

Deep sequence modelling for predicting COVID-19 mRNA vaccine degradation

Talal S. Qaid^{1,2,*}, Hussein Mazaar^{3,*}, Mohammed S. Alqahtani⁴,
Abeer A. Raweh^{1,2} and Wafaa Alakwaa³

¹ Computer Science Department, College of Computer Science, King Khalid University, Abha, Saudi Arabia

² Faculty of Computer Science, Hodeidah University, Hodeidah, Yemen

³ Computer Science Department, College of Science & Arts in Tanumah, King Khalid University, Abha, Saudi Arabia

⁴ Radiological Sciences Department, College of Applied Medical Sciences, King Khalid University, Abha, Saudi Arabia

* These authors contributed equally to this work.

ABSTRACT

The worldwide coronavirus (COVID-19) pandemic made dramatic and rapid progress in the year 2020 and requires urgent global effort to accelerate the development of a vaccine to stop the daily infections and deaths. Several types of vaccine have been designed to teach the immune system how to fight off certain kinds of pathogens. mRNA vaccines are the most important candidate vaccines because of their capacity for rapid development, high potency, safe administration and potential for low-cost manufacture. mRNA vaccine acts by training the body to recognize and response to the proteins produced by disease-causing organisms such as viruses or bacteria. This type of vaccine is the fastest candidate to treat COVID-19 but it currently facing several limitations. In particular, it is a challenge to design stable mRNA molecules because of the inefficient in vivo delivery of mRNA, its tendency for spontaneous degradation and low protein expression levels. This work designed and implemented a sequence deep model based on bidirectional GRU and LSTM models applied on the Stanford COVID-19 mRNA vaccine dataset to predict the mRNA sequences responsible for degradation by predicting five reactivity values for every position in the sequence. Four of these values determine the likelihood of degradation with/without magnesium at high pH (pH 10) and high temperature (50 degrees Celsius) and the fifth reactivity value is used to determine the likely secondary structure of the RNA sample. The model relies on two types of features, namely numerical and categorical features, where the categorical features are extracted from the mRNA sequences, structure and predicted loop. These features are represented and encoded by numbers, and then, the features are extracted using embedding layer learning. There are five numerical features depending on the likelihood for each pair of nucleotides in the RNA. The model gives promising results because it predicts the five reactivity values with a validation mean columnwise root mean square error (MCRMSE) of 0.125 using LSTM model with augmentation and the codon encoding method. Codon encoding outperforms Base encoding in MCRMSE validation error using the LSTM model meanwhile Base encoding outperforms codon encoding due to less over-fitting and the difference between the training and validation loss error is 0.008.

Submitted 19 February 2021

Accepted 26 May 2021

Published 22 June 2021

Corresponding author

Hussein Mazaar,
hmazaar@kku.edu.sa

Academic editor

Ahmed Elazab

Additional Information and
Declarations can be found on
page 19

DOI 10.7717/peerj-cs.597

© Copyright
2021 Qaid et al.

Distributed under
Creative Commons CC-BY 4.0

OPEN ACCESS

Subjects Bioinformatics, Artificial Intelligence, Data Mining and Machine Learning, Data Science
Keywords COVID-19 Vaccine, mRNA, Recurrent neural networks (RNN), Gated recurrent unit (GRU), Long short time memory (LSTM), Embedding

INTRODUCTION

COVID-19 is the most important pandemic of the 21st century. It started in Wuhan, China, in December 2019. As of May 20, there have been more than 165 million confirmed COVID-19 cases and over 3.4 million deaths worldwide (*Chakraborty & Parvez, 2020*). In February 2020, it has been considered a pandemic by the World Health Organization (WHO) due to its global impact. Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) causes COVID-19 disease, and it is similar to other coronaviruses that have appeared in the past 2 decades, namely, the Middle East respiratory syndrome coronavirus (MERS-CoV) and severe acute respiratory distress syndrome coronavirus (SARS-CoV).

A large-scale search for powerful medications effective against the COVID-19 coronavirus has been undertaken but to date has not produced any results. Therefore, it is necessary to develop a vaccine because vaccination is an efficient approach to stop the pandemic with 95% effectiveness and is available to entire countries at the lowest cost. The vaccine activates the immune system of the body to identify and resist pathogenic agents such as bacteria, viruses and any related microorganisms (*Khuroo et al., 2020*). There are two vaccine designs: gene based and protein based. A protein-based vaccine teaches the immune system to fight the viruses or bacteria while the gene-based vaccines mimic the natural infection by holding the genetic instructions of the cell to generate the antigen, and particularly the surface spike protein in the case of coronaviruses (*Abbasi, 2020*). Many academic institutions and companies are working on COVID-19 vaccines with different strategies including mRNA in lipid nanoparticles, DNA, recombinant vectors, and proteins. Some vaccines such as mRNA have reached advanced phases testing; these vaccines encode the viral spike protein by mRNA as an antigen to elicit immune system response and produce neutralizing antibodies (*Sempowski et al., 2020*).

mRNA is one of the most promising and efficient approaches to the development of vaccines for COVID-19 and is based on the idea that the S protein in SARS-CoV can be used as the same mechanism to be applied on SARS-Cov-2 to build efficient vaccines against COVID-19 (*Marian, 2021; Jackson et al., 2020*). The mRNA-1273 vaccine is the first vaccine against COVID-19 (*Jackson et al., 2020*). It has many advantages such as an easy development process, low time required to develop, reduced risk of pre-existing immunity against the vaccine, speed, ease of automation, easy development of multiple prototype vaccines, and a less expensive process. Moreover, the greatest benefits of mRNA are safety and its non-infectious nature due to its direct injection and translation from messenger RNA into protein in the human body (*Synced, 2020*).

mRNA vaccines have become the leading candidates for immunization for COVID-19, but currently they face key potential restrictions. Designing stable mRNA molecules is currently the greatest challenge. It has been observed that RNA particles degrade suddenly; this is a significant limitation that means that the mRNA vaccine is useless if the mRNA suffers even one cut. Moreover, the mRNA vaccine for COVID-19 must be stored and

transported under heavy refrigeration due to little knowledge of details where the backbone of RNA affected. Enhancing the stability of mRNA was a steep challenge in mRNA vaccine development prior to the pandemic, and the delivery of a refrigerator-stable vaccine against SARS-CoV-2 (COVID-19) is an important objective.

In this paper, we seek to use modern data science techniques, in particular, the deep learning paradigm, to design models and rules for predicting the degradation in mRNA molecules. The model will predict the likelihood of the degradation rates for each position inside the mRNA molecule. The model was trained on the Eterna dataset composed of 3000 mRNA sequences and structures and the scoring and predictions tested at Stanford University in parallel to our modeling ([Stanford University, 2020](#)).

The rest of this paper is organized as follows. A review of the literature for the latest research on COVID-19 is presented in “Literature Review”. Another part of the dataset used in this paper is described in “DataSet”. The proposed model architecture is described in detail in “Model Architecture”. The simulation and discussion are summarized in “Simulation and Discussions”, and the paper is concluded in “Conclusion and Future Works”.

LITERATURE REVIEW

Vaccines save millions of lives each year, and according to the WHO, currently prevent 2–3 million deaths. There are currently over 169 COVID-19 vaccine candidates under development, with 26 of these in the human trial phase ([World Health Organization \(WHO\), 2020](#)).

Artificial intelligence and machine learning, particularly deep learning, have led to huge improvements in many fields of science and engineering because of their ability to learn features deeply. Vaccine discovery has been the most highly impacted area ([Keshavarzi Arshadi et al., 2020](#)). Recently, some powerful deep learning techniques as LSTM and GRU have been applied to the field of DNA and RNA sequence modeling. Artificial intelligence can be utilized to fight against COVID-19 pandemic and find a solution for the different areas such as drug discovery, vaccine development, public communications, and integrative medicine ([Ahuja, Reddy & Marques, 2020](#)).

[Abbasi \(2020\)](#) and [Zhang et al. \(2020\)](#) suggested that mRNA-based vaccines have emerged as a rapid and versatile platform to quickly respond to the challenge of COVID-19 pandemic and the success of mRNA vaccine had lead to the use of this approach in a variety of fields that are far away from COVID-19 to establish a broad platform for use against both other known and emerging pathogens.

[Ong et al. \(2020\)](#) tested some proteins for vaccine development against SARS and MERS. They used a machine learning Vaxign-ML reverse vaccinology tool and predicted six structural proteins with other five unstructured proteins to be adhesins that are crucial to the viral adhering and host invasion. They also provide a review of the current status of the coronavirus vaccine and found that there were only three SARS-CoV, six MERS-CoV and six SARS-CoV2 vaccine in clinical trials. Only one of these 15 vaccines was an mRNA-based vaccine (S protein) produced in the United Kingdom for

SARS-CoV2, whereas [Wang et al. \(2020\)](#) reviewed 13 COVID-19 vaccines of which two are mRNA-based vaccines, the first developed by Moderna/NIH and the second by Pfizer/BioNTech, the vaccines candidates are mRNA1273 and BNT162b2. They mentioned that chemical modifications of the mRNA molecules may alter their proinflammatory activity, but the delivery vehicles and the mRNA condensing lipids can both induce unwanted proinflammatory responses.

[Jackson et al. \(2020\)](#) and [Lu et al. \(2020\)](#) conducted a phase1 open-label trial for mRNA-1273 vaccine on 45 healthy adults 18–55 years of age with different doses and concluded that the mRNA-1273 vaccine induced anti-SARS-CoV-2 immune responses in all participants, and no trial-limiting safety concerns were identified. On the other hand, [Lu et al. \(2020\)](#) mentioned that coronavirus virus-like particles assembly requires at least three structural proteins: S, M, and E and designed three mRNA vaccine candidates for COVID-19, and they encode various forms of antigens in vaccinated hosts.

Pfizer (New York, NY, USA) and BioNTech (Germany) introduced the BNT162b2 vaccine candidate based on mRNA. The vaccine candidate advanced to phase 2/3 study. They claim that their decision to select this candidate reflects the primary goal to bring a well-tolerated, highly effective vaccine to market as quickly as possible ([Pfizer, 2020](#)).

DATASET

A Stanford COVID-19 mRNA vaccine dataset uploaded to Kaggle for competition is used as the primary dataset. It has two different types of sequence data and bpps data ([Stanford University, 2020](#)).

Sequences data

The sequence data include two files for training and testing as shown in [Table 1](#). The training file contains 19 fields, and the test file contains seven fields. The features fields are sequence, structure and predicted_loop_type; these fields describe the RNA sequence, whether a base is paired or unpaired and the structural context, respectively. The sequence is a combination of A, G, U, and C for each sample, and the structure is an array of (,), and ‘.’. The predicted loop type field is a set of characters where each character in sequence as M: Multi-loop S: paired “Stem” B: Bulge I: Internal loop H: Hairpin loop X: eXternal loop E: dangling End. All of these fields have a length of 107 characters, but only 68 bases are scored.

The predicted fields are: reactivity which determines the likely secondary structure of the RNA, deg_Mg_pH10 and deg_pH10 that determine the probability of degradation at the base after incubating with/without magnesium at high pH (pH 10). deg_Mg_50C and deg_50C determine the probability of degradation at the base after incubating with/without magnesium at high temperature (50 degrees Celsius) ([Stanford University, 2020](#)).

BPPS data

The bpps data contains 6,034 .npy files. The bpps data are symmetric square matrices pre-calculated for each sequence with the same length as the sequence. This matrix gives

Table 1 mRNA COVID-19 vaccine dataset.

File name	No. of samples	Sequences length	Sequences scored
Train	2,400	107	68
Augmented	2,400	107	68

the probability that each pair of nucleotides in the RNA forms a base pair in the ensemble of RNA secondary structures so that it provides more robust and rich information about the structures than a single RNA secondary structure (*World Health Organization (WHO), 2020; Keshavarzi Arshadi et al., 2020*).

Augmented data

Data augmentation is a highly important process for increasing the number of the training samples and overcoming the problem of over-fitting. In the COVID-19 dataset, we need to predict the sequence, structure, and loops. Many packages to carry out these tasks are available, such as ARNIE, which is an API library in python. ARNIE supports multiple secondary structure packages. The size of augmented data is 2,400 which equals the original data.

MODEL ARCHITECTURE

The proposed model is based on sequence models. It is composed of bidirectional GRU, LSTM, and Hybrid models applied on mRNA sequence data and bpps data discussed under Dataset. The model first starts with feature engineering to extract the features and apply a sequence model to predict the mRNA sequences responsible for the degradation by predicting five reactivity values for every position in the sequence. The features engineering extracts two types of features called numerical features and categorical features. First, the categorical features are encoded, and an embedding layer is adopted to capture the relationships in the sequences that are very difficult to capture otherwise as shown in Fig. 1. Then, feature extraction is applied to extract numerical features using statistical and mathematical equations and categorical features as per shown in Figs. 2 and 3. Finally, the two types of features are concatenated to apply the bidirectional model as shown in Fig. 4.

Features engineering

This work applies two approaches to discover and create the most important features based on the existing features to improve the predictive model performance. The following two sections explain the two types of features.

Numerical features

We tested many numerical formulae and the formulae selected based on the empirical results. All of the following formulae rely on the pre-calculated base pair probabilities in bpps matrices for each sequence. Every position in the sequence has a vector called bpp,

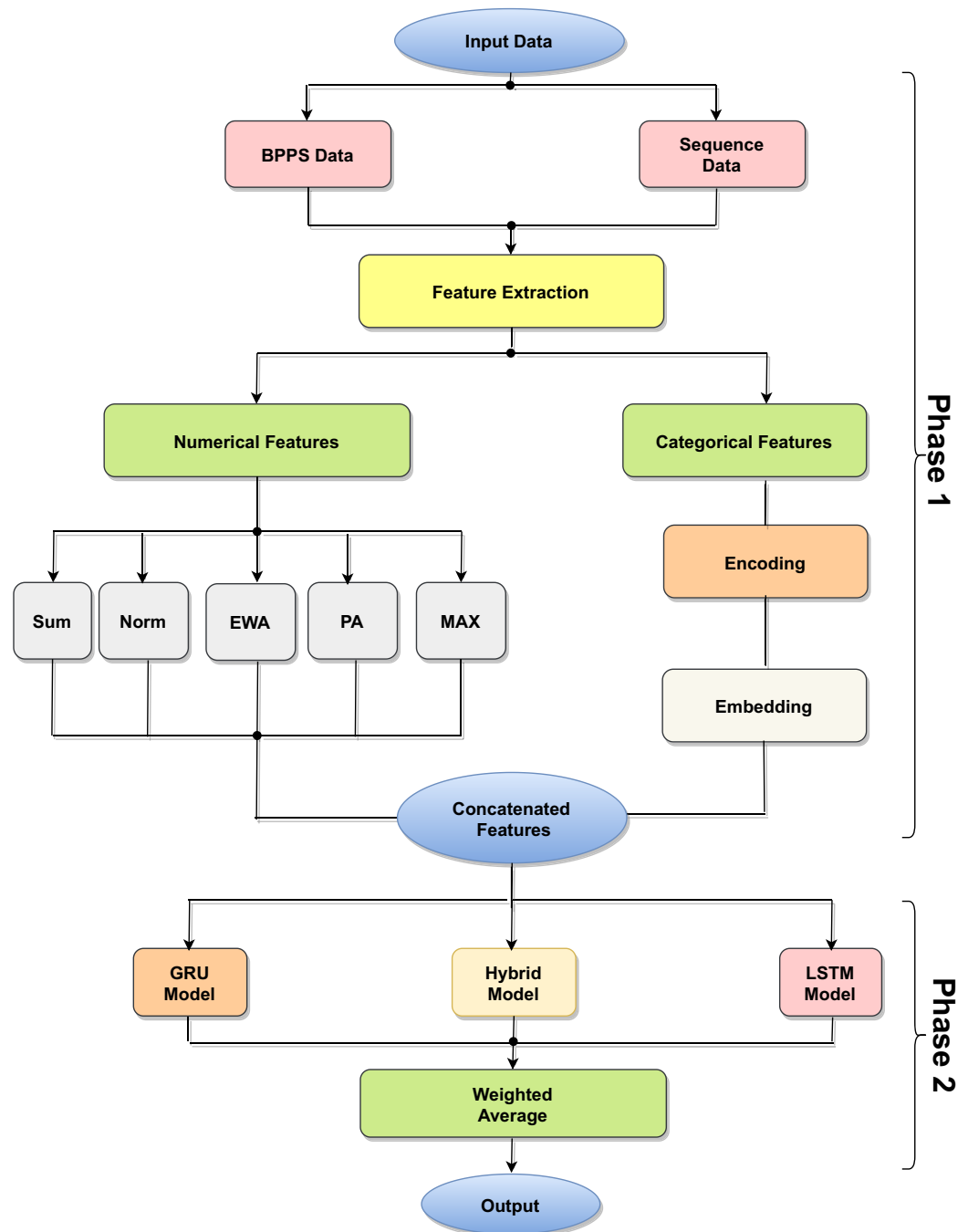


Figure 1 Model architecture process.

Full-size  DOI: 10.7717/peerj-cs.597/fig-1

which is a column in the bpps matrix. Here, n is the length of the bpp vector, and $bppi$ is the probability of the given base with the position i for all the following formulae.

$$Sum = \sum_{i=1}^n bppi \quad (1)$$

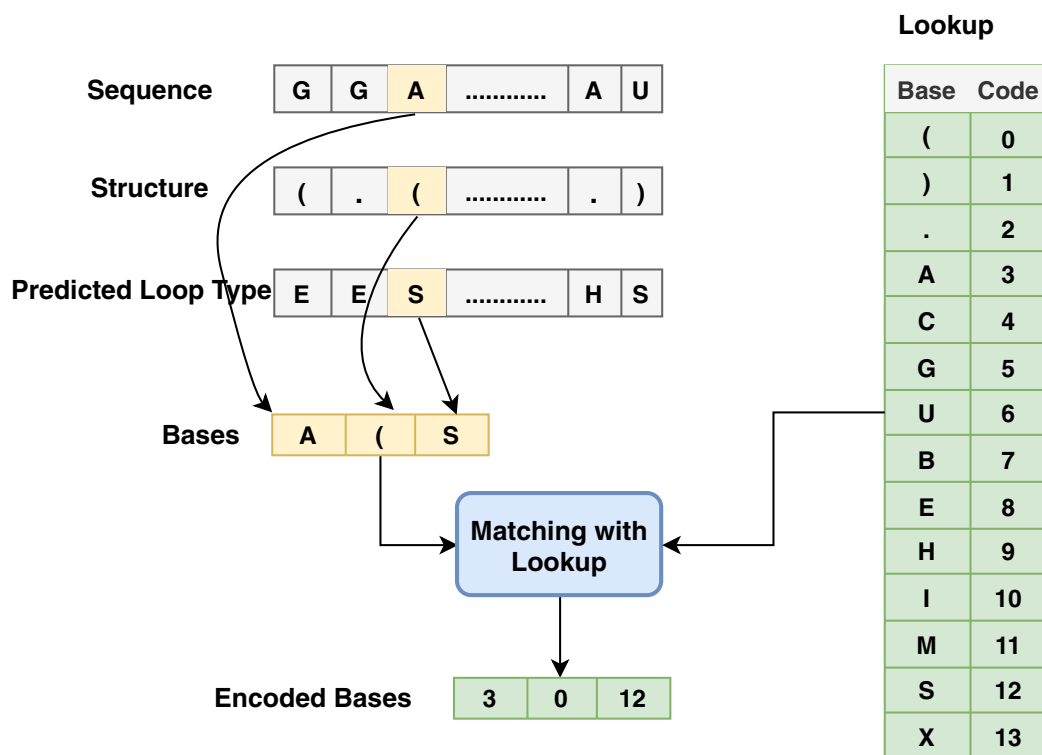


Figure 2 Base method steps for the encoding process associated with sequence, structure and predicted loop type data. [Full-size !\[\]\(fd7fe780e8fd8eece60268c87d0c3e04_img.jpg\) DOI: 10.7717/peerj-cs.597/fig-2](https://doi.org/10.7717/peerj-cs.597/fig-2)

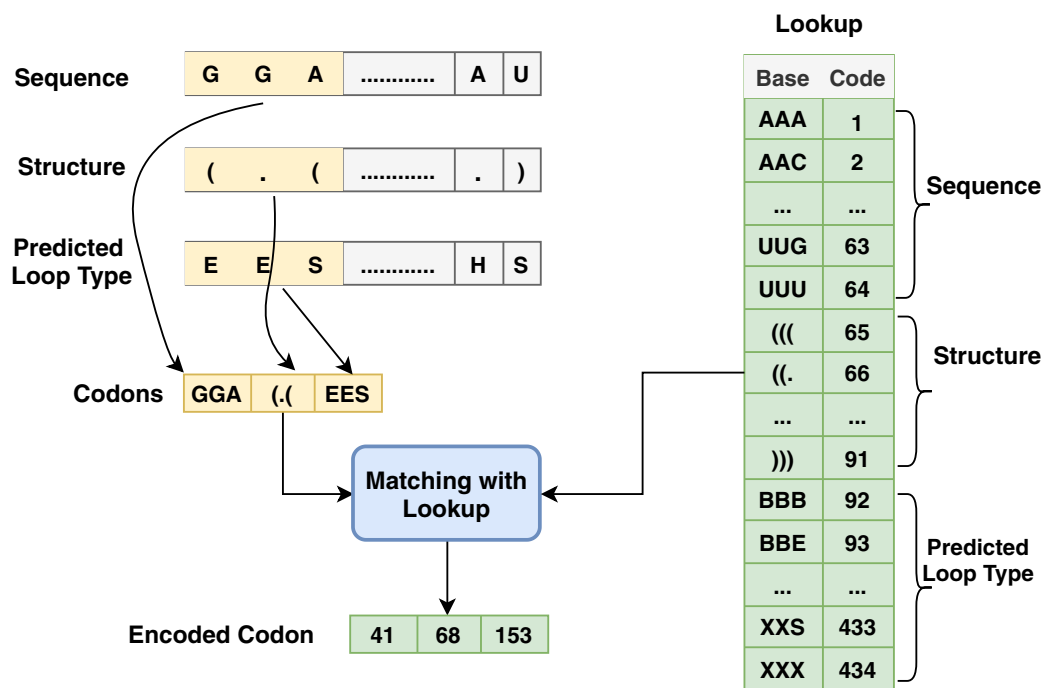


Figure 3 Codon method steps for encoding process associated with sequence, structure and predicted loop type data. [Full-size !\[\]\(86257f54800c9844bc7e863bea396fba_img.jpg\) DOI: 10.7717/peerj-cs.597/fig-3](https://doi.org/10.7717/peerj-cs.597/fig-3)

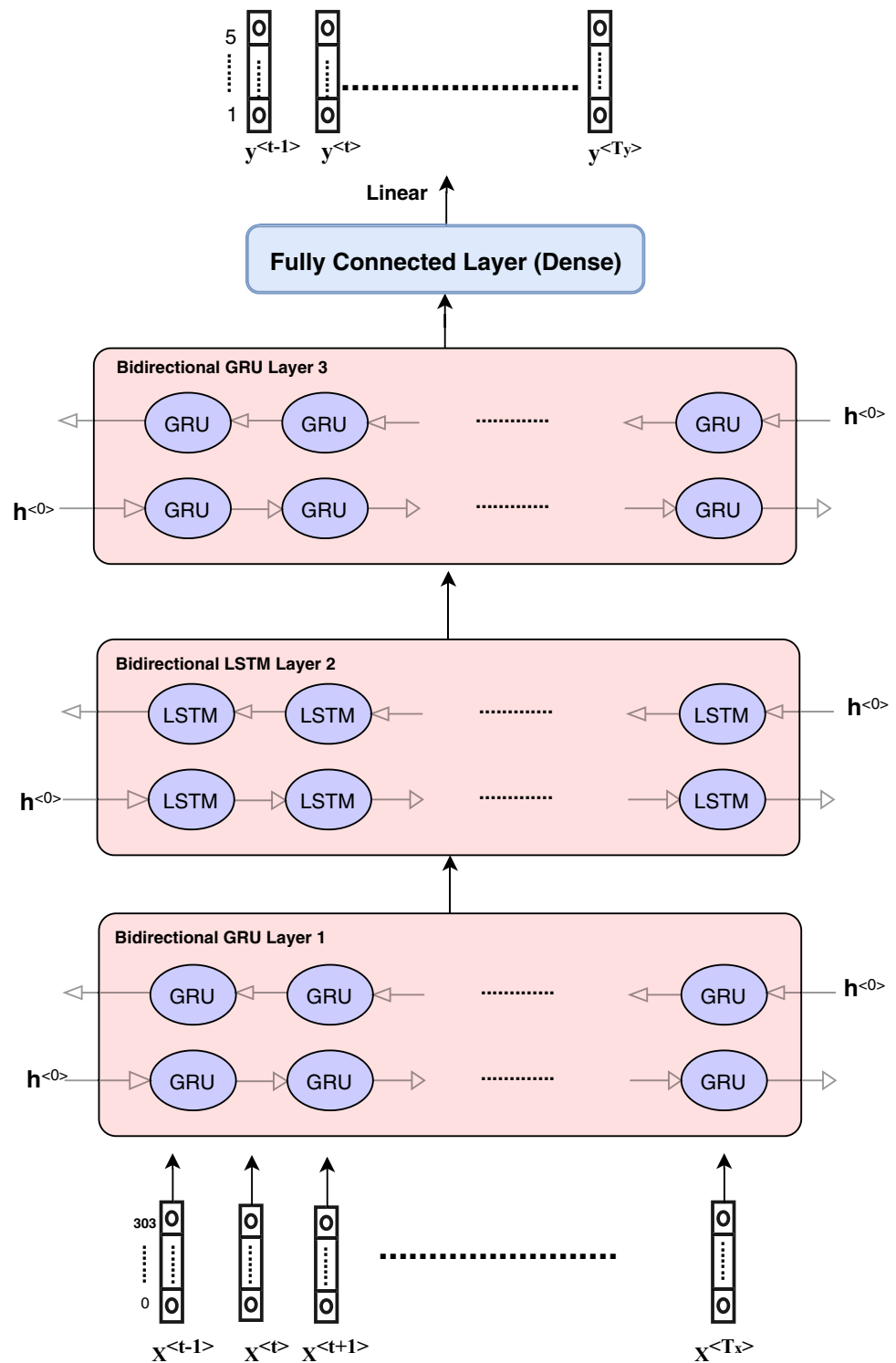


Figure 4 Architecture of hybrid bidirectional GRU and LSTM models.

Full-size  DOI: 10.7717/peerj-cs.597/fig-4

$$Max = \max(bpp) \quad (2)$$

The following formula normalizes the average μ based on all bpps matrix values and standard deviation σ .

$$Norm = \frac{\left(\frac{\sum_{i=1}^n bppi}{n} \right) - \mu}{\sigma} \quad (3)$$

This formula is the exponential weighted average (EWA) for every vector of probabilities in the bpps matrix for a given base.

$$EWA = \sum_{i=1}^{n+1} V_i \quad (4)$$

$$V_i = \beta V_{i-1} + (1 - \beta) bppi_{i-1} \quad (5)$$

where V_i is a weighted probability for every position i in the bpp vector and β is a constant value equal to 0.9

$$PA = \frac{\sum_{i=1}^n i * bpp(i)}{n} \quad (6)$$

This feature describes the position multiplying the value of position in bpp to provide the position average value.

Categorical features

Since most machine learning techniques accept numerical features only, preprocessing and preparing the data to convert the categorical features into numerical features is a necessary step for enabling the technique to process the data and extract valuable information.

In the mRNA vaccine data, there are three types of sequence data to be encoded using **base** method: sequence (that has AGUC characters) encoded from 3 to 6, structure (which has (). Characters) encoded from 0 to 2 and modified-loop (which has BEHIMSX characters) encoded from 7 to 13 as shown in Fig. 2.

Another type of encoding used in this work depends on codon-based encoding which used in many researches (Zhang et al., 2019; Hu et al., 2020) as an initial representation of the data and it provides deep learning techniques to extract high level features and capture the long dependency within the sequences. Each group of codon-based method has three bases in mRNA that constitutes a codon where each codon specifies a particular amino acid. The chain of amino acids forms a protein during mRNA translation. The sequence has AGUC characters and is encoded from 1 to 64, and the structure has (). characters and is encoded from 65 to 91; meanwhile, predicted-loop has BEHIMSX characters and is encoded from 92 to 434 as shown in Fig. 3.

After encoding, the resulting codes are fed into the embedding layer, which is a powerful feature that allows additional information to be automatically inserted into the neural network. Embedding layers are often used in natural language processing (NLP); however,

they can be used in this work because we can insert a larger vector instead of the index values. Embedding layer is a dimension expansion that provides more information to the model. In this work, the embedding layer extracts 300 features instead of the 14 input encoding features, where the bases are represented by dense vectors that represent the projection of the base into a continuous vector space. The position of a base within the vector space is learned from the mRNA sequence and is based on the bases that surround the base when it is used.

Sequence modeling

Sequence modeling, specifically recurrent neural networks (RNNs), is designed to utilize the structure data or sequence data. It has an inner state to read input sequence and allow RNNs to capture the interactions between the different elements through the mRNA sequences and others. There are two types of RNNs: long short term memory units (LSTM) and gated recurrent units (GRU). LSTM is a complex type of RNNs proposed by [Hochreiter & Schmidhuber \(1997\)](#) and can solve sequential complicated hierarchical decomposition problems that cannot be solved with RNNs. It is an effective way to capture long-term sequential dependencies. Moreover, it does not suffer from any optimization in contrast to simple recurrent networks (SRNs). GRU is a new technique proposed by [Cho et al. \(2014\)](#). It is similar to LSTM for capturing the dependencies with varied time scale, and also can solve the vanishing gradient problems by using the update gate and reset gate.

The proposed model architecture is described in [Fig. 4](#) and consists of different layers and models. The first model is composed of three bidirectional GRU layers and is called the GRU model. The second model is composed of three bidirectional LSTM layers and is called the LSTM model. The third model is a hybrid model in which GRU and LSTM are used together as the bidirectional GRU layer, bidirectional GRU layer, and bidirectional LSTM layer, respectively. The output is weighted for all models as an ensemble model that provides better results.

The hybrid model consists of the GRU and LSTM units. After data pre-processing steps, encoding and embedding processes are prepared and described in the 'Features engineering' section, and the features are ready to enter the sequence modeling as inputs. Each sequence or sample includes input dimension as ($T_x, Vector_Length$) such that $T_x = 107$ and $vector_length$ depends on the number of numerical features and categorical features. The categorical features length equals to 300 and numerical features length is equal to 5.

The hidden units at each layer are 256 in each direction, which means that the number of bidirectional layer units is equal to 512. The bidirectional layer is used to optimize the results as the data are passed in the forward and backward directions to more flexibly capture the information in the sequence data. The hybrid model is presented in detail in [Fig. 4](#). The GRU model can be described according to the layers and parameters in [Table 2](#). Other models have the same architecture and parameters.

For each time t , $X_t \in R^{m \times d}$ is the mini-batch input (number of observations: m , number of inputs: d) and σ is the hidden layer activation function. We suggest the forward and backward hidden states in the bidirectional architecture for this time step are

Table 2 GRU sequence model summary for layers, output shape and number of parameters in each layer (keras summary function).

Layer (type)	Output shape	Number of parameters
input_2 (Input Layer)	[(None, 107, 8)]	0
tf_op_layer_strided_slice_2	[(None, 107, 3)]	0
embedding (Embedding)	(None, 107, 3, 100)	1,400
tf_op_layer_Reshape (TensorFlow)	[(None, 107, 300)]	0
tf_op_layer_strided_slice_1(TF)	[(None, 107, 3)]	0
concatenate (Concatenate)	(None, 107, 305)	0
bidirectional (Bidirectional)	(None, 107, 512)	861,696
bidirectional_1 (Bidirectional)	(None, 107, 512)	1,182,720
bidirectional_2 (Bidirectional)	(None, 107, 512)	1,182,720
tf_op_layer_strided_slice_4	[(None, 68, 512)]	0
dense (Dense)	(None, 68, 5)	2,565
Total params: 3,231,101		
Trainable params: 3,231,101		
Non-trainable params: 0		

$\vec{H}_t \in \mathbb{R}^{m \times h}$ and $\overleftarrow{H}_t \in \mathbb{R}^{m \times h}$ respectively. Here h refers to the number of hidden units. The forward and backward hidden state updates are computed as follows:

$$\vec{H}_t = \sigma(X_t W_{xh}^{(f)} + \vec{H}_{t-1} W_{hh}^{(f)} + b_h^{(f)}), \quad (7)$$

$$\overleftarrow{H}_t = \sigma(X_t W_{xh}^{(b)} + \overleftarrow{H}_{t+1} W_{hh}^{(b)} + b_h^{(b)}), \quad (8)$$

where the weight parameters $W_{xh}^{(f)} \in \mathbb{R}^{d \times h}$, $W_{hh}^{(f)} \in \mathbb{R}^{h \times h}$, $W_{xh}^{(b)} \in \mathbb{R}^{d \times h}$, and $W_{hh}^{(b)} \in \mathbb{R}^{h \times h}$ and bias parameters $b_h^{(f)} \in \mathbb{R}^{1 \times h}$ and $b_h^{(b)} \in \mathbb{R}^{1 \times h}$ are all model parameters.

Then, the forward and backward hidden states are concatenated \vec{H}_t and \overleftarrow{H}_t to build the hidden state $H_t \in \mathbb{R}^{n \times 2h}$ and input it to the output layer. The data are transferred as inputs to the next bidirectional layer in deep bidirectional RNNs. Finally, $Y_t \in \mathbb{R}^{m \times q}$ (m : number of observations and q : number of outputs) is the output layer and is calculated as follows:

$$Y_t = H_t W_{hq} + b_q, \quad (9)$$

where the model parameters are as follows: $W_{hq} \in \mathbb{R}^{2h \times q}$ is the weight parameter. $b_q \in \mathbb{R}^{1 \times q}$ is bias parameter of the output layer. The number of hidden units can be different in the bidirectional model.

Accordingly, the information from past and future is utilized to estimate the current state. This means that the output is estimated by the the information from both ends of the sequence which is the main feature in the bidirectional RNN. The global parameters that are used and configured in each bidirectional layers are summarized in Table 3.

The dropout process is a regularization technique used to overcome the over-fitting problem in a learning process such that the model can generalize the results in stable

Table 3 Bidirectional configuration parameters.

Parameter	Value
Hidden units	256
Dropout	0.4
Return sequences	True
Kernel initializer	'orthogonal'

manner with less error through testing and scoring in production. In this method, random neurons are dropped out and ignored during training while other neurons are used to represent the predictions. This makes the network less sensitive to the particular weights so that it can generalize better in testing and avoid the over-fitting problem.

Finally, the dense layer has five outputs and the activation function is linear because the problem is a regression problem. The optimizer is 'Adam', which is an efficient optimization method to update the weights in an efficient manner. It is used in deep learning due to its many advantages with respect to computation, memory, simplicity, and because it is appropriate for noisy data and non-stationary objectives. The loss function is MCRMSE as described in Eq. (10).

SIMULATION AND DISCUSSIONS

The experiments are applied on the Stanford COVID-19 mRNA vaccine dataset as described in 'Dataset'. The features are the combination of numerical features with length of 5 as described in 'Numerical features' and categorical features extracted from encoding and embedding processes, respectively, with embed_dim = 100 for each character from the sequence, structure, and predicted loop type and total concatenated length = 300 as described in 'Categorical features'.

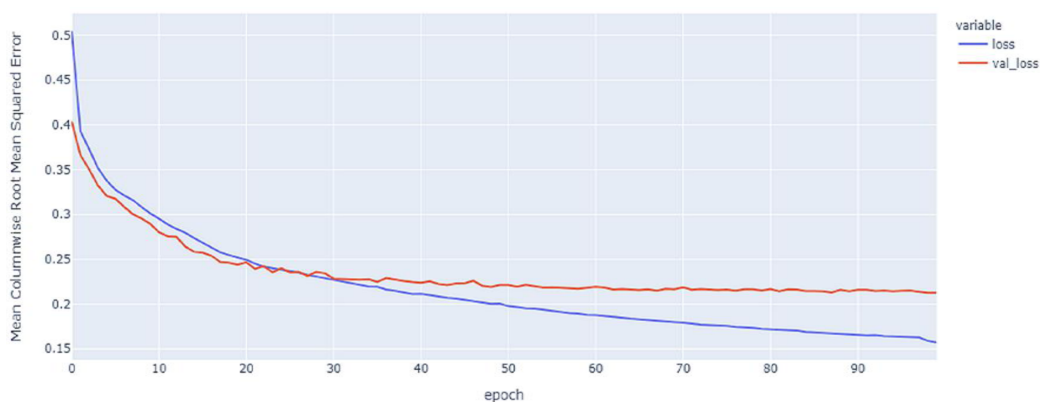
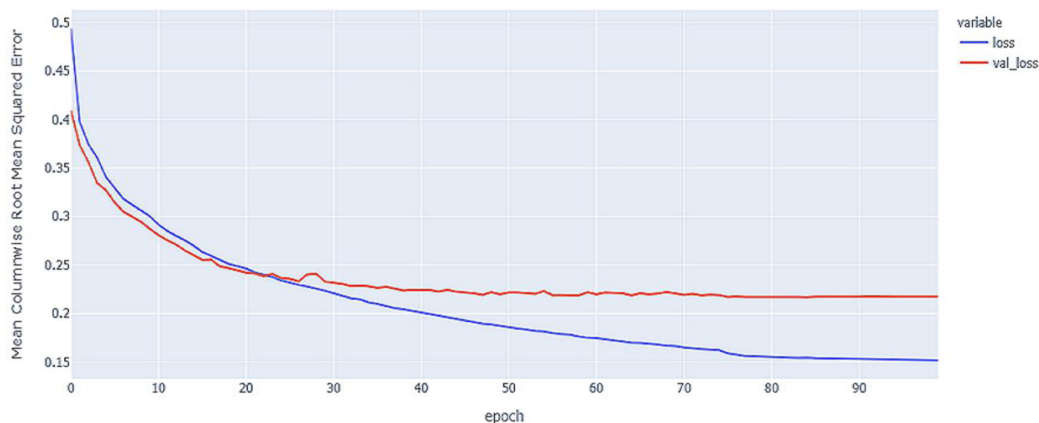
The experiments are performed in the GPU environment. The training data have 2,400 samples or observations. We filter the training data with the signal_to_noise feature to exclude the noisy samples that have values ≤ 1 . Signal_to_noise is defined as mean (measurement value over 68 nts)/mean (statistical error in measurement value over 68 nts). Therefore, after applying the filter, there are 2096 training data. To reduce the complexity, we split the data into the training data containing 1886 samples for the construction of the model and the validation data containing 210 samples for the tuning and monitoring of the model, and to select the best model. The final size for the training data set after embedding is (1,886, 107, 8). The most important configuration of the model is: seq_len = 107, pred_len = 68, dropout = 0.5, embed_dim = 100, and hidden_dim = 256.

The model predicts five target columns of reactivity, deg_Mg_pH10, deg_pH10, deg_Mg_50C, and deg_50C and the measure of evaluation is the mean columnwise root mean squared error (MCRMSE). It is given by:

$$MCRMSE = \frac{1}{N_t} \sum_{j=1}^{N_t} \sqrt{\frac{1}{n} \sum_{i=1}^n (y_{ij} - \hat{y}_{ij})^2} \quad (10)$$

Table 4 MCRMSE results of the sequence models based on the numerical and categorical features and base encoding method without augmentation.

Model name	Training data	Validation data
GRU	0.157	0.213
LSTM	0.152	0.217
HYBRID	0.157	0.214
Weighted average	0.155	0.215

**Figure 5** GRU model MCRMSE results on the categorical and numerical features without augmentation based on base encoding. [Full-size](#) DOI: 10.7717/peerj-cs.597/fig-5**Figure 6** LSTM model MCRMSE results on the categorical and numerical features without augmentation based on base encoding. [Full-size](#) DOI: 10.7717/peerj-cs.597/fig-6

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

As shown in Table 4 and Figs. 5, 6, 7, these results describe the change in MCRMSE with increasing number of epochs in our experiments using original data without augmentation and using Base method encoding and the GRU, LSTM, and Hybrid models. In Table 5 and Figs. 8, 9, 10, the MCRMSE are presented for the different number of epochs in our

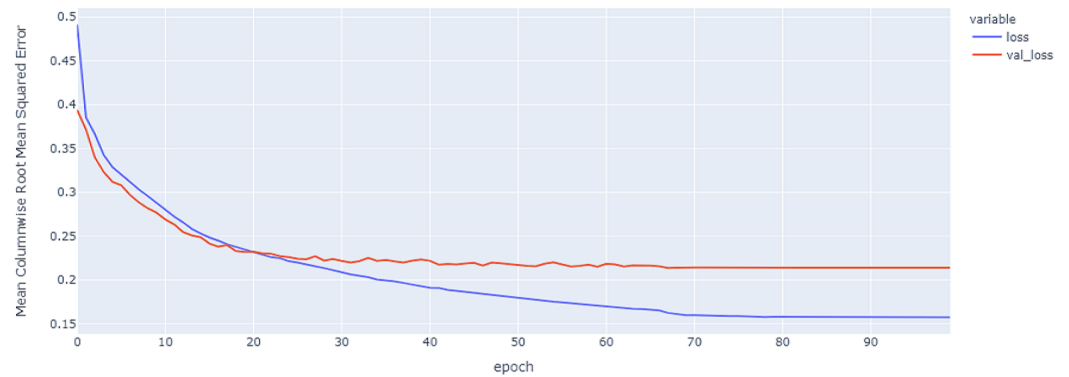


Figure 7 Hybrid LSTM model MCRMSE results on the categorical and numerical features without augmentation based on base encoding. [Full-size !\[\]\(1679558f37f6db0dd8360a2a7e913e90_img.jpg\) DOI: 10.7717/peerj-cs.597/fig-7](https://doi.org/10.7717/peerj-cs.597/fig-7)

Table 5 MCRMSE results of the sequence models based on the numerical and categorical features and codon encoding method without augmentation.

Model name	Training data	Validation data
GRU	0.158	0.201
LSTM	0.133	0.206
HYBRID	0.131	0.138
Weighted average	0.141	0.182

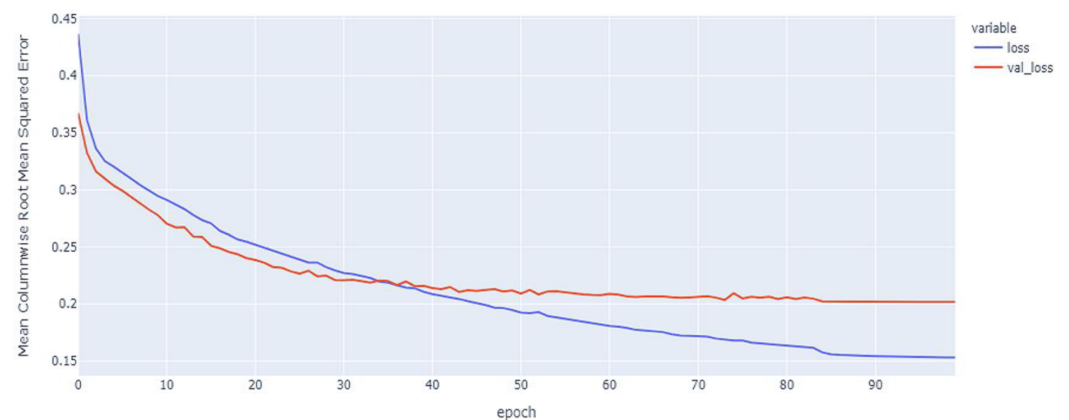


Figure 8 GRU model MCRMSE results on the categorical and numerical features without augmentation based on codon encoding. [Full-size !\[\]\(759816eba407ea14e648418beb667f53_img.jpg\) DOI: 10.7717/peerj-cs.597/fig-8](https://doi.org/10.7717/peerj-cs.597/fig-8)

experiments using original data without augmentation and using codon method encoding and the GRU, LSTM, and Hybrid models.

The previous experiments suffer from over-fitting, and there is a large difference between the training loss error and validation loss error. This over-fitting occurs when the model fits too well to the training set because of the increasing number of features comparing with the smaller number of samples. In this paper, two different approaches are used to solve this problem: the first approach is dropout regularization for reducing over-

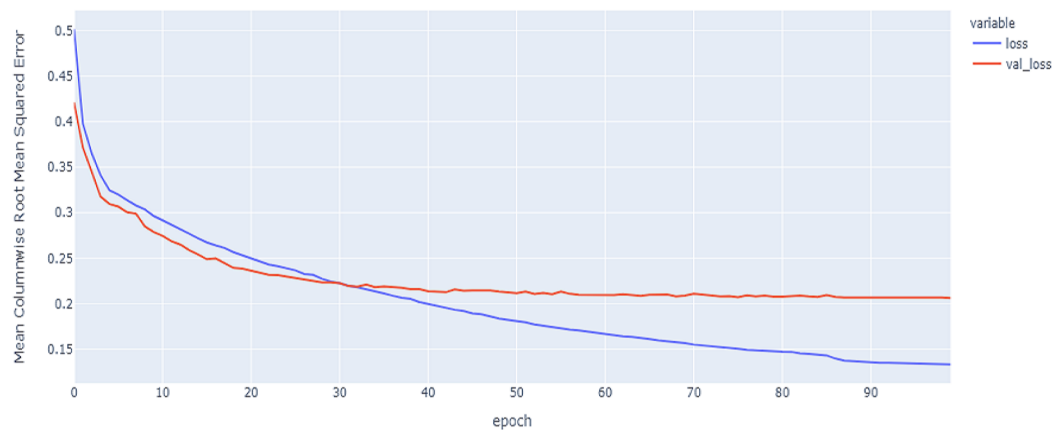


Figure 9 LSTM model MCRMSE results on the categorical and numerical features without augmentation based on codon encoding. [Full-size !\[\]\(fd7fe780e8fd8eece60268c87d0c3e04_img.jpg\) DOI: 10.7717/peerj-cs.597/fig-9](https://doi.org/10.7717/peerj-cs.597/fig-9)

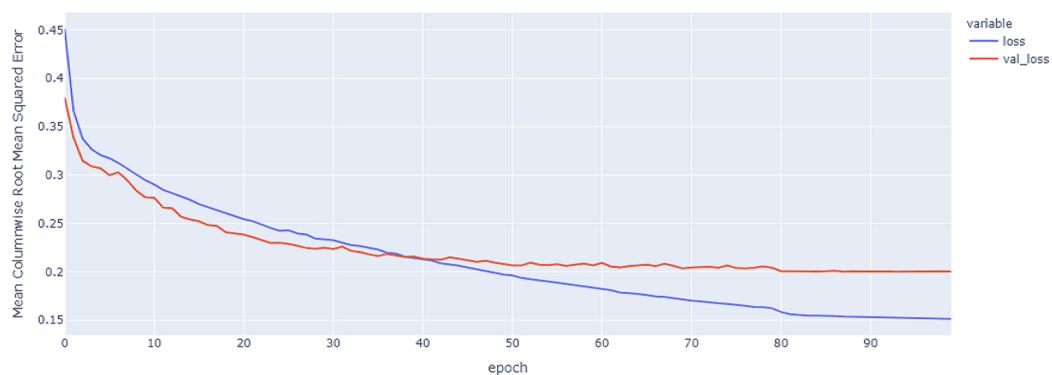


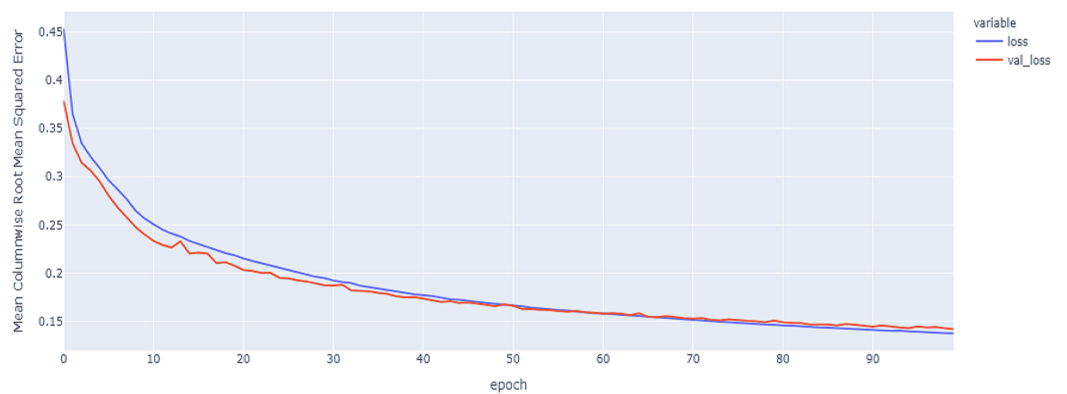
Figure 10 Hybrid model MCRMSE results on the categorical and numerical features without augmentation based on codon encoding. [Full-size !\[\]\(86257f54800c9844bc7e863bea396fba_img.jpg\) DOI: 10.7717/peerj-cs.597/fig-10](https://doi.org/10.7717/peerj-cs.597/fig-10)

fitting and improving the generalization of deep neural networks. The network becomes less sensitive to the specific weights of neurons and becomes more capable of better generalization and is less likely to overfit the training data. The second approach used data augmentation to increase the number of samples. This approach increases or augments the diversity of the data so that at each training stage, the model encounters a different version of the original data.

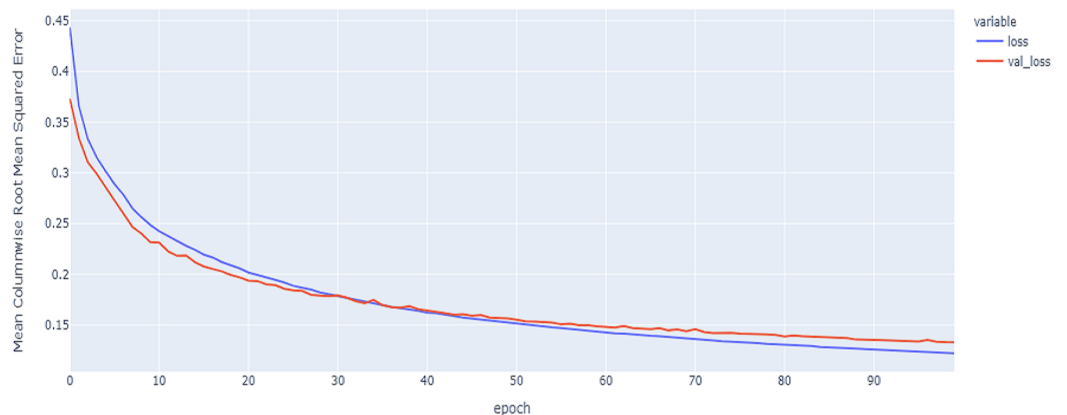
In our implementations for data augmentation, we used the Vienna package to obtain the required data ([tito:kaggle, 2020](https://doi.org/10.26434/chemrxiv-2020-08-01)). The augmented data are concatenated with the original data to obtain 4,800 samples. The noisy data are excluded based on the signal_to_noise variable. Any sample that is ≤ 1 will be excluded. The training set size is 4,192. To reduce the complexity, we split the data into the training data with 3,772 samples to for the construction of the model and validation data with 420 samples to tune and monitor the models and select the best model. The final size for the training data after embedding is equal to: (3,772, 107, 8). As shown in [Table 6](#) and [Figs. 11, 12 and 13](#), the results of MCRMSE applied on e thGRU, LSTM, and Hyprid models based on Base encoding after augmentation and validation and training samples are closer and no over-fitting is

Table 6 MCRMSE results of sequence models based on numerical and categorical features and base encoding after augmentation.

Model name	Training data	Validation data
GRU	0.138	0.142
LSTM	0.122	0.133
HYBRID	0.131	0.138
Weighted average	0.130	0.138

**Figure 11** GRU model MCRMSE results on the categorical and numerical features and base encoding after augmentation.

Full-size DOI: 10.7717/peerj-cs.597/fig-11

**Figure 12** LSTM model MCRMSE results on the categorical and numerical features and base encoding after augmentation.

Full-size DOI: 10.7717/peerj-cs.597/fig-12

observed. As shown in Table 7 and Figs. 14, 15 and 16, the results describe MCRMSE as a function of the number of epochs based on codon encoding after augmentation. Accordingly, augmentation resolved the issue and the model can generalize the prediction in production and real work.

The best model in our experiments is the LSTM model, which provides an MCRMSE of 0.125 using the codon encoding method and concatenated data of numerical and

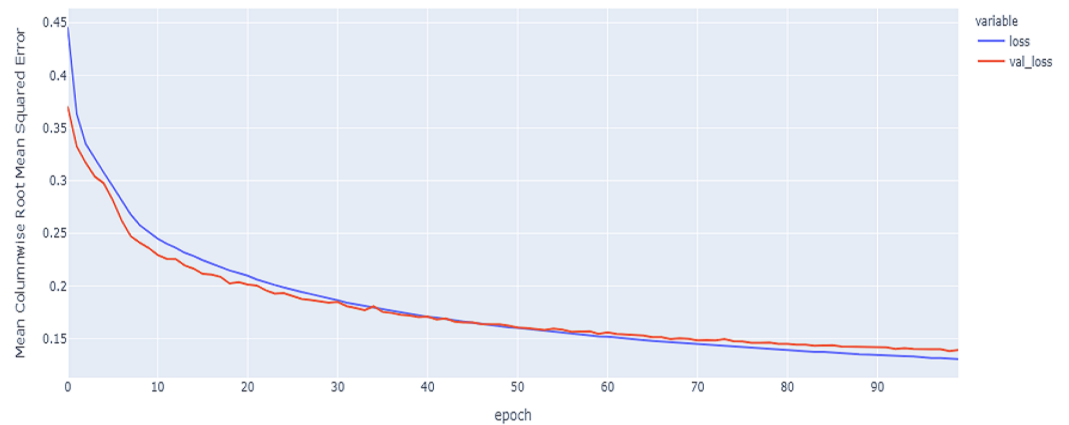


Figure 13 Hybrid model MCRMSE results on the categorical and numerical features and base encoding after augmentation. [Full-size](#) DOI: 10.7717/peerj-cs.597/fig-13

Table 7 MCRMSE results of sequence models based on the numerical and categorical features based on codon encoding method with augmentation.

Model name	Training data	Validation data
GRU	0.125	0.139
LSTM	0.109	0.125
HYBRID	0.112	0.128
Weighted average	0.115	0.131

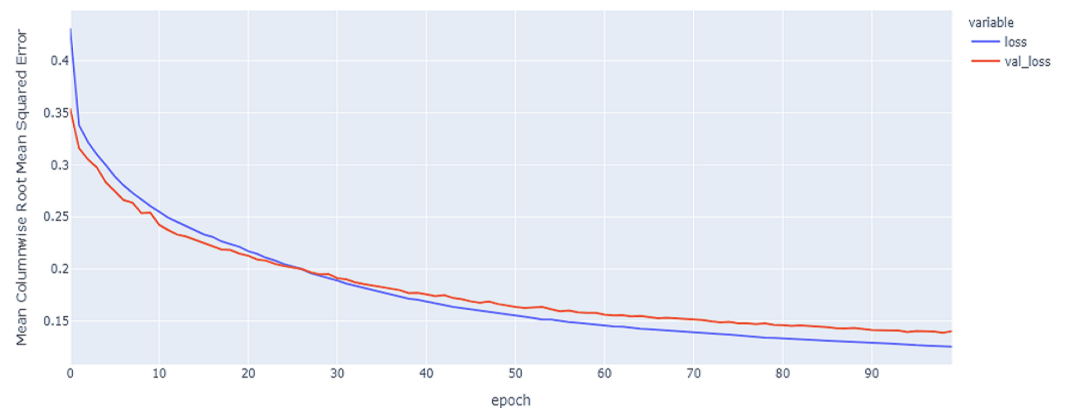


Figure 14 GRU model results on the categorical and numerical features with augmentation based on codon encoding. [Full-size](#) DOI: 10.7717/peerj-cs.597/fig-14

categorical features after augmentation. The best models for generalization with less overfitting are mentioned in Table 6 with base encoding and after applying augmentation which the differences in MCRMSE are minimized between training loss error and validation loss error. The difference in MCRMSE is 0.008 using the weighted average of three sequence models. Accordingly, codon encoding outperforms base encoding with respect to MCRMSE validation error using LSTM model; meanwhile, base encoding outperforms codon encoding due to minimized MCRMSE and less overfitting.

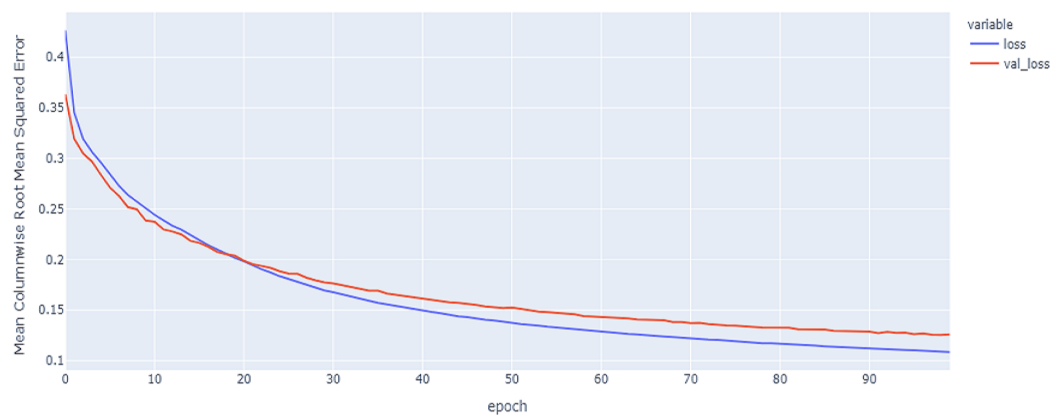


Figure 15 LSTM model results on the categorical and numerical features with augmentation based on codon encoding. [Full-size !\[\]\(1663bb69f307a960345edb0e712f8c02_img.jpg\) DOI: 10.7717/peerj-cs.597/fig-15](https://doi.org/10.7717/peerj-cs.597/fig-15)

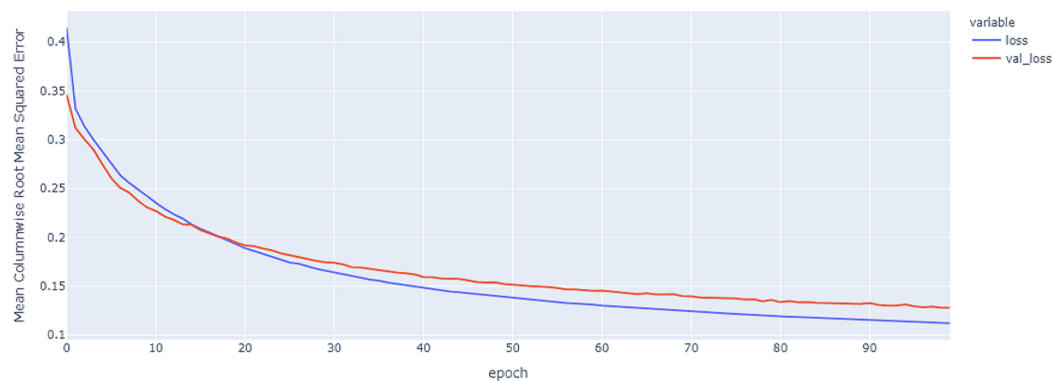


Figure 16 Hybrid model results on the categorical and numerical features with augmentation based on codon encoding. [Full-size !\[\]\(7c47b229ca7bdb95c18f544ee7ceb332_img.jpg\) DOI: 10.7717/peerj-cs.597/fig-16](https://doi.org/10.7717/peerj-cs.597/fig-16)

CONCLUSION AND FUTURE WORKS

mRNA vaccines are the fastest vaccine candidates for the treatment of COVID-19 but currently are facing several limitations such as degradation. The sequence modeling approach uses sequence data as mRNA vaccine sequences. This paper used sequence modeling, GRU, LSTM and Hybrid for predicting mRNA sequences responsible for the degradation of COVID-19 mRNA vaccine. The sequence models predict degradation by predicting five reactivity values for every position in the sequence. We applied two encoding methods called base encoding and codon encoding methods. Moreover, we used augmentation techniques to generate new data using the ARNIE package that resolved the issue of over-fitting in our implementation to increase the number of samples and make the model more generalized in the prediction of new data. In another method to resolve the over-fitting, we configured the dropout regularization method in the bidirectional layers of GRU, LSTM, and Hybrid sequence models.

The model extracted 300 features using encoding and embedding layer extraction for the categorical features and five features as the numerical features. Then, the features are concatenated to the model. The models give promising results since they predict the

five reactivity values with less validation MCRMSE of 0.125 using LSTM with codon encoding and after applying augmentation data. Meanwhile, the best models are presented with base encoding and augmentation with difference equals to 0.008 using weighted average of MCRMSE for three sequence models between training loss error and validation loss error; the results of training and validation loss errors are closer, and no over-fitting is obtained.

In the future, we plan to investigate more numerical features based on the characteristics of bioinformatics and RNA sequences to enhance the accuracy of the models and apply different classification models. Moreover, we also suggest applying a convolutional neural network (CNN) with sequence models to build a hybrid model with new features to improve the predictions. Graph convolutional networks (GCNs) have the ability to represent learning on a graph with the advantage of stacking deeper layers; therefore, this approach can be applied to evaluating mRNA degradation to get better results.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

This work was supported by the Institute of Research and Consulting Studies at King Khalid University—Saudi Arabia through grant number #(2-20-S-2020). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures

The following grant information was disclosed by the authors:

Institute of Research and Consulting Studies at King Khalid University—Saudi Arabia: #2-20-S-2020.

Competing Interests

The authors declare that they have no competing interests.

Author Contributions

- Talal S. Qaid conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the paper, and approved the final draft.
- Hussein Mazaar conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the paper, writing and formatting in Latex, and approved the final draft.
- Mohammed S. Alqahtani conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the paper, and approved the final draft.
- Abeer A. Raweh conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the paper, and approved the final draft.

- Wafaa Alakwaa conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the paper, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:

The scripts are available in the [Supplemental Files](#). The data are available at Kaggle: <https://www.kaggle.com/c/stanford-covid-vaccine/data>.

Supplemental Information

Supplemental information for this article can be found online at <http://dx.doi.org/10.7717/peerj-cs.597#supplemental-information>.

REFERENCES

- Abbasi J. 2020.** Covid-19 and mRNA vaccines—first large test for a new approach. *JAMA* **324**(12):1125–1127 DOI [10.1001/jama.2020.16866](https://doi.org/10.1001/jama.2020.16866).
- Ahuja AS, Reddy VP, Marques O. 2020.** Artificial intelligence and covid-19: a multidisciplinary approach. *Integrative Medicine Research* **9**(3):100434 DOI [10.1016/j.imr.2020.100434](https://doi.org/10.1016/j.imr.2020.100434).
- Chakraborty R, Parvez S. 2020.** COVID-19: an overview of the current pharmacological interventions, vaccines, and clinical trials. *Biochemical Pharmacology* **180**:114–184 DOI [10.1016/j.bcp.2020.114184](https://doi.org/10.1016/j.bcp.2020.114184).
- Cho K, Van Merriënboer B, Bahdanau D, Bengio Y. 2014.** On the properties of neural machine translation: encoder-decoder approaches. In: *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Doha, Qatar:103–111.
- Hochreiter S, Schmidhuber J. 1997.** Long short-term memory. *Neural Computation* **9**(8):1735–1780 DOI [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735).
- Hu H, Liang Y, Zhong X, Pan ZL, Huang L, Zhang HL, Xu Y, Zhou W, Liu Z. 2020.** Codon optimization with deep learning to enhance protein expression. *Scientific Reports* **10**(1):17617 DOI [10.1038/s41598-020-74091-z](https://doi.org/10.1038/s41598-020-74091-z).
- Jackson LA, Anderson EJ, Roupael NG, Roberts PC, Makhene M, Coler RN, McCullough MP, Chappell JD, Denison MR, Stevens LJ, Pruijssers AJ, McDermott A, Flach B, Doria-Rose NA, Corbett KS, Morabito KM, O’Dell S, Schmidt SD, Swanson PA II, Padilla M, Mascola JR, Neuzil KM, Bennett H, Sun W, Peters E, Makowski M, Albert J, Cross K, Buchanan W, Pikaart-Tautges R, Ledgerwood JE, Graham BS, Beigel JH. 2020.** An mRNA vaccine against sars-cov-2—preliminary report. *New England Journal of Medicine* **383**(20):1920–1931 DOI [10.1056/NEJMoa2022483](https://doi.org/10.1056/NEJMoa2022483).
- Keshavarzi Arshadi A, Webb J, Salem M, Cruz E, Calad-Thomson S, Ghadirian N, Collins J, Diez-Cecilia E, Kelly B, Goodarzi H, Yuan JS. 2020.** Artificial intelligence for covid-19 drug discovery and vaccine development. *Frontiers in Artificial Intelligence* **3**:65 DOI [10.3389/frai.2020.00065](https://doi.org/10.3389/frai.2020.00065).
- Khuroo MS, Khuroo M, Khuroo MS, Sofi AA, Khuroo NS. 2020.** COVID-19 vaccines: a race against time in the middle of death and devastation!. *Journal of Clinical and Experimental Hepatology* **10**(6):610–621 DOI [10.1016/j.jceh.2020.06.003](https://doi.org/10.1016/j.jceh.2020.06.003).
- Lu J, Lu G, Tan S, Xia J, Xiong H, Yu X, Qi Q, Yu X, Li L, Yu H, Xia N, Zhang T, Xu Y, Lin J. 2020.** A covid-19 mRNA vaccine encoding sars-cov-2 virus-like particles induces a strong antiviral-like immune response in mice. *New England Journal of Medicine* **30**(10):936–939.

- Marian AJ. 2021.** Current state of vaccine development and targeted therapies for covid-19: impact of basic science discoveries. *Cardiovascular Pathology* **50(13)**:107–278
DOI 10.1016/j.carpath.2020.107278.
- Ong E, Wong MU, Huffman A, He Y. 2020.** COVID-19 coronavirus vaccine design using reverse vaccinology and machine learning. *Frontiers in Immunology* **11**:1581
DOI 10.3389/fimmu.2020.01581.
- Pfizer. 2020.** Pfizer and biontech choose lead mrna vaccine candidate against covid-19 and commence pivotal phase 2/3 global study. Available at <https://www.pfizer.com/news/press-release/press-release-detail/pfizer-and-biontech-choose-lead-mrna-vaccine-candidate-0> (accessed 15 September 2020).
- Sempowski GD, Saunders KO, Acharya P, Wiehe KJ, Haynes BF. 2020.** Pandemic preparedness: developing vaccines and therapeutic antibodies for covid-19. *Cell* **181(7)**:1458–1463
DOI 10.1016/j.cell.2020.05.041.
- Stanford University. 2020.** OpenVaccine: COVID-19 mRNA vaccine degradation prediction. Available at <https://www.kaggle.com/c/stanford-covid-vaccine/data> (accessed 8 October 2020).
- Synced. 2020.** New baidu algorithms boost mRNA vaccine development for sars-cov-2. Available at <https://syncedreview.com/2020/05/12/new-baidu-algorithms-boost-mrna-vaccine-development-for-sars-cov-2/> (accessed 8 October 2020).
- tito:kaggle. 2020.** How to generate augmentation data. Available at <https://www.kaggle.com/its7171/how-to-generate-augmentation-data> (accessed 15 September 2020).
- Wang J, Peng Y, Xu H, Cui Z, Williams RO. 2020.** The covid-19 vaccine race: challenges and opportunities in vaccine formulation. *AAPS PharmSciTech* **21(6)**:225.
- World Health Organization (WHO). 2020.** Covid-19 vaccines. Available at https://www.who.int/emergencies/diseases/novel-coronavirus-2019/COVID-19-vaccines?gclid=Cj0KCQjwzvbv7BRDIARIsAM-A6_04NMCMhRJdcRFsKW65qU5gZnjrrGFWRm48-GtN8ZtBLcruSTDcQB0aAogREALw_wcB (accessed 8 October 2020).
- Zhang N-N, Li X-F, Deng Y-Q, Zhao H, Huang Y-J, Yang G, Huang W-J, Gao P, Zhou C, Zhang R-R, Guo Y, Sun S-H, Fan H, Zu S-L, Chen Q, He Q, Cao T-S, Huang X-Y, Qiu H-Y, Nie J-H, Jiang Y, Yan H-Y, Ye Q, Zhong X, Xue X-L, Zha Z-Y, Zhou D, Yang X, Wang Y-C, Ying B, Qin C-F. 2020.** A thermostable mrna vaccine against covid-19. *Cell* **182(5)**:1271–1283 DOI 10.1016/j.cell.2020.07.024.
- Zhang K, Pan X, Yang Y, Shen H-B. 2019.** CRIP: predicting circRNA–RBP-binding sites using a codon-based encoding and hybrid deep neural networks. *RNA* **25(12)**:1604–1615
DOI 10.1261/rna.070565.119.