molecules. The model played a part in the development of the vaccine but did not devise a standalone vaccine itself (Kaggle, n.d.).

Based on the BCS standards, this project was built with integrity and to show competence but it is evident that it is not the ultimate solution to the problem statement, that's why it would be possible to improve on the project in the future. Future improvements on this project would be made keeping in mind the BCS Code of Conduct. Perfect knowledge, skills and resources would be used for undertaking tasks and completing them (Bcs.org, 2019). This project would also support interested IT colleagues and other members in their growth both personally and professionally (Bcs.org, 2019).

6.4. Synoptic Assessment

This project had benefitted me in a number of ways. I have been able to create Deep Learning models using some novel techniques that I had never worked with before. At the start I had very little knowledge on how LSTMs or GRUs worked. However, during the course of the project I was able to gain an in-depth understanding on these topics. Moreover, independent research work helped me to learn about the WaveNet model. I had to carry on some extensive research work and read a bunch of research papers to figure out the fact that WaveNet could be used for regression problems apart from audio detection techniques. Finally, I was able to successfully implement it in building one of the learning models. A lot of techniques that I have implemented in this project like data visualization techniques was also covered throughout the AI module in this and the previous academic year which was an additional help.

This was the first time that I had worked on such a complex AI problem with a large dataset. The project had incited my interests in the fields of vaccines and I hope to work on improving this project in the future keeping in mind all the points that I had mentioned in the section of future work. Lastly, I would like to mention that the successful completion of this project has uplifted my spirits and have encouraged me to continue working on my dreams of becoming an AI scientist someday.

References

- WHO (2021). WHO COVID-19 dashboard. [online] covid19.who.int. Available at: https://covid19.who.int/.
- www.pbs.org. (n.d.). Can Scientists Use RNA to Create a Coronavirus Vaccine? [online] Available at: https://www.pbs.org/wgbh/nova/video/rna-coronavirus-vaccine/.
- kaggle.com. (n.d.). OpenVaccine: COVID-19 mRNA Vaccine Degradation Prediction. [online] Available at: https://www.kaggle.com/c/stanford-covid-vaccine/overview/description [Accessed 18 Apr. 2021].
- Conger, A.K. (2020). Stanford biochemist works with gamers to develop COVID-19 vaccine. [online] Scope. Available at: https://scopeblog.stanford.edu/2020/05/20/stanford-biochemist-works-with-gamers-to-develop-covid-19-vaccine/.
- Pardi, N., Hogan, M.J., Porter, F.W. and Weissman, D. (2018). mRNA vaccines — a new era in vaccinology. Nature Reviews Drug Discovery, [online] 17(4), pp.261–279. Available at: https://www.nature.com/articles/nrd.2017.243.
- ServickDec. 16, K., 2020 and Pm, 1:25 (2020). Messenger RNA gave us a COVID-19 vaccine. Will it treat diseases, too? [online] Science | AAAS. Available at: https://www.sciencemag.org/news/2020/12/messenger-rna-gave-us-covid-19-vaccine-will-it-treat-diseases-too.
- Yang, Y., Xiao, Z., Ye, K., He, X., Sun, B., Qin, Z., Yu, J., Yao, J., Wu, Q., Bao, Z. and Zhao, W. (2020). SARS-CoV-2: characteristics and current advances in research. Virology Journal, 17(1).
- Zhu, F.-C., Li, Y.-H., Guan, X.-H., Hou, L.-H., Wang, W.-J., Li, J.-X., Wu, S.-P., Wang, B.-S., Wang, Z., Wang, L., Jia, S.-Y., Jiang, H.-D., Wang, L., Jiang, T., Hu, Y., Gou, J.-B., Xu, S.-B., Xu, J.-J., Wang, X.-W. and Wang, W. (2020). Safety, tolerability, and immunogenicity of a recombinant adenovirus type-5 vectored COVID-19 vaccine: a dose-escalation, open-label, non-randomised, first-in-human trial. The Lancet, [online] 0(0). Available at: https://www.thelancet.com/journals/lancet/article/PIIS0140-6736(20)31208-3/fulltext.
- Rachlin, E. and Watson, M. (2017). mRNA Vaccines: Disruptive Innovation in Vaccination. [online] . Available at:

https://www.modernatx.com/sites/default/files/RNA_Vaccines_White_Paper_Moderna_050317_v8_4.pdf.

- Steele, Kim, Choe, Nicol, Oguri, Sperberg, Huang, Eterna Participants, Das (2020). Theoretical basis for stabilizing messenger RNA through secondary structure design. [online]. Available at: https://www.biorxiv.org/content/10.1101/2020.08.22.262931v1.full.pdf.
- Crommelin, D.J.A., Anchordoquy, T.J., Volkin, D.B., Jiskoot, W. and Mastrobattista, E. (2020). Addressing the Cold Reality of mRNA Vaccine Stability. Journal of Pharmaceutical Sciences.
- kaggle.com. (n.d.). OpenVaccine: COVID-19 mRNA Vaccine Degradation Prediction. [online] Available at: https://www.kaggle.com/c/stanford-covid-vaccine/data.
- Coursera. (n.d.). COVID-19 mRNA Vaccine Degradation Prediction. [online] Available at: https://www.coursera.org/projects/covid-19-mrna-vaccine-degradation-prediction [Accessed 20 Apr. 2021].
- shadowburning. (2020). TCN+LSTM+GRU. [online]. Available at: https://www.kaggle.com/shadowburning/tcn-lstm-gru.
- gagankarora. (2020). Transfer Learning+gru+lstm+Basics 101. [online]. Available at: https://www.kaggle.com/gagankarora/transfer-learning-gru-lstm-basics-101.
- fernandoramacciotti. (2020). LGBM+BPP features. [online]. Available at: https://www.kaggle.com/fernandoramacciotti/lgbm-bpp-features.
- vbmokin. (2020). GRU & LSTM mix & custom loss – tuning by 3D visual. [online]. Available at: https://www.kaggle.com/vbmokin/gru-lstm-mix-custom-loss-tuning-by-3d-visual.
- ruko. (2020). OpenVaccine-GRU+LSTM. [online]. Available at: https://www.kaggle.com/rkuo2000/openvaccine-gru-lstm.
- anzhemeng. (2020). OpenVaccine-GRU+LSTM(with custom loss function. [online]. Available at: https://www.kaggle.com/anzhemeng/openvaccine-gru-lstm-with-custom-loss-function.
- omarvivas. (2020). Ensemble CB LightGBM XGB Openvaccine v2. [online]. Available at: https://www.kaggle.com/omarvivas/ensemble-cb-lightgbm-xgb-openvaccine-v2.
- Vandewiele, G. (2020). Predicting mRNA Degradation using GNNs and RNNs in the Search for a COVID-19 Vaccine. [online] Medium. Available at:

https://towardsdatascience.com/predicting-mrna-degradation-using-gnns-and-rnns-in-the-search-for-a-covid-19-vaccine-b3070d20b2e5.

- Balaban, J. (2019). How WaveNet Works. [online] Medium. Available at: https://towardsdatascience.com/how-wavenet-works-12e2420ef386 [Accessed 21 Apr. 2021].
- Wikipedia. (2021). WaveNet. [online] Available at: https://en.wikipedia.org/wiki/WaveNet [Accessed 21 Apr. 2021].
- Liu, T., Zhang, Y., Qi, L., Dong, J., Lv, M. and Wen, Q. (2019). WaveNet: learning to predict wave height and period from accelerometer data using convolutional neural network. IOP Conference Series: Earth and Environmental Science, 369, p.012001.
- Mandot, P. (2018). What is LightGBM, How to implement it? How to fine tune the parameters? [online] Medium. Available at: https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc.
- research.google.com. (n.d.). Colaboratory – Google. [online] Available at: https://research.google.com/colaboratory/faq.html#:~:text=Colaboratory%2C%20or%20%E2%80%9CColab%E2%80%9D%20for.
- Digraje, N. (2019). Colab — The coolest Gift to data scientists from Google. [online] Medium. Available at: https://medium.com/datasciencebuddy/colab-the-coolest-gift-to-data-scientists-from-google-817487acb8de [Accessed 22 Apr. 2021].
- Wikipedia Contributors (2018). Long short-term memory. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Long_short-term_memory.
- Rs-online.com. 2021. Predicting weather using LSTM. [online] Available at: <https://www.rs-online.com/designspark/predicting-weather-using-lstm> [Accessed 22 April 2021].
- Wikipedia Contributors (2019). Gated recurrent unit. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Gated_recurrent_unit [Accessed 11 May 2019].
- 2021. [online] Available at: <https://www.researchgate.net/publication/339094287_Prediction_of_Driver%27s_Attention_Points_Based_on_Attention_Model> [Accessed 22 April 2021].

- Medium. 2021. Understanding WaveNet architecture. [online] Available at: <https://medium.com/@satyam.kumar.iiitv/understanding-wavenet-architecture-361cc4c2d623> [Accessed 22 April 2021].
- AIM Consulting Group, LLC. (2016). Finding Strong Scrum Resources in a Less Than Standardized World. [online] Available at: https://aimconsulting.com/insights/finding-strong-scrum-resources-less-standardized-world/ [Accessed 23 Apr. 2021].
- Air. (n.d.). Python logo / Air. [online] Available at: https://air.inc/blog/logos/Python-Logo-slack-theme [Accessed 23 Apr. 2021].
- www.deviantart.com. (n.d.). HTML 5 and CSS 3 Logo PSD by webdesignerbag on DeviantArt. [online] Available at: https://www.deviantart.com/webdesignerbag/art/HTML-5-and-CSS-3-Logo-PSD-297522879 [Accessed 23 Apr. 2021].
- kaggle.com. (n.d.). OpenVaccine: COVID-19 mRNA Vaccine Degradation Prediction. [online] Available at: https://www.kaggle.com/c/stanford-covid-vaccine/data.
- www.who.int. (n.d.). Weekly epidemiological update on COVID-19 - 20 April 2021. [online] Available at: https://www.who.int/publications/m/item/weekly-epidemiological-update-on-covid-19---20-april-2021.
- Bcs.org. (2019). BCS Code of Conduct. [online] Available at: https://www.bcs.org/membership/become-a-member/bcs-code-of-conduct/.

Appendices

Appendix A

A.1

No.	Objectives	Specific	Measurable	Achievable	Relevant	Time-Bound
1.	Describe the Eterna dataset.	A detailed explanation of the different features present in the dataset and how they contribution towards the degradation of the RNA molecules.	Visual representation through graphs and plots can be used to judge the importance of a particular feature in determining the degradation rate.	There are evidences from existing solutions on this problem where the dataset has been thoroughly described. This proves that it is possible to achieve this particular objective.	Data description and graphical visualization will serve to identify the main features needed to be taken into consideration for training the data.	16.1.2021
2.	Create a 'cleaned' version of the dataset.	Elimination of those features which plays little or no role in determining the degradation rate.	Visualization of the dataset after deleting the features. Graphs can be used to determine the relevance of the current features with respect to cell degradation.	Evidences from previous solutions on the problem proves that it is possible to achieve this objective.	Cleaning the dataset will play an important role in determining the accuracy of the trained model.	18.1.2021
3.	Create a 'modified' version of the cleaned dataset.	Modifying the structure of the dataset to fit the training model	Line graphs can be used to determine what impact the newly engineered features are having on the target variable.	All the previous solutions to this problem has used a modified version of the dataset for training which shows this objective can be accomplished.	Pruning the dataset plays a very important role in training the data. A perfectly pruned dataset can be used to create a perfect training model. This is what we are	2.2.2021

					trying to achieve here.	
4.	Generalize the learning model.	Attempting to create a model that will give a high accuracy score when trained on most ML/ DL models.	An average high accuracy score on different ML and DL scoring models will determine that successful creation of a generalized prediction model.	Previous solutions to this problem shows that it is possible to get a generalized model with a high accuracy score on various ML and DL scoring models.	This is the ultimate goal of this particular project.	15.2.2021
5.	Create an user interface for the programmer.	Developing a web app using Flask, HTML and CSS. This can enable the programmer to browse through and check the visualizations, predicted results and accuracy scores of the different ML and DL models created in the project.	The amount of time it takes for a new user to get used to the environment determines how good/ bad the interface is.	There are not many evidences showing the development of an user interface. Hence, this objective possess a high risk of failure.	This objective steps up the project to a new level. This web app may only be helpful to the coders. However, it is not essential in determining the main aim of this project.	1.3.2021

Table 7: The Table above shows the SMART Objectives.

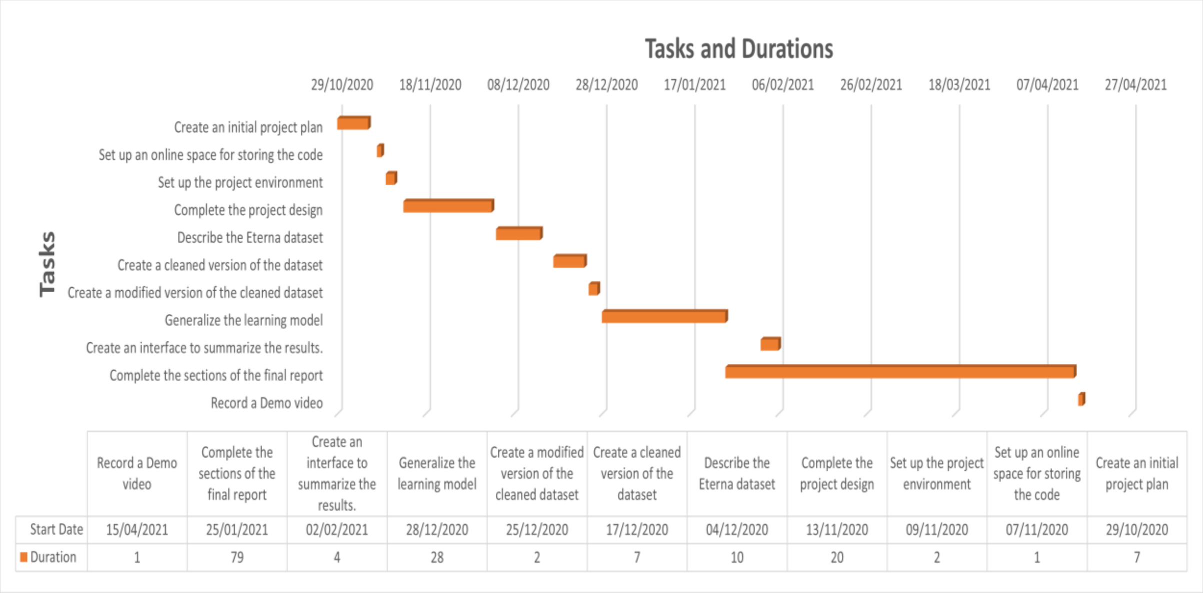
A.2

Tasks	Deliverables
1. Create a project plan before the start of the project: 1.1 Research on the project topic chosen. 1.2 Identify the aims and objectives of the project. 1.3 Identify the potential risks of the project. 1.4 Determine a prototype for completing the project. 1.5 Assemble the points mentioned above in a document. 1.6 Get the document reviewed by the supervisor.	Project Planning Document
2. Set up an online space for storing the code.	No Deliverables
3. Set up the project Environment: 3.1 Watch tutorials on how to use Google Colab. 3.2 Download the Eterna dataset from Kaggle. 3.3 Install the packages and libraries necessary.	No Deliverables

4. Complete Project Design: 4.1 Identify the main use cases and classes in the project. 4.2 Create sequence diagram to depict the flow of information among the primary objects in the project. 4.3 Identify the operations and attributes of the objects.	Design section in the final report.
5. Describe the Eterna dataset: 5.1 Peek at the data. 5.2 Check the dimensions of the dataset. 5.3 Check class distribution and correlation among attributes. 5.4 Use Pandas to create graphs illustrating the relationships among the different features.	No Deliverables
6. Create a ‘cleaned’ version of the dataset: 6.1 Delete columns that contain single values. 6.2 Drop irrelevant columns. 6.3 Delete columns with unique values.	No Deliverables
7. Create a modified version of the cleaned dataset: 7.1 Convert the columns having textual data to numeric values using encoding. 7.2 Generate dummy values for fields with missing data. 7.3 Normalize the dataset.	No Deliverables
8. Generalize the learning model: 8.1 Check the accuracy score across the various ML/DL models. 8.2 Keep training the model till an average high accuracy score is achieved across various ML and DL scoring models.	No Deliverables
9. Create an UI for the programmers: 9.1 Watch tutorials on how to develop GUI on Python platforms like Jupyter.	No Deliverables
10. Complete the other sections of the final report.	Final Report
11. Record a video demonstrating the project.	Demo Video

Table 8: The table above shows the Tasks and Deliverables.

A.3



	Record a Demo video	Complete the sections of the final report	Create an interface to summarize the results.	Generalize the learning model	Create a modified version of the cleaned dataset	Create a cleaned version of the dataset	Describe the Eterna dataset	Complete the project design	Set up the project environment	Set up an online space for storing the code	Create an initial project plan
Start Date	15/04/2021	25/01/2021	02/02/2021	28/12/2020	25/12/2020	17/12/2020	04/12/2020	13/11/2020	09/11/2020	07/11/2020	29/10/2020
■ Duration	1	79	4	28	2	7	10	20	2	1	7

Figure 33: Gantt Chart

Appendix B

B.1

No.	Author/ Year	Title	Aim/Purpose	Summary of Results
1.	Pardi et al.,2018	mRNA vaccines — a new era in vaccinology. Nature Reviews Drug Discovery	To provide a detailed overview of mRNA vaccines and address future challenges and how these challenges can be terminated to ensure the vaccine's success in the world of medicine.	The study gave evidences to show that mRNA vaccines have experienced a burst in basic and clinical research. It hinted that further research is needed to determine how animal species respond to mRNA vaccines. Also, it talked about how different companies are trying to commercialize the production of the vaccine.
2.	ServickDec. 16, 2020 and Pm, 2020	Messenger RNA gave us a COVID-19 vaccine. Will it treat diseases, too?	To summarize the success of mRNA vaccine in clinical trials for COVID-19, why they have created a buzz, what diseases they can combat and their drawbacks.	This study provided evidences that showed mRNA vaccines can be produced easily over a large scale and that they are capable of combating a number of diseases other than COVID-19. The study further analysed the adverse effects of the vaccine when administered repeatedly to a person. It suggested reduction in the amount of protein body that is made from a dose of mRNA and the reduction of the frequency and the size of the dose.
3.	Yang et al., 2020	SARS-CoV-2: characteristics and current advances in research.	To provide a comprehensive survey of the latest coronavirus and provide further information on the cardiovascular diseases that might affect a COVID-19 patient and suggestion for their treatment.	The study summarized the differences between the SARS-CoV-2 virus and SARS-CoV with regards to classification, amino acid composition and protein structure, epidemiological and pathological characteristics. The pathogenic mechanisms of SARS-CoV-2 has been summarized here.
4.	Zhu et al., 2020	Safety, tolerability, and immunogenicity of a recombinant	To assess the safety, tolerability and immunogenicity of a recombinant	The study summarized the aftermath of the first in human clinical trial of Ad5 vectored COVID-19 vaccine. It showed

		adenovirus type-5 vectored COVID-19 vaccine: a dose-escalation, open-label, non-randomised, first-in-human trial.	adenovirus type-5 vectored by COVID-19 vaccine.	that the adverse effects caused by the vaccine were fever, fatigue, headache and muscle pain; evidence of high reactogenicity profile at a high dose. The study further showed the specific roles of specific antibodies or T cells in building effective protection were not known. Results of this study were confined to small size of the cohort, the short duration of follow up and absence of a randomised control group.
5.	Rachlin and Watson, 2017	mRNA Vaccines: Disruptive Innovation in Vaccination.	To show the remarkable successes of traditional vaccines while addressing their drawbacks and throw light on why there is still room left for innovation in vaccine research, development, manufacturing and delivery.	This study summarized the challenges faced by traditional vaccines (lengthy preparation process and demonstrable efficacy empirically). It showed that the vaccines required special facilities for production and deployment and highlighted the advantages of nucleic vaccines like _ rapid discovery stage, standardized production, ability to mimic viral infections. The study also showed how mRNA vaccines combined the advantages of DNA vaccines while overcoming the risk of DNA integration.
6.	Steele et al., 2020	Theoretical basis for stabilizing messenger RNA through secondary structure design.	To present simple calculations for estimating RNA stability against hydrolysis and present a model that combines the average unpaired probability of an mRNA/ AUP with overall rate of hydrolysis.	The study compared the optimization of AUP by conventional mRNA design methods algorithm, a new Monte Carlo tree search named Ribo Tree and crowdsourcing through the OpenVaccine Challenge on the Eterna platform for characterizing the stabilization achievable through structure design. Results showed that the Eterna and the Ribo Tre had significantly lower AUP while

				maintaining a large diversity of sequence and structural features correlating with translation, biophysical size and immunogenecity. Evidences showed that further research can help increase the half-life of vitro mRNA immediately.
7.	Crommelin et al., 2020	Addressing the Cold Reality of mRNA Vaccine Stability. Journal of Pharmaceutical Sciences.	To describe company proposals for the storage of mRNA vaccines and review of mRNA vaccine candidates on the pharmaceutical stability. The study aimed to provide an account on the attempts made to improve the vaccine's stability and analytical techniques used for this purpose including regulatory guidelines covering product characterization and storage stability.	This study summarized that the systematic approaches for finding key physiochemical degradation mechanisms of formulated mRNA vaccine candidates are currently lacking. Design of optimally stable mRNA vaccines and their storage conditions must be given the top priority by the pharmacies. The study provided evidences to show mRNA vaccines can adapt to new emerging infectious diseases other than COVID-19.
8.	Singhal, n.d.	Application and Comparison of Deep Learning Methods in the Prediction of RNA Sequence Degradation and Stability.	To propose and evaluate three Deep Learning models _ LSTM, GRU and GCN for predicting the stability/reactivity and risk of degradation of a sequence of RNA.	The study provided evidences to show that the predictions would be beneficial in the development of mRNA vaccines and that they can reduce the number of sequences synthesized and tested by helping to identify the most promising candidates. Results showed that among the three RNNs, GCN had the best result(RMSE=0.249) and GRU was the best at predicting degradation rates under various circumstances (RMSE=0.266). Overall, GRU had the best accuracy value at 76% and results suggested the feasibility of applying such

				methods in mRNA vaccine in the near future.
9.	Vandewiele, 2020	Predicting mRNA Degradation using GNNs and RNNs in the Search for a COVID-19 Vaccine.	To create a model using two RNN layers _ LSTM and GRU and make four combinations of them (GRU + GRU; LSTM + GRU; GRU + LSTM and LSTM + LSTM) for predicting the mRNA base degradation rate.	The study used the ARNIE software package for generating more information on the RNA bases; one-hot encoded the base type, predicted loop type and positional features and captured edge information using BPPs. Moreover, the study showed evidences of modified loss function, hyper-parameter tuning and data augmentation. The study showed that the usage of angle information deteriorated the performance of AE + LSTM and many software packages like RNAup and Shaker were not informative.

Table 9: Feature Comparison

Appendix C

C.1

No.	Library Name	Reason why it was used it in the project
1.	Pandas	Used for data manipulation and analysis and for manipulating numerical tables.
2.	NumPy	Used for working with arrays, had functions used for working on linear algebra, fourier transform and matrices.
3.	MatPlotLib	Used for data visualization, graphical plotting and for embedding plots in GUI applications.
4.	Json	Used for storing the train and test datasets.
5.	ast	Used for processing trees of Python abstract syntax grammar.
6.	seaborn	Was used for visualization and generating high level interfaces for drawing attractive and informative statistical graphs.
7.	os	Provided Python functions for interacting with the operating system and provided a portable way of using OS dependent functionality. The os and os. path modules provided functions for interacting with the file system.
8.	sklearn	Used for statistical modelling – regression and dimensionality reduction for this project.
9.	itertools	Used for standardizing a core set of fast, memory efficient tools.
10.	tqdm	Was used to output a smart progress bar by wrapping around the iterable. It not only showed how much time had elapsed but also the estimated remaining time for the iterable.
11.	TensorFlow	Used for fast numerical computing, for creating Deep Learning models directly and by using wrapper libraries.
12.	Keras	Used for running Deep Learning models fast and easily on TensorFlow.
13.	math	Used for performing all sorts of mathematical tasks on Python.
14.	random	Used for generating pseudo-random variables and selecting random elements from lists and shuffling elements randomly.
15.	lightgbm	Used for implementing LightGBM Regressor.
16.	warnings	Used for displaying warning messages in situations which are not necessarily exceptions. Warnings were used to display messages whenever certain programming elements like keywords, functions or class got deprecated.

Table 10: Library names along with reason for their usage within the project

C.2

No.	Names	Description
1.	id	An arbitrary identifier for each sample (kaggle.com, n.d.).
2.	seq_scored	(68 in Train and Public Test, 91 in Private Test) Integer value denoting the number of positions used in scoring with predicted values. This should match the length of reactivity, deg_* and *_error_* columns. Note that molecules used for the Private Test will be longer than those in the Train and Public Test data, so the size of this vector will be different (kaggle.com, n.d.).
3.	seq_length	(107 in Train and Public Test, 130 in Private Test) Integer values, denotes the length of sequence. Note that molecules used for the Private Test will be longer than those in the Train and Public Test data, so the size of this vector will be different (kaggle.com, n.d.).
4.	sequence	(1x107 string in Train and Public Test, 130 in Private Test) Describes the RNA sequence, a combination of A, G, U, and C for each sample. Should be 107 characters long, and the first 68 bases should correspond to the 68 positions specified in seq_scored (note: indexed starting at 0) (kaggle.com, n.d.).
5.	structure	(1x107 string in Train and Public Test, 130 in Private Test) An array of (,), and . characters that describe whether a base is estimated to be paired or unpaired. Paired bases are denoted by opening and closing parentheses e.g. (....) means that base 0 is paired to base 5, and bases 1-4 are unpaired (kaggle.com, n.d.).
6.	reactivity	(1x68 vector in Train and Public Test, 1x91 in Private Test) An array of floating point numbers, should have the same length as seq_scored. These numbers are reactivity values for the first 68 bases as denoted in sequence, and used to determine the likely secondary structure of the RNA sample (kaggle.com, n.d.).
7.	deg_pH10	(1x68 vector in Train and Public Test, 1x91 in Private Test) An array of floating point numbers, should have the same length as seq_scored. These numbers are reactivity values for the first 68 bases as denoted in sequence, and used to determine the likelihood of degradation at the base/linkage after incubating without magnesium at high pH (pH 10) (kaggle.com, n.d.).
8.	deg_Mg_pH10	(1x68 vector in Train and Public Test, 1x91 in Private Test) An array of floating point numbers, should have the same length as seq_scored. These numbers are reactivity values for the first 68 bases as denoted in sequence, and used to determine the likelihood of degradation at the base/linkage after incubating with magnesium in high pH (pH 10) (kaggle.com, n.d.).
9.	Deg_50C	(1x68 vector in Train and Public Test, 1x91 in Private Test) An array of floating point numbers, should have the same length as seq_scored. These numbers are reactivity values for the first 68 bases as denoted in sequence, and used to determine the likelihood of degradation at the base/linkage after incubating without magnesium at high temperature (50 degrees Celsius) (kaggle.com, n.d.).

10.	Deg_Mg_50C	(1x68 vector in Train and Public Test, 1x91 in Private Test) An array of floating point numbers, should have the same length as seq_scored. These numbers are reactivity values for the first 68 bases as denoted in sequence, and used to determine the likelihood of degradation at the base/linkage after incubating with magnesium at high temperature (50 degrees Celsius) (kaggle.com, n.d.).
11.	*_error_ *	An array of floating point numbers, should have the same length as the corresponding reactivity or deg_* columns, calculated errors in experimental values obtained in reactivity and deg_* columns (kaggle.com, n.d.).
12.	Predicted_loop_type	(1x107 string) Describes the structural context (also referred to as 'loop type')of each character in sequence. Loop types assigned by bpRNA from Vienna RNAfold 2 structure. From the bpRNA_documentation: S: paired "Stem" M: Multiloop I: Internal loop B: Bulge H: Hairpin loop E: dangling End X: eXternal loop (kaggle.com, n.d.).
13.	S/N filter	Indicates if the sample passed filters (kaggle.com, n.d.).

Table 11: Names of the different features in the dataset along with their description

C.3

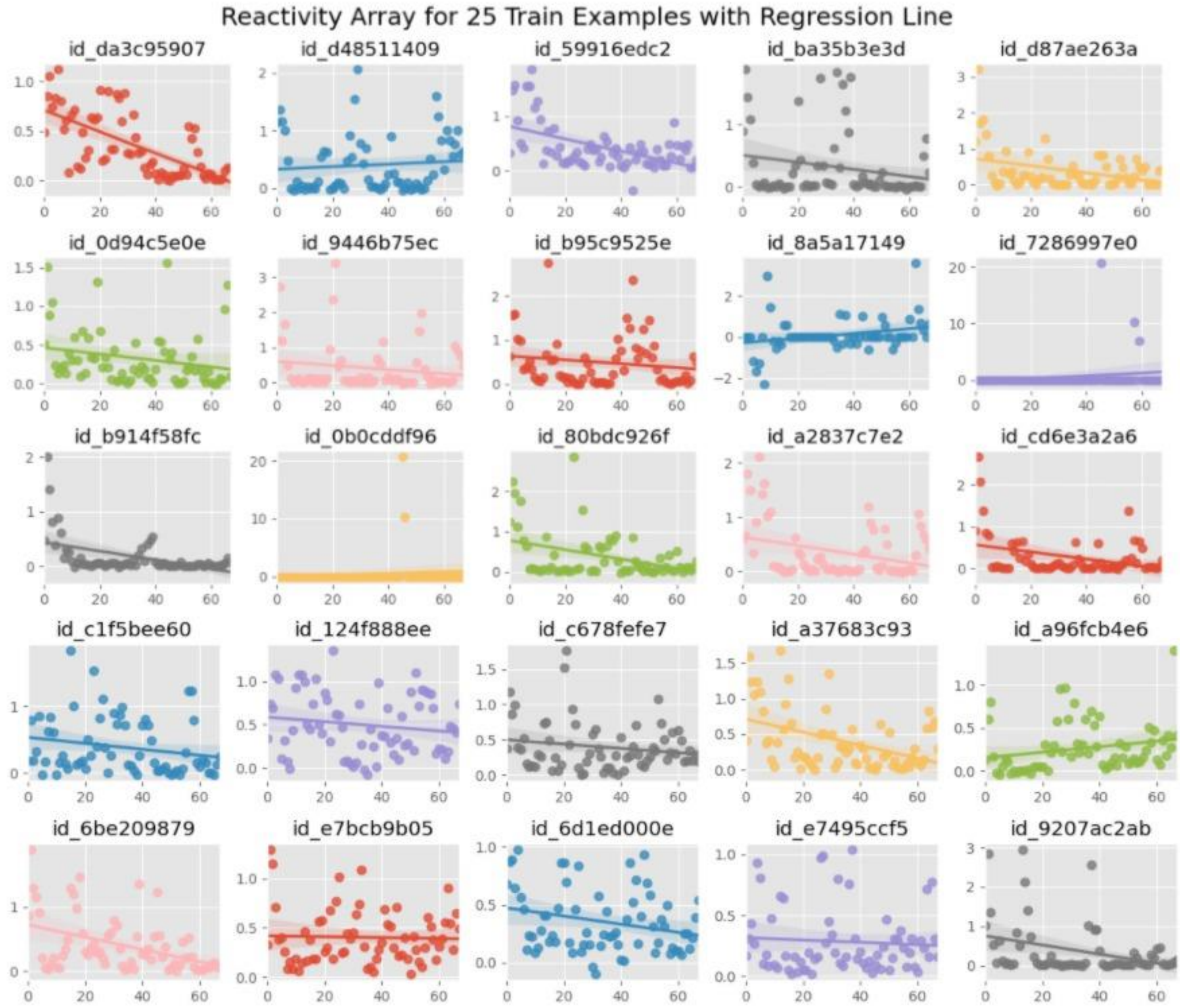


Figure 34: Regplot showing the distribution of reactivity in the first 25 samples of the training dataset.

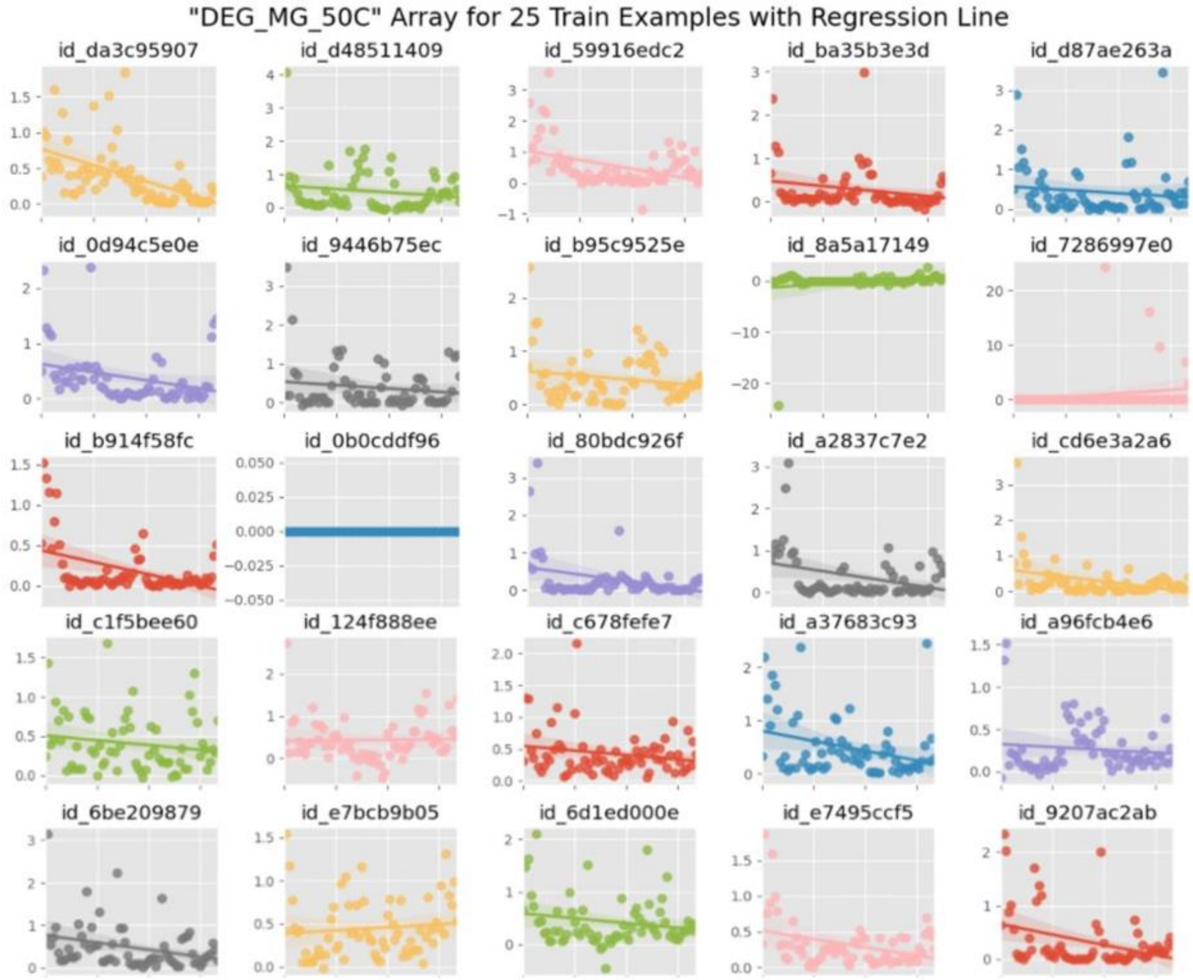


Figure 35: Regplot showing the distribution of deg_Mg_pH10 in the first 25 samples of the training dataset.

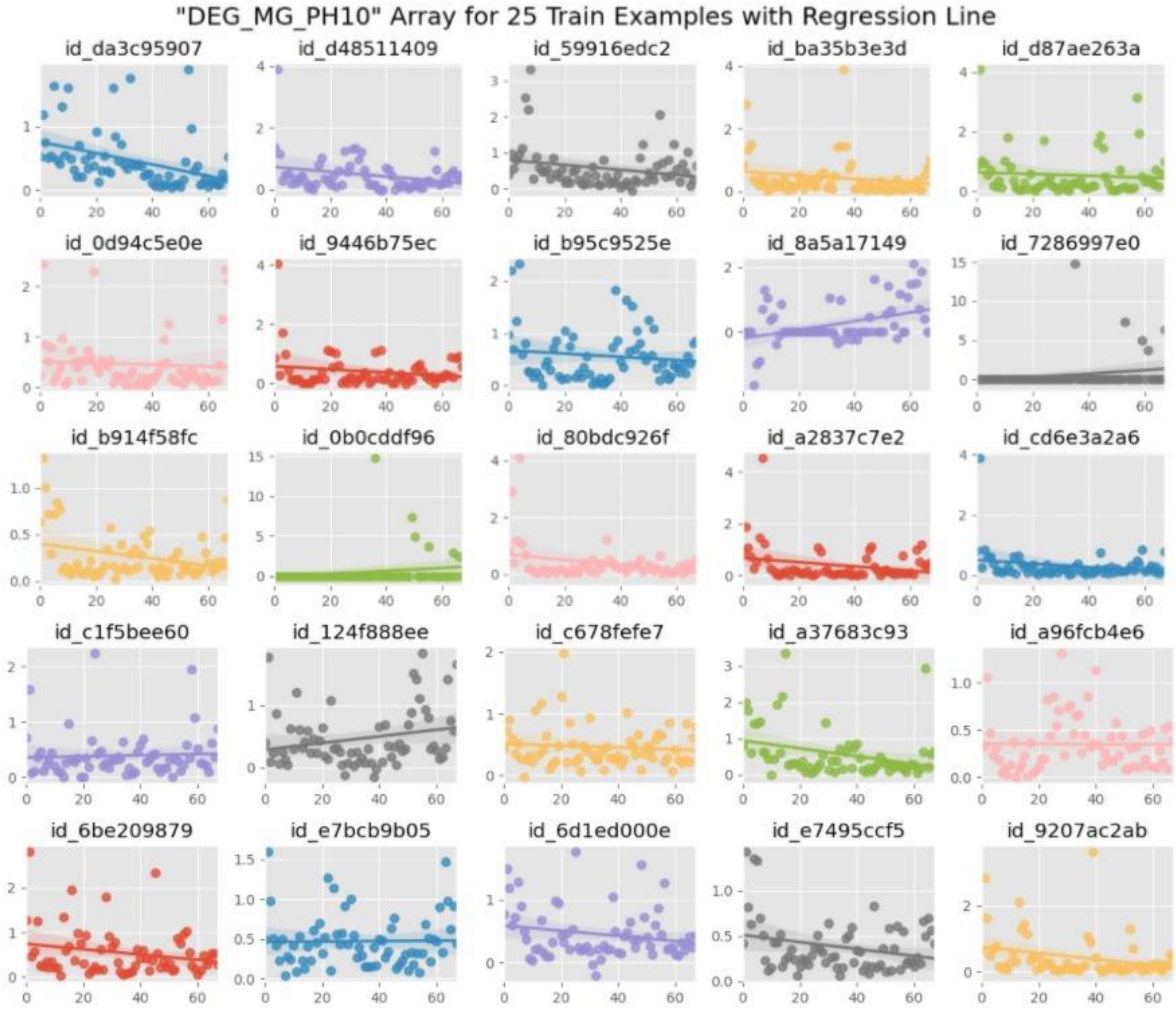


Figure 36: Regplot showing the distribution of deg_Mg_pH10 in the first 25 samples of the training dataset.

C.4

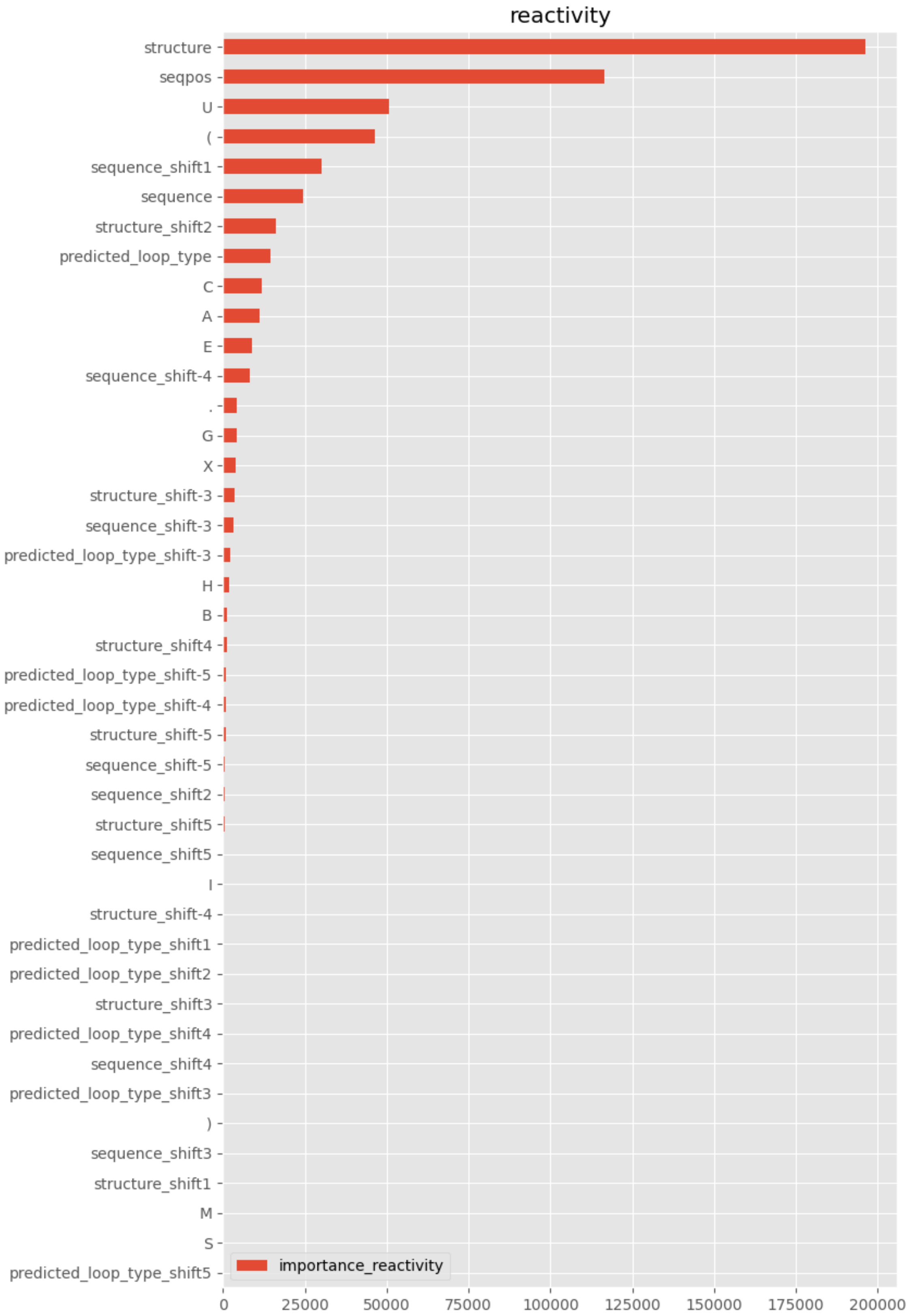


Figure 37: Horizontal bar graph showing the relative importance of reactivity column.

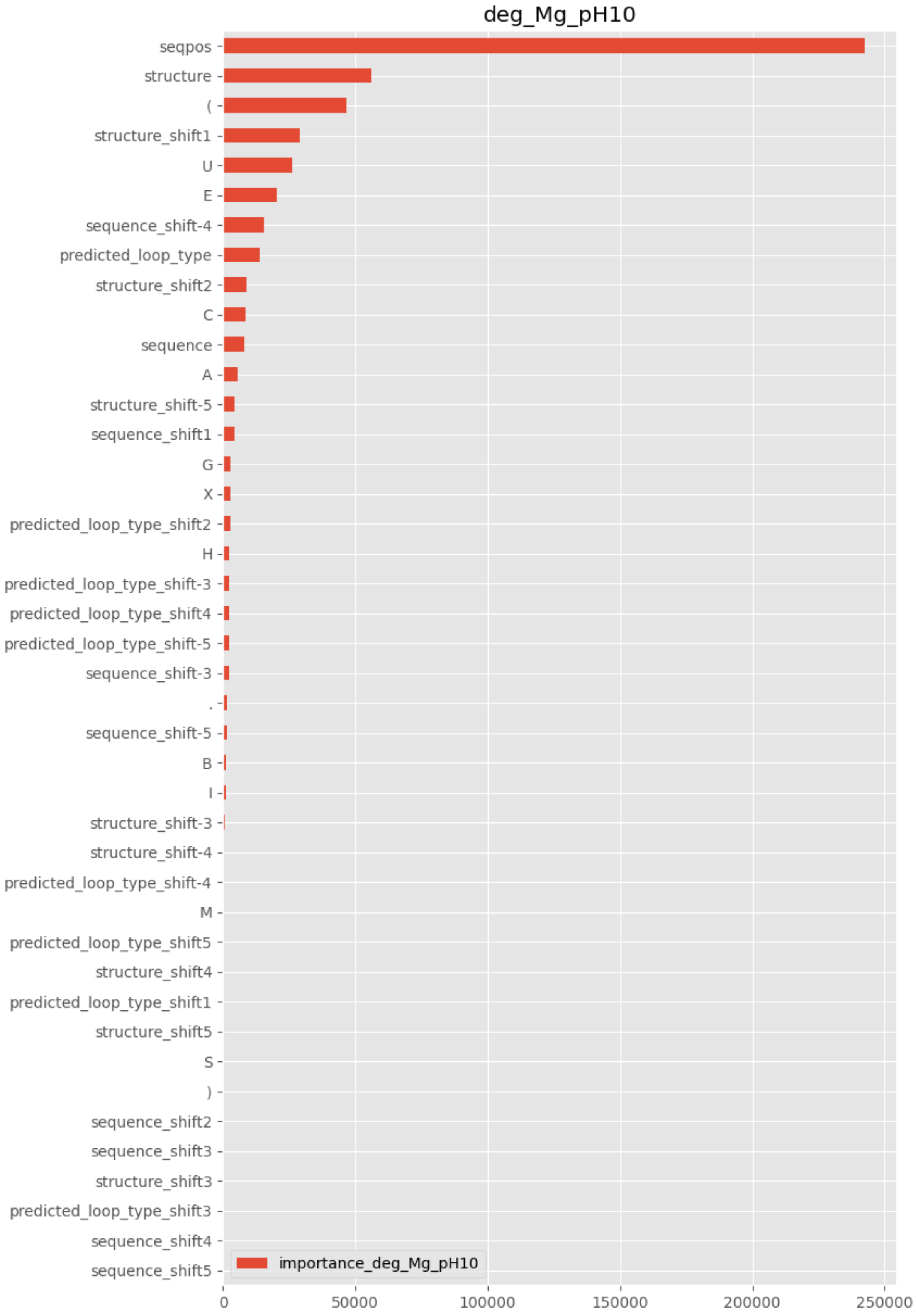


Figure 38: Horizontal bar graph showing the relative importance of deg_Mg_pH10 column.

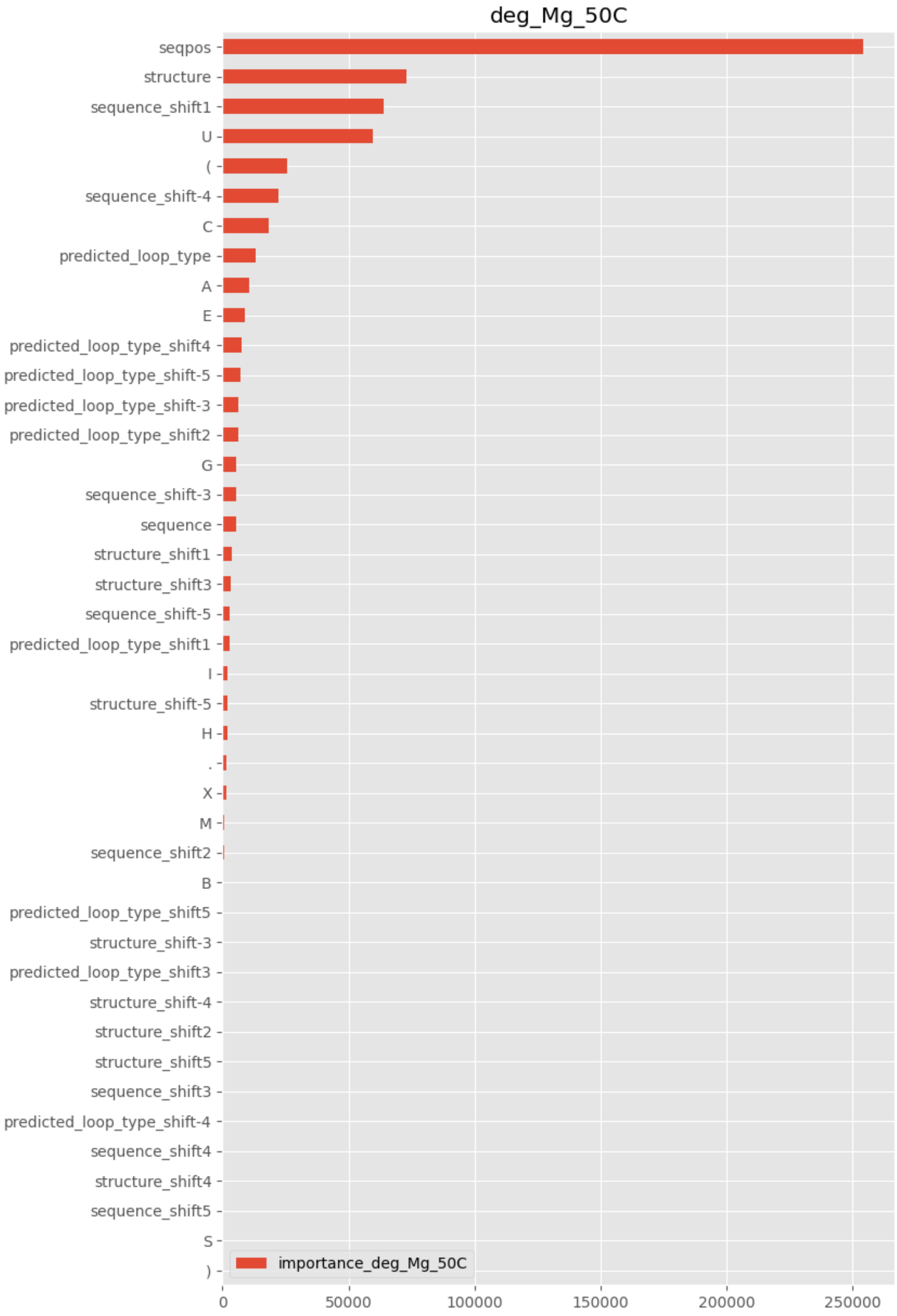


Figure 39: Horizontal bar graph showing the relative importance of deg_Mg_50C column.

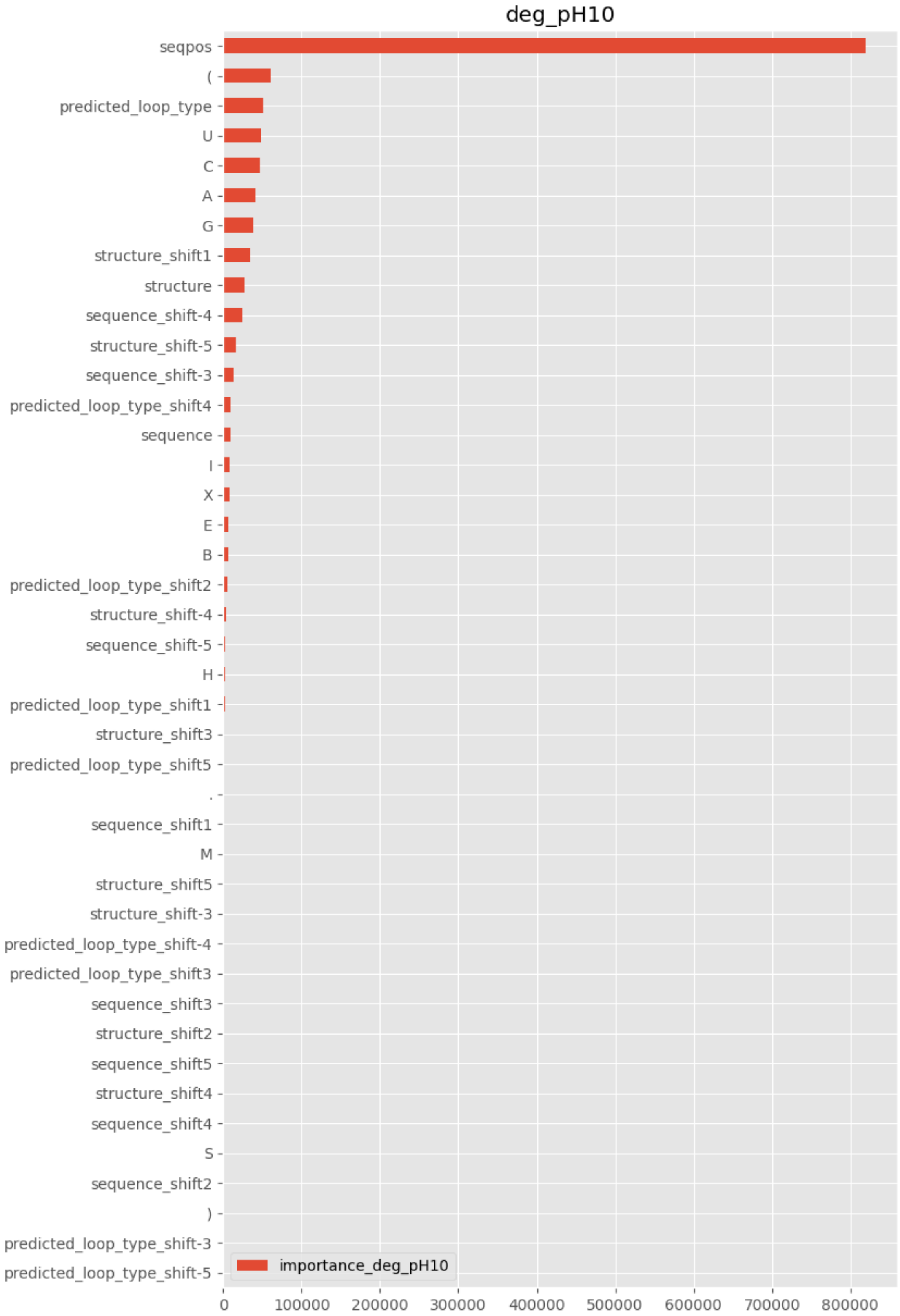


Figure 40: Horizontal bar graph showing the relative importance of deg_pH10 column.

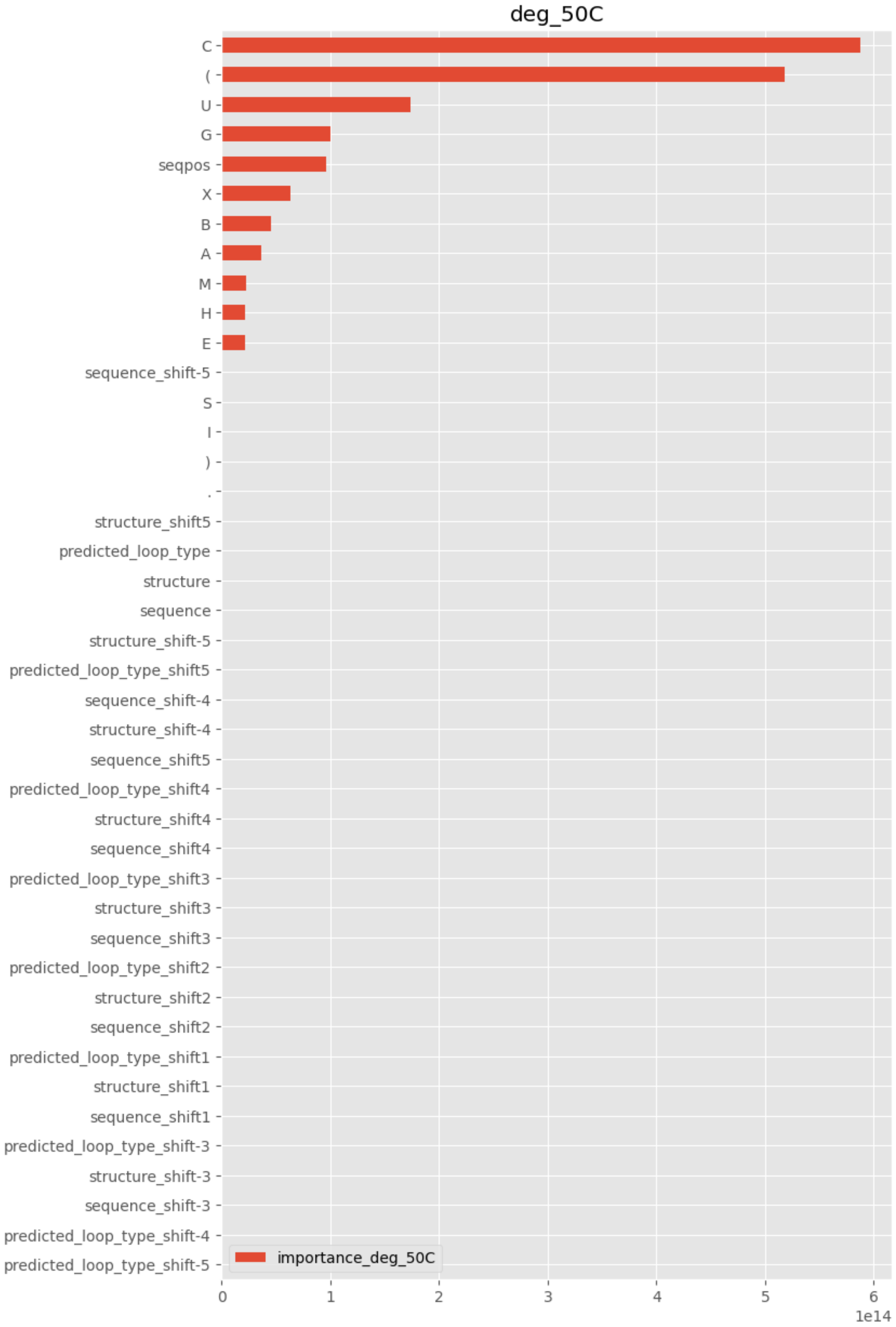


Figure 41: Horizontal bar graph showing the relative importance of deg_50C column.

C.5

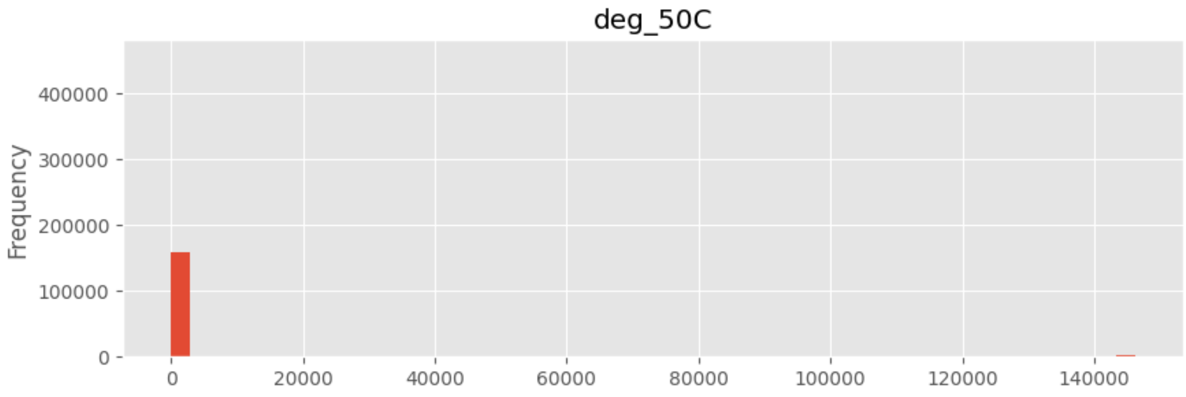


Figure 42: Histogram plotted for deg_50C for train and test data.

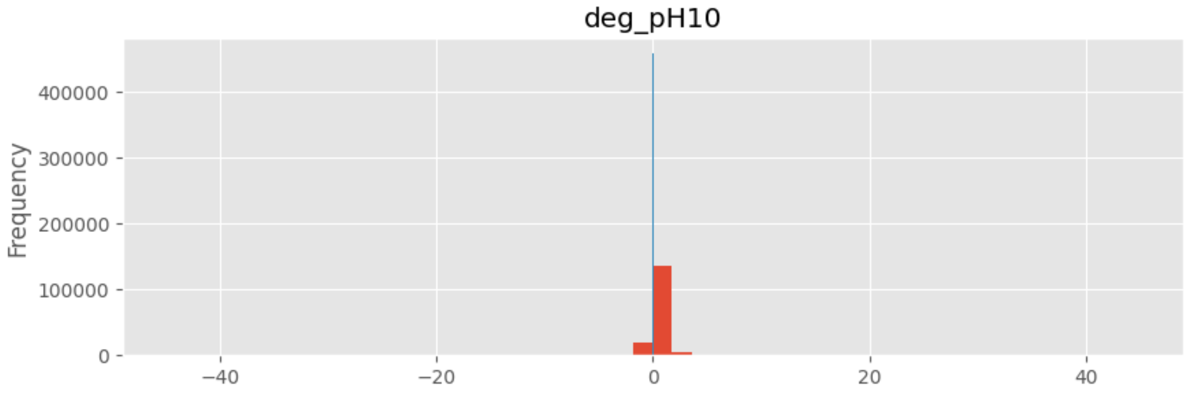


Figure 43: Histogram plotted for deg_pH10 for train and test data.

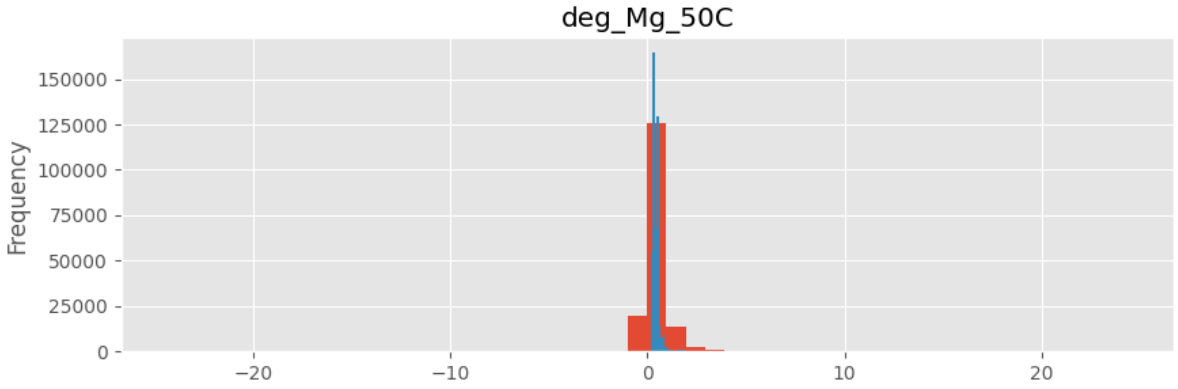


Figure 44: Histogram plotted for deg_Mg_50C for train and test data.

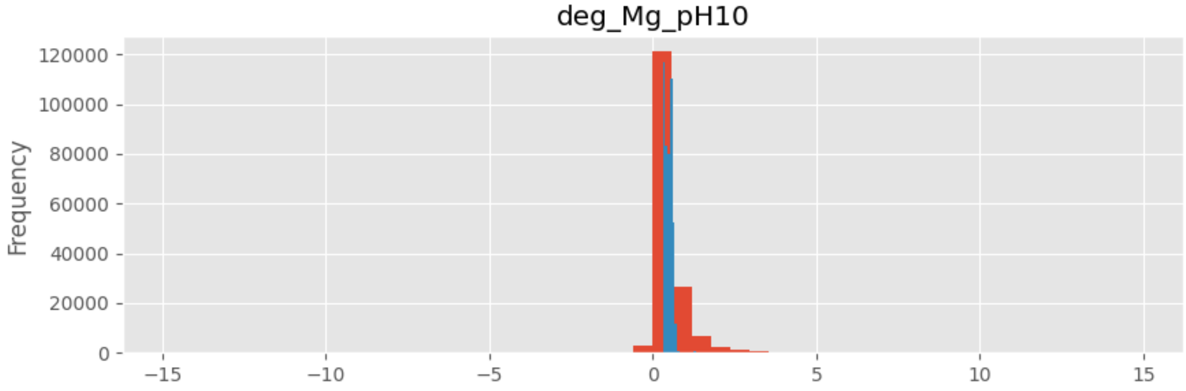


Figure 45: Histogram plotted for deg_Mg_pH10 for train and test data.

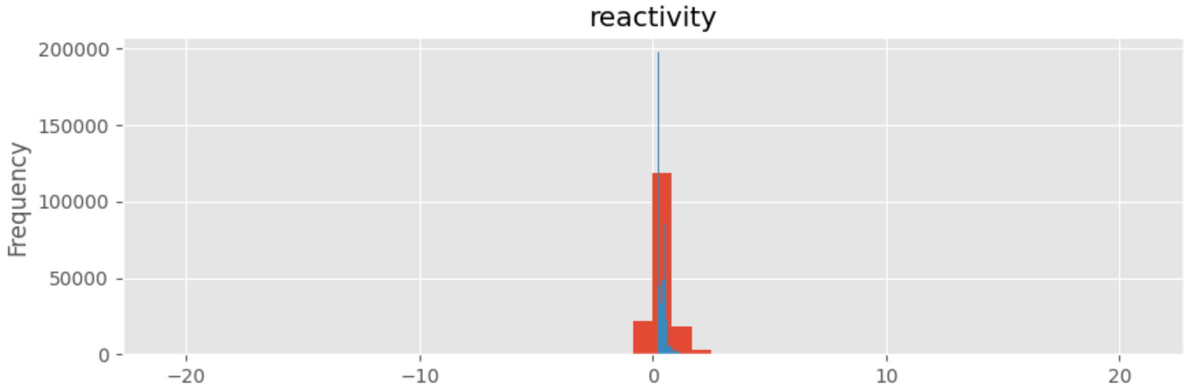


Figure 46: Histogram plotted for reactivity for train and test data.

C.6.

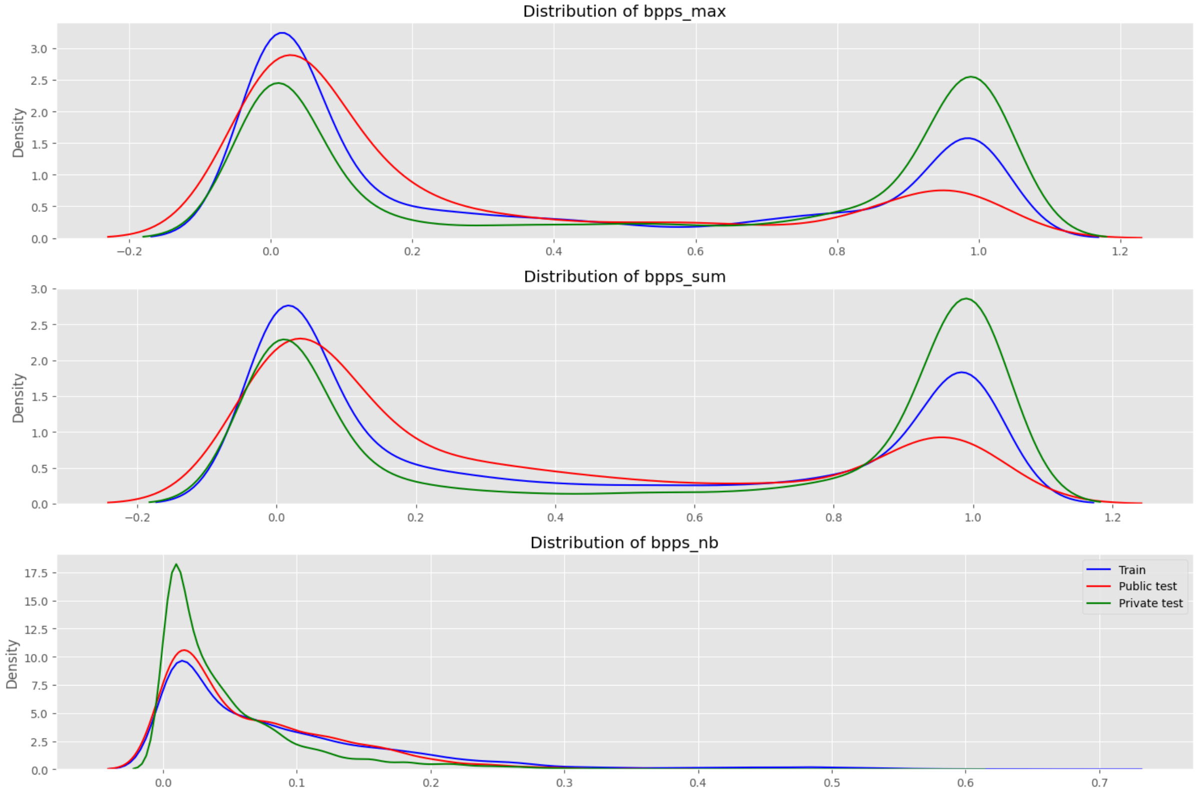


Figure 47: Kdeplots showing the distribution of bpps_max, bpps_sum and bpps_nb.