Improving COVID-19 mRNA vaccine degradation prediction

Vyom Pathak[1*], Rahul Roy[1+],

1 Department of Computer and Information Science and Engineering, University of Florida, Gainesville, Florida, United States

* v.pathak@ufl.edu
+ roy.rahul@ufl.edu

Abstract

The COVID-19 pandemic has shown devastating effects on a global scale for the past 2 years. An effective vaccine that can be distributed easily is very important to stop the COVID-19 pandemic. While mRNA vaccines are the fastest candidates, they tend to be unstable due to the spontaneous degradation of RNA molecules even with a single cut. In this paper, we develop modeling techniques to predict the rules of degradation of mRNA molecules in the COVID-19 vaccine using Deep learning techniques like Hybrid GRU-LSTM Networks, Graph Convolutional networks, and autoencoders. We state the improvement over mRNA Vaccine Degradation prediction by comparing both these methods for their MCRMSE loss values. We take the RNA structure along with their loop type as input and predict five values to understand the degradation criteria at each point in an RNA molecule using the Eterna Dataset consisting of 3029 mRNA sequences. We also used an augmented dataset generated using the ARNIE package to improve the previously described deep learning techniques. We show an overall improvement of 0.007 while using Graph Transformers as compared to the Hybrid GRU-LSTM models.

Author summary

COVID-19 vaccines are important in this day and age to tackle the global pandemic. The most effective COVID-19 vaccine that has been developed i.e. the mRNA vaccine is quite unstable at high temperatures. Thus, predicting when a particular mRNA molecule will degrade will help the scientist to form much more robust mRNA COVID-19 vaccines. This paper entails modeling techniques using deep learning to predict the rules of degradation of mRNA molecules in a COVID-19 vaccine. Our potential deep-learning-based solutions include GCN, Hybrid Bi-LSTM-Bi-GRU, auto-encoders for data de-noising, and Transformer-based models for encapsulating nuanced features of the mRNA structure; which are described in detail in the model section. We used the Eterna dataset to run our experiments. We also used augmentation techniques like ARNIE to generate more data to improve the described modeling methods. At a high level, the models take RNA sequence structure and its loop type as input; and predict the five-factor values which affect the degradation of mRNA molecule in the COVID-19 vaccine described in the modeling section. In the result section, we compare each models' loss, along with their respective accuracy on the given dataset. We compare the weighted average of the Hybrid GRU-LSTM models with the weighted average of the Graph Transformer pre-trained with and without the autoencoder over the Public as well as the Private Test Set. In the discussion sections, we answer any nuances from our experiments and talk about the upsides and downsides

of each model. We also talk about future work that can improve the said techniques like transfer learning, knowledge distillation technique for scaling the proposed method for online devices, and extending our work for low-resource settings. In the end, we conclude by giving an overview of the whole paper and pointing out the results along with their implications for the betterment of the scientific community.

Introduction

COVID-19 has been a turning point for humankind in the 21^{st} century. As of February 15, 2022, there have had more than 400 Million confirmed COVID-19 cases, and over 5 million deaths worldwide[1]. Because of the global devastating impact, it was declared as a worldwide pandemic by the World Health Organization (WHO) in late February 2020 [1]. It is caused by the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2); similar to other corona-viruses that having been appeared in the past 20 years which are the Middle Eastern respiratory syndrome coronavirus (MERS-CoV) and severe acute respiratory distress syndrome coronavirus (SARS-CoV) [2].

To date, there have been several attempts at finding an effective medication against the COVID-19 coronavirus, but no product has come close to fully solving the problem of curing the disease. Thus, it became necessary to develop a vaccine that can hinder the spread of the virus and curb the virus altogether, given the low-cost availability, and having 95% effectiveness against the virus [3]. The vaccine activates the body's immune system to identify and resist pathogenic agents like viruses, bacteria, and any other related microorganism [4]. There are 2 types of vaccines, protein-based, and gene-based. A protein-based vaccine teaches the immune system how to fight the virus or a bacteria, while the gene-based vaccine acts as a natural infection by grabbing the genetic instructions of the cell to generate the antigen, more specifically, the surface spike protein in the case of corona-viruses [5]. Because of the decade's worth of research focused on protein-based vaccines [first SARS viral pandemic, and then the Middle Eastern Coronavirus], scientists were focused on the mRNA vaccine development which is a lot less time-consuming, and easy to produce and scale.

The first vaccine for COVID-19 was developed in record time in just under a year [6]. This in itself is not an easy feat to achieve where the whole process was fastened by overlapping several stages of Vaccine trials. Even then, the actual discovery of the vaccine was still a big bottleneck in earlier times, but scientists were able to solve that as well by using messenger-RNA [mRNA] vaccines [7]. mRNA is a single-stranded copy of the DNA built by our cell which holds the information on how to make one type of protein. These in-structures are read by the ribosomes to make a particular protein. This is where the mRNA vaccine comes into the picture. It is one of the most promising and efficient approaches to developing a vaccine for COVID-19 whereas we used the S-protein from the SARS-CoV, we can use the same technique for the SARS-CoV-2 [7,8]. It has many advantages such as low time required to develop, ease of automation, reduced risk of pre-existing immunity against the vaccine, a less expensive process, and easy development of multiple prototype vaccines. Moreover, the greatest benefits of mRNA vaccines are safety and their non-infectious nature due to their direct injection and translation from messenger RNA into protein in the human body [9].

While mRNA vaccines currently are solving the global pandemic, there are several caveats as well to these types of vaccines. One of the biggest challenges is to design a stable mRNA molecule. It has been interposed that RNA particles degrade sporadically, even with a single cut. Also, the mRNA vaccine COVID-19 vaccine needs to be transported and stored under very low-temperature i.e. strong refrigerators to maintain

[1] https://www.who.int/publications/m/item/weekly-epidemiological-update-on-covid-19—15-february-2022

their stability [10]. This was a huge challenge before the pandemic, which is solved up to some extent because of the refrigerator-stable COVID-19 vaccine.

Machine Learning, particularly deep learning has resulted in a significant advantage in a huge range of contexts in all fields of technology because of its learning capacity. Recently, advancements in RNN, CNN, and GCNs have opened doors for the modeling of RNA and DNA sequences. RNNs are utilized to predict the patterns from an acyclic graph data-representations [11]. Moreover, RNN captures the interactions between different elements through mRNA sequences. Particularly, there are two types of RNNs: gated recurrent units (GRU), and long short-term memory units (LSTM). GRUs can solve sequentially complex decomposition problems which cannot be solved by vanilla RNNs [12]. It also doesn't suffer from any optimization constraint as opposed to vanilla RNN. LSTMs are similar to GRU in terms of sequential applications, as well as better than GRU at solving the vanishing gradient problems by using 3 gates and memory cell [13]. More research generalized these RNNs into Graph Neural Networks (GNNs) to capture directed as well as undirected graph structures [14]. Convolutions (CNN) was then designed to handle representations from the spatial domain to a graphical domain [15]. Graph convolutional networks (GCNs) are an amalgamation of GNN and CNN [16]; modeling the structural as well as spatial features from an mRNA sequence.

In this paper, we develop modeling techniques to predict the rules of degradation of mRNA molecules in the COVID-19 vaccine using modern data science techniques, particularly deep-learning algorithms. The model predicts the likelihood of the degradation rate for each position inside the mRNA molecule. The modeling techniques were trained on 3000 mRNA sequences and structures and the scoring and the predictions were done on a small part of the same dataset[2]. We increased the amount of data by using augmentation using the ARNIE model. Our first method uses hybrid Bi-LSTM-Bi-GRU models, as the data consists of sequences with spatial patterns which can be easily deciphered and understood by such modeling techniques [17, 18]. Furthermore, we incorporate Graph Convolutional Neural Networks to solve the said problem, as graph convolutions capture the sequential information over the graph as a deep-stacked structure. We look at de-noising the input as well by using auto-encoders to encode the input data-points [16]. We compare both of these techniques based on their MCRMSE test loss values and state the importance of Attention and Graph-based models over RNN models for mRNA Degradation prediction tasks.

Materials and methods

Dataset

The primary dataset is the Stanford COVID19 mRNA vaccine dataset[2] posted to Kaggle. It consists of sequences data and bpps data. The data consists of 3029 RNA sequences. This is divided into the train (2400) and test sets (629).

The training file has 13 fields including 5 ground truth values. The sequence, structure, and predicted loop type parameters explain the RNA sequence, whether a base is paired or unpaired, and the structural context, respectively. The sequence contains 3 features - RNA sequence, structure, and predicted loop type of each character in the sequence. The public test set is of length 107 sequence length of which only the first 68 are scored while the private test set is of length 130. The test file consists of 7 fields excluding the ground truth values.

The target includes five values. The reactivity which determines the secondary structure of the mRNA sequence, deg_Mg_pH10 and deg_pH10 determine the likelihood of degradation at the base after incubating with/without magnesium at high pH (pH

[2]https://www.kaggle.com/c/stanford-covid-vaccine/data

10) and; deg_Mg_50C and deg_50C likelihood of degradation at the base after incubating with/without magnesium at high temperature. The 13 fields in the dataset are described as follows:

1. **id**: An arbitrary identifier for each sample.
2. **seq_scored**: Integer value denoting the number of positions used in scoring with predicted values. It is 68 in Train and Public Test and 91 in Private Test.
3. **seq_length**: Integer value denoting the length of the sequence. It is 107 in Train and Public Test and 130 in Private Test.
4. **sequence**: Describes the RNA sequence, a combination of A, G, U, and C for each sample. It is 107 characters long, and the first 68 bases correspond to the 68 positions specified in seq_scored
5. **structure**: 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
6. **reactivity**: 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.
7. **deg_pH10**: 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).
8. **deg_Mg_pH10**: 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).
9. **deg_50C**: 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 a high temperature (50 degrees Celsius).
10. **deg_Mg_50C**: 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 a high temperature (50 degrees Celsius).
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.
12. **predicted_loop_type**: Describes the structural context of each character in sequence.
13. **S/N filter**: Indicates if the sample passed the filters.

The bpps data are symmetric square matrices pre-calculated for each sequence with the same length as the sequence. The bbp matrix indicates the likelihood that each pair of nucleotides in the RNA will form a base pair in the ensemble of RNA secondary structures, providing more robust and detailed information than a single RNA secondary structure [19].

Augmented Data

One of the most important steps for increasing the number of training examples and overcoming the over-fitting problem is the use of data augmentation. For the COVID-19 mRNA dataset, we need the sequence, structure, and loops. We used the ARNIE library in python which supports multiple secondary structure packages. Specifically, we perform the secondary structure prediction using the Vienna RNA sub-package. The size of the augmented data is 2400 which is equal to the original train data.

Modeling Techniques

Since the data consists of sequences with spatial patterns we look at different sequence modeling techniques which are as follows:

- Ensemble model of Bi-LSTM [13], Bi-GRU and Hybrid Bi-LSTM-Bi-GRU models.
- Graph Convolutional Network [16] with attention [17] for memory and Auto encoders for preprocessing.

We compare both methods as the weighted average of the Hybrid GRU-LSTM models and the weighted average of the Graph Transformer network with and without autoencoder pre-training based on the MCRMSE Loss value for the Public Test and Private Test Set.

Hybrid GRU-LSTM Model

This method is composed of bidirectional GRU, LSTM, and Hybrid models applied to mRNA sequence data and bpps data. Firstly, we performed data augmentation to increase the number of data points and decrease overfitting. The approach then begins with feature engineering to extract the features before employing a sequence model to predict the mRNA sequences responsible for degradation by estimating five reactivity values for each place in the sequence. The features engineering process extracts two kinds of features: numerical features and categorical features. The categorical features are encoded and then dense embedding has been extracted for capturing the sequential relationships. Then, numerical features are extracted from the BPPS data. These features are combined and then passed onto the Hybrid model. The proposed model pipeline is described in Fig 1. This architecture was inspired from [20].

Feature Engineering

The following two sections define the two types of features:

Categorical Features

Because Deep Learning-based approaches only take quantitative features, it is critical to preprocess and transform category variables into numerical features for the technique to process the data and extract important information. In mRNA there are three types of sequence data to encode using the base method:

1. Structure Encoded from 0 to 2 (which has (). characters)
2. Sequence Encoded from 3 to 6 (which has AGUC characters)
3. Modified Loop Encoded from 7 to 13 (which has BEHIMSX characters)

After encoding, the result is passed through an embedding layer, which allows information extraction as a dense vector. Here, we extract 300 features using the embedding layer from the input encoding feature. Here the dense vectors represent the projection of the base into a continuous vector space [21].

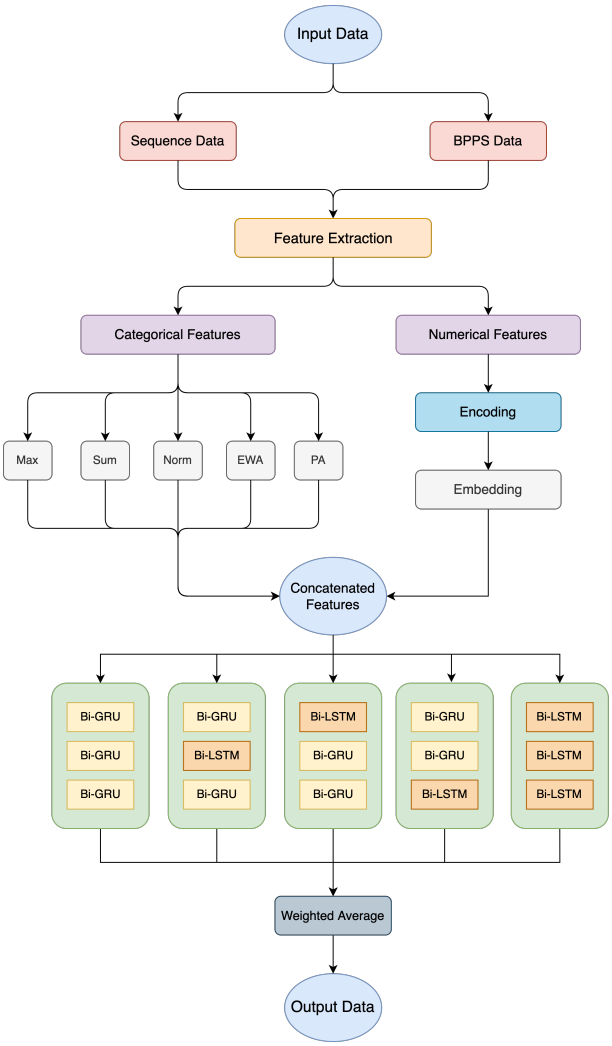

Fig 1. Hybrid GRU-LSTM Model Architecture. The architecture for the ensemble of Bi-GRU, Bi-LSTM, and hybrid BiGRU-Bi-LSTM models. Firstly, we perform the feature extraction from sequence and BPPS data. For the BPPS data, we use formulae to find 5 categorical features. For the Sequence data, we perform encoding using a pre-built dictionary and then calculate the embeddings and derive the respective numerical features. We then concatenate both these features to generate about 305 features and then pass them to the 5 models. We train them individually and then for testing we ensemble the weighted average of all these models. We use the GRU-LSTM model to capture the temporal and spatial features from the RNA structure.

Numerical Features

Based on empirical results, we selected the appropriate numerical formulae for this step. All the features rely on the pre-computed base pair probabilities from bpps matrix for each sequence data. For all of the following calculations, n is the length of the bpp vector, and bpp_i is the probability of the given base with the position i.

$$Max = max(bpp) \tag{1}$$

$$Sum = \sum_{i=1}^{n} bpp_i \tag{2}$$

$$Norm = \frac{(\frac{\sum_{i=1}^{n} bpp_i}{n} - \mu)}{\sigma} \tag{3}$$

$$EWA = \sum_{i=1}^{n+1} V_i \tag{4}$$

$$V_i = \beta V_{i-1} + (1 - \beta) bpp_{i-1} \tag{5}$$

$$PA = \frac{\sum_{i=1}^{n} i * bpp_i}{n} \tag{6}$$

Here, EWA is the exponential Weighted Average of each vector of probabilities in the given bpps matrix for a particular base, and V_i is the weighted probability for each position i in the bpp vector and β is a constant value equal to 0.9. PA describes the position multiplying the value of position in bpp provide the position average value.

Sequence Modeling

The proposed model architecture is described in Fig 1 with the following different variations:

1. Type 0 Model - Three Bidirectional GRU Layers.
2. Type 1 Model - One Bidirectional GRU Layer, followed by one Bidirectional LSTM Layer, and finally a Bidirectional GRU Layer.
3. Type 2 Model - One Bidirectional LSTM Layer, followed by one Bidirectional GRU Layer, and finally a Bidirectional GRU Layer.
4. Type 3 Model - One Bidirectional GRU Layer, followed by one Bidirectional GRU Layer, and finally a Bidirectional LSTM Layer.
5. Type 4 Model - Three Bidirectional LSTM Layers.

Each layer has 256 units in each direction and uses a dropout of value 0.5 to prevent over-fitting. The use of bidirectional layers enhances the results for collecting information in sequential data. All 5 models have a final densely connected layer with 5 outputs which are deg_Mg_pH10, deg_pH10, deg_Mg_50C, and deg_50C, and linear activation function because we have to solve a regression problem. There are a total of about 3.5 million parameters. The optimizer is adam optimizer as it is suitable for noisy data and non-stationary objectives. The loss function is Mean Column-wise Root Mean Square Error (MCRMSE) shown in the below equation. This loss is also used to compare the models as well as used for testing in the Kaggle Competition.

$$MCRMSE = \frac{1}{N_t} \sum_{j=1}^{N_t} \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_{ij} - \hat{y}_{ij})^2} \tag{7}$$

where N_t is the number of predicted ground truth columns, and y and $\hat{y}$ are the actual and predicted values respectively.

Graph Convolution Network

Feature Engineering

To use the GCN model for predicting degradation, the mRNA structure should be represented as a graph. The adjacency matrix was created by concatenating 3 parts. The base pair probability matrix is given along with the data that forms the first part of the adjacency matrix. Secondly, to represent the actual base pairing, we computed a structure adjacency matrix from the column *structure* given in the data. Lastly, we also computed a distance matrix as the inverse of the absolute distance between the sequences. The nodes are formed by one-hot encoding the *sequence* and *predicted loop type* columns.

Modeling

The proposed GCN [16] model architecture is shown in Fig 2. It consists of a convolution block followed by a multi-head attention block and a final densely connected layer with 5 outputs which are deg_Mg_pH10, deg_pH10, deg_Mg_50C, and deg_50C [22]. The convolution block has 128 units of kernel size 3, 64 units of kernel size 6, 32 units of Kernel size 15, and finally 16 unit layers with kernel size 30. Each of these convolution layers uses layer normalization with Leaky ReLU as the activation function. This is then followed by multi-head attention [17] which computes the scaled dot product attention multiple times in parallel. We used a linear activation function with glorot uniform as kernel and bias initializer for the attention layers [17]. The dropout was set at 0.4. This model was also trained using MCRMSE as a loss function with adam optimizer. We also pretrained the model using denoising autoencoders. The entire dataset including the test set was used for this. This will help in reducing the effects of the noisy input. We used a spatial dropout of 0.3 for the denoising autoencoders.

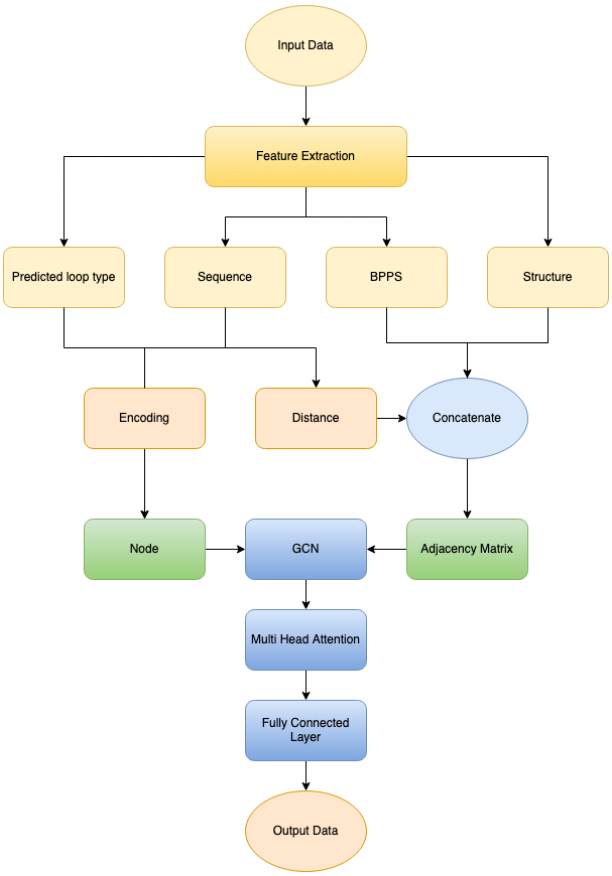

Fig 2. GCN Architecture. The architecture of the Graph Convolutional Network with multi-head attention. The predicted loop type and the sequence data are encoded to form the node features. The distance within the sequence is concatenated with the bpps data and the structure information to form the adjacency matrix. The node features and the adjacency matrix is passed as input to the GCN and then to attention layers. Finally, a fully connected layer is used to predict the 5 target values.

All the codes related to this project are uploaded to the following GitHub repository: mRNA-Vaccine-Degradation-Prediction. The dataset can be found in the Kaggle Competition: OpenVaccine: COVID-19 mRNA Vaccine Degradation Prediction

Results

Hybrid GRU-LSTM Model

For the training phase, we concatenated the augmented data along with the original train data to form a total of 4800 data points. We excluded the noisy data based on the *signal_to_noise* values leaving us with 4192 training examples. We used the Group-KFold method where we first clustered (200 cluster centers) the data based on similarity, and then divided data into 10 folds for better Cross-Validation results. Thus, we have 838 samples for validation (for selecting the best model) and 3354 samples for training all 5 models. As discussed, we encode and embed the categorical features into 100 embeddings for each feature giving us 300 features. We used a batch size of 64 for training the models for about 60 epochs. The training was done on Hipergator

Tensorflow-2.7.2 Environment with 1 A100 GPU taking about 1.5 hours for all folds for all 10 models.

The testing was done on the test set of the COVID-19 mRNA dataset. The testing set was divided into two parts which are 107 sequence length data for the public Kaggle Test set, and 130 sequence length data for the private Kaggle Test set. The output from all folds for each model is concatenated by the weighted average and the final results are submitted to the Kaggle Competition for evaluation.

Fig 3 shows the training and validation losses for each model. We can see that even with the introduction of augmented data for data diversity there is still some amount of overfitting. This might be the case because the size of the model is larger than the dataset required to train it i.e. the model is not able to learn the data distribution. Table 1 shows the Validation MCRMSE Loss for each model averaged over 10 folds for the Hybrid GRU-LSTM model and 7 folds for the GCN Transformer with and without autoencoder pretraining; along with their weighted average respectively. It shows how the GCN transformer-based method exceeds the performance of the Hybrid GRU-LSTM model. Table 2 shows the Kaggle Public and Private MCRMSE score for the ensemble of the models. We can see that the validation results match very closely with the public test set. By using the GRU and LSTM architecture, we can see that we can get similar results for the public test set. The GCN Transformer without the autoencoder gave better accuracy than with the autoencoder model across the validations set. However, as seen in Table 2 the GCN Transformer with autoencoder gives better results as compared to the model without autoencoder; outperforming the GRU-LSTM model. For the private test set, the data distribution is quite different as per the dataset author and thus the results are quite far apart from each other.

Table 1. MCRMSE results for the Hybrid GRU-LSTM model, and GCN Transformer with Auto-Encoder

Model Name	Validation Loss (MCRMSE)
$Bi-GRU\ Bi-GRU\ Bi-GRU$	0.22661
$Bi-GRU\ Bi-LSTM\ Bi-GRU$	0.22616
$Bi-LSTM\ Bi-GRU\ Bi-GRU$	0.22708
$Bi-GRU\ Bi-GRU\ Bi-LSTM$	0.22761
$Bi-LSTM\ Bi-LSTM\ Bi-LSTM$	0.23067
$Weighted\ Average\ Ensemble\ Hybrid\ GRU-LSTM$	0.22762
$GCN\ with\ autoencoders$	0.17561
$GCN\ without\ autoencoders$	0.17041
$Weighted\ Average\ Ensemble\ Hybrid\ GRU-LSTM$	0.17301

MCRMSE results of the Hybrid GRU-LSTM models averaged over 10 folds based on the numerical and categorical features and base encoding method with augmentation for Type 0: Bi-GRU*3, Type 1: Bi-LSTM*1 Bi-GRU*2, Type 2: Bi-GRU*1 Bi-LSTM*1 Bi-GRU*1, Type 3: Bi-GRU*2 Bi-LSTM*1, and Type 4: Bi-LSTM*3 and for the weighted ensemble of Hybrid GRU-LSTMs, as well as the GCN Transformer without and with the autoencoder model respectively.

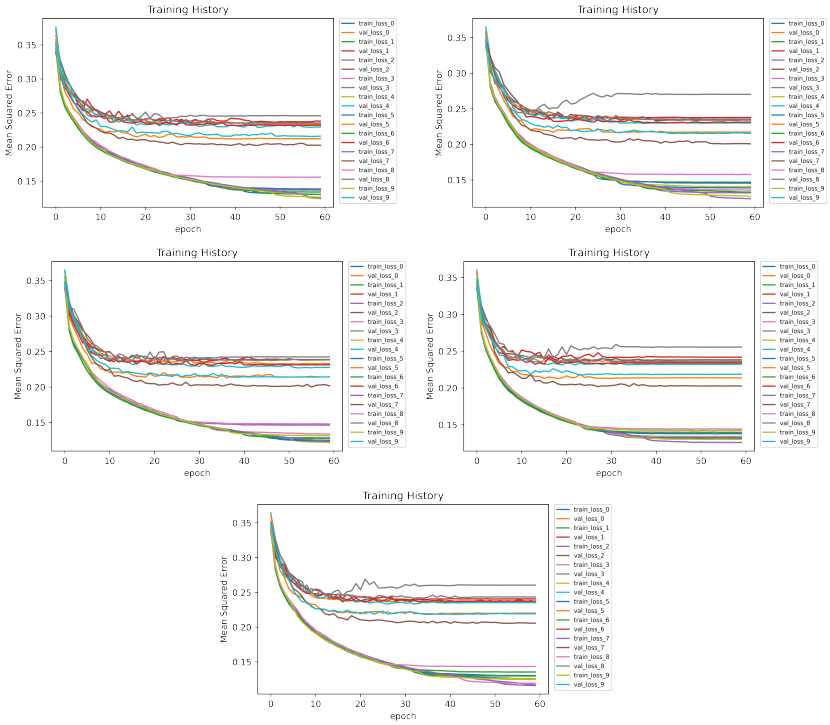

Fig 3. Hybrid GRU-LSTM Model Loss The MCRMSE loss for Bi-GRU and Bi-LSTM model variants with their respective types i.e. Type 0: Bi-GRU*3, Type 1: Bi-LSTM*1 Bi-GRU*2, Type 2: Bi-GRU*1 Bi-LSTM*1 Bi-GRU*1, Type 3: Bi-GRU*2 Bi-LSTM*1, and Type 4: Bi-LSTM*3 respectively. It shows how the training, as well as validation loss, is decreasing, albeit they are not decreasing hand-in-hand. Using 10 folds weighted average helps us to overcome some of the over fittings. We can also see from the graph that the best performing model across all the folds is the Type 2 model.

Table 2. Comparison of the Hybrid GRU-LSTM model and the GCN Transformer with Auto-Encoder Model

Model Name	Public Test Set (MCRMSE)	Private Test Set (MCRMSE)
Weighted Average Ensemble GRU-LSTM Models	0.24966	0.36468
Weighted Average of GCN Transformer with and without Auto-Encoder Models	0.24421	0.35776

MCRMSE results of the Weighted Ensemble Hybrid GRU-LSTM models based on the numerical and categorical features and base encoding method with augmentation compared and the GCN Transformer Auto-Encoder Model over Public Test Set, and Private Test Set respectively.

Graph Convolution Network

For training, we used the K-Fold cross-validation technique. Again, the noisy data were excluded based on the *signal_to_noise* values leaving us with 2096 training examples. We initially pre-trained the model using the de-noising autoencoder for 20 epochs divided into 5 epochs each for the training and test set. We then used a k value of 7 for

the K-Fold and trained the model for 100 epochs using a batch size of 32. We also trained the GCN model without the pretraining to compare the effect of the autoencoder.

The testing was done on the test set of the COVID-19 mRNA dataset. The testing set was divided into two parts which are 107 sequence length data for the public Kaggle Test set, and 130 sequence length data for the private Kaggle Test set. The output from all folds for the model is concatenated by the weighted average and the final results are submitted to the Kaggle Competition for evaluation.

Fig 4 shows the training and validation losses for the GCN model with autoencoder pretraining and Fig 5 shows the same without using the autoencoders. We can see that, there is no significant difference in the validation losses for both methods. This might be because we have already removed part of the data with high noise using the *signal_to_noise* parameter in the input data.

Table 1 shows the Validation MCRMSE Loss as a weighted average of 7 folds for the model with and without pretraining. Table 2 shows the Kaggle Public and Private MCRMSE score for the model with pretraining. We can see that the validation results match very closely with the public test set.

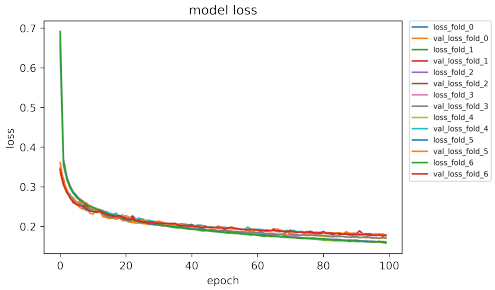

Fig 4. GCN Loss with Auto Encoders. The MCSE loss for Graph Convolutional Network with autoencoders for de-noising. It shows the model convergence over 100 epochs for 7 folds. We can see that the validation loss and training loss all converge hand-in-hand. This confirms the importance of using a graph-based network along with attention-based feature extraction. It also shows the smooth convergence of the model because of the auto-encoder pre-training.

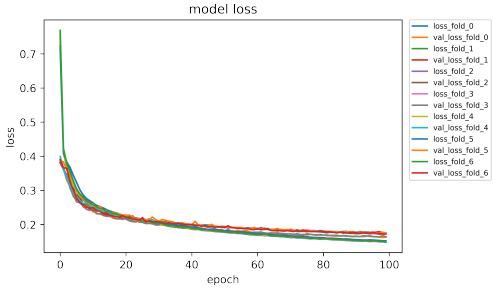

Fig 5. GCN Loss without Auto Encoders. The MCRMSE loss for Graph Convolutional Network without Autoencoders for de-noising. It shows the model convergence over 100 epochs for 7 folds. We can see that the validation loss and training loss all converge hand-in-hand. This confirms the importance of using a graph-based network along with attention-based feature extraction. This also shows that although the convergence is not as smooth as the GCN Transformer with the Autoencoder, in the end, this model slightly outperforms the former.

We compare both the GCN Transformer with Auto-Encoder for pre-processing method along with the Hybrid GRU-LSTM methods with each other in Table 2 based on their MCRMSE Public Test and Private Test set. We can see that the GCN method outperforms the Hybrid GRU-LSTM method because of the graphical nature of the mRNA data, as well as using the attention mechanism [17] helps us to learn the high-level features even better. This can be further improved by incorporating augmentation data for the GCN Transformer method.

Discussion

We developed two methods to predict the factors responsible for the mRNA vaccine degradation. Firstly, we used the sequential modeling technique using Bi-LSTM, BI-GRU, and a Hybrid Bi-GRU Bi-LSTM model as described in Fig 1. We trained this ensemble over 10 folds based on empirical experiments and show that the training, as well as validation loss, do converge albeit the validation loss is that smooth as shown in Fig 3. We used this technique to capture the temporal features of the dataset. We then trained the Graph Convolutional Network with an attention mechanism as described in Fig 2. We trained the GCN Model with and without denoising autoencoders. We compared the training and validation loss over 7 folds in Fig 4, and Fig 5 respectively. Although the final loss was less for the GCN trained without autoencoders, we found out that the convergence was smoother when using the autoencoders. This is because we had already filtered the input data with a large signal-to-noise ratio before training, resulting in less noisy data for the de-noising autoencoder to make an impact. For both of these methods, we increased the amount of data by using the ARNIE package for augmentation.

We then compare both of these models over the validation dataset as shown in Table 1. We can see that the GCN model is performing much better for the MCRMSE Loss value. This confirms that RNA sequence data can be better represented using graph structures, and hence using the Graph Convolutional Neural Networks should give better results. Furthermore, the attention-based model helped in learning high-level features much better. These models also show that the GCN models generalize much better than the Hybrid GRU-LSTM models. At last, we compare both the model's performance over the Public and Private Test Set over the Kaggle Competition as shown in Table 2. For this comparison, we use the weighted average of the Hybrid GRU-LSTM, GRU, and LSTM models and the weighted average of the GCN Transformer network with and without the denoising autoencoders. We can see that we achieve a 0.007 improvement over the Private Test Set by using the GCN models as compared to the Hybrid GRU-LSTM models. These results also show that the GCN models generalize much better than the Hybrid GRU-LSTM models.

The novelty of the results includes the use of 5 different combinations of the Bi-LSTM and Bi-LSTM models, the use of this specific augmentation for both the techniques for the given data points based on empirical experiments and the user attention-based architecture with the GCN model compared over the use of autoencoder for denoising. We show how Transformer based architectures perform better than normal seq-to-seq models.

This approach can be further improved using the Transfer-Learning approach for a large model trained over a similar task to achieve better results. Moreover, we can also scale this proposed approach for online devices [making models with low latency as well as requiring low compute power] by using knowledge distillation. Lastly, this technique can be further modified to encounter low-resource settings in a real-life scenario.

Conclusion

mRNA vaccines are the most rapid vaccination options for COVID-19 therapy, however, they have significant disadvantages, including degradation. To track this degradation, we develop and compare 2 deep learning-based methods. Both of these methods also used augmentation to increase the amount of data using the ARNIE package.

We formulate the mRNA data sequence as a sequential task, we used an ensemble of Bi-GRU, Bi-LSTM, and Hybrid Bi-GRU Bi-LSTM based models to predict the mRNA sequence degradation by predicting five reactivity values for every position in the sequence. For this method, we used two types of features: numerical features developed as a base encoding converted into neural embedding, and the categorical features from the Probabilistic Matrix values using mathematical formulas. This model gave 0.24966 and 0.36468 MCRMSE values over the public and private test sets respectively.

Sequential data like RNA can be better represented using graphs, so we used a Graph Convolutional Network with Transformer architecture to tackle this problem. We also, compared a denoising autoencoder for preprocessing. For this model, we used two types of features. The first is the node feature formed by encoding the predicted loop type and the sequence data and second is the edge features represented as adjacency matrix that contains the distance within the sequence concatenated with the bpps data and the structure information. We then passed both of these as input to the GCN and then to the attention module. The final densely connected layers predicts the 5 target values. Our comparison shows that the models' validation loss with and without autoencoder is similar but the autoencoder training is much smoother because of the denoiser. Then we ensemble both the GCN Transformer with and without denoising autoencoder models and test it over the public and private test set and get the MCRMSE scores as 0.24421 and 0.35776 respectively.

These values clearly show that the graph-based architecture captures the sequential pattern in the mRNA data much better than the seq-to-seq model showing a 0.007 improvement in the MCRMSE loss on the private test set. Moreover, we also evinced that the attention-based model helped in learning the high-level features much better. In future work, we plan to investigate the application of transfer learning from a model trained on a similar downstream task to achieve better results. To scale the proposed approach for online devices we suggest using knowledge distillation for lower latency. We also aim to modify the proposed approach to solve the said problem in low-resource settings. Finally, a well-trained model can help engineers get insights into the process of creating robust and stable synthetic mRNA molecules.

References

1. Organization WH, et al.. WHO Director-General's opening remarks at the media briefing on COVID-19-11 March 2020; 2020.

2. Payne S. Family coronaviridae. Viruses. 2017; p. 149.

3. Chagla Z. The BNT162b2 (BioNTech/Pfizer) vaccine had 95% efficacy against COVID-19≥ 7 days after the 2nd dose. Annals of internal medicine. 2021;174(2):JC15.

4. Khuroo MS, Khuroo M, Khuroo MS, Sofi AA, Khuroo NS. COVID-19 vaccines: a race against time in the middle of death and devastation! Journal of clinical and experimental hepatology. 2020;10(6):610–621.

5. Abbasi J. COVID-19 and mRNA vaccines—first large test for a new approach. Jama. 2020;324(12):1125–1127.

6. Tanne JH. Covid-19: FDA approves Pfizer-BioNTech vaccine in record time; 2021.

7. Jackson LA, Anderson EJ, Rouphael NG, Roberts PC, Makhene M, Coler RN, et al. An mRNA vaccine against SARS-CoV-2—preliminary report. New England Journal of Medicine. 2020;.

8. Marian AJ. Current state of vaccine development and targeted therapies for COVID-19: impact of basic science discoveries. Cardiovascular Pathology. 2021;50:107278.

9. Thomas K. New Pfizer results: Coronavirus vaccine is safe and 95% effective. The New York Times. 2020;18.

10. Holm MR, Poland GA. Critical aspects of packaging, storage, preparation, and administration of mRNA and adenovirus-vectored COVID-19 vaccines for optimal efficacy. Vaccine. 2021;39(3):457.

11. Frasconi P, Gori M, Sperduti A. A general framework for adaptive processing of data structures. IEEE transactions on Neural Networks. 1998;9(5):768–786.

12. Cho K, Van Merriënboer B, Bahdanau D, Bengio Y. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:14091259. 2014;.

13. Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997;9(8):1735–1780.

14. Gori M, Monfardini G, Scarselli F. A new model for learning in graph domains. In: Proceedings. 2005 IEEE international joint conference on neural networks. vol. 2; 2005. p. 729–734.

15. Goodfellow I, Bengio Y, Courville A. Deep learning. MIT press; 2016.

16. Duvenaud DK, Maclaurin D, Iparraguirre J, Bombarell R, Hirzel T, Aspuru-Guzik A, et al. Convolutional networks on graphs for learning molecular fingerprints. Advances in neural information processing systems. 2015;28.

17. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. Advances in neural information processing systems. 2017;30.

18. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:14090473. 2014;.

19. Keshavarzi Arshadi A, Webb J, Salem M, Cruz E, Calad-Thomson S, Ghadirian N, et al. Artificial intelligence for COVID-19 drug discovery and vaccine development. Frontiers in Artificial Intelligence. 2020;3:65.

20. Qaid TS, Mazaar H, Alqahtani MS, Raweh AA, Alakwaa W. Deep sequence modelling for predicting COVID-19 mRNA vaccine degradation. PeerJ Computer Science. 2021;7:e597.

21. Renardy M, Rogers RC. An introduction to partial differential equations. vol. 13. Springer Science & Business Media; 2006.

22. Wang Y. Predicting the Degradation of COVID-19 mRNA Vaccine with Graph Convolutional Networks. In: 2021 6th International Conference on Machine Learning Technologies; 2021. p. 111–116.