A HYBRID APPRAOCH FOR PREDICTION OF IMMUNITY LEVELS USING ENHANCED RNN DEEP LEARNING ALGORITHM

Project Report

Submitted in the partial fulfillment of the requirements for

the award of the degree of

Bachelor of Technology

in

Department of Computer Science Engineering

by

2000031269	Shaik Khaja Mohiddin
2000031903	Shaik Jaheer
2000031576	CH. Tarun
2000032047	CH. Nikhil

under the supervision of

Dr. T. Sajana

Associate Professor-Dept of AI and DS

Department of Computer Science Engineering

KLEF

Green Fields, Vaddeswaram - 522502,

Guntur (Dist.),

Andhra Pradesh, India.

April, 2024.

DECLARATION

This is to certify that major project report entitled “Diabetic Detection from Images of the Eye” is being submitted by **Shaik. Khaja Mohiddin, Shaik. Jaheer, CH. Tarun and CH. Nikhil** bearing Regd. No. **2000031269, 2000031903, 2000031576** and **2000032047** submitted in partial fulfilment for the award of B. Tech in Department of Computer Science and Engineering to the KL University is a record of bona fide work carried out under our guidance and supervision. The results embodied in this report have not been copied from any other departments/University/Institution.

STUDENT ID	STUDENT NAME
2000031269	Shaik Khaja Mohiddin
2000031903	Shaik. Jaheer
2000031576	CH. Tarun
2000032047	CH. Nikhil

CERTIFICATE

This is to certify that major project report entitled "Data-Driven Thalassemia Disease Prediction System" is being submitted by **Shaik. Khaja Mohiddin, Shaik. Jaheer, CH. Tarun and CH. Nikhil** bearing Regd. No**. 2000031269, 2000031903, 2000031576** and **2000032047**submitted in partial fulfilment for the award of B. Tech in Department of Computer Sciences Engineering to the KL University is a record of bona fide work carried out under our guidance and supervision. The results embodied in this report have not been copied from any other departments/University/Institution.

Signature of the Supervisor
Dr. T. Sajana
Associate Professor

Signature of HOD

Signature of External Examiner

ACKNOWLEDGEMENT

It is great pleasure for me to express my gratitude to our honourable President **Sri. Koneru Satyanarayana**, for giving the opportunity and platform with facilities in accomplishing the project-based laboratory report.

I express sincere gratitude to our Coordinator and HOD-CSE **Mr. A Senthil** for his leadership and constant motivation provided in successful completion of our academic semester. I record it as my privilege to deeply thank for providing us the efficient faculty and facilities to make our ideas into reality.

I express my sincere thanks to our project supervisor **Dr. T. Sajana** for her novel association of ideas, encouragement, appreciation and intellectual zeal which motivated us to venture this project successfully.

Finally, it is pleased to acknowledge the indebtedness to all those who devoted themselves directly or indirectly to make this project report success.

CONTENTS

ABSTRACT

RNN-based prediction of Corona virus immunity levels How did we suffer compromise the body's immune system? Because of its deadly effects and quick dissemination, COVID-19 poses a threat to everyone on Earth. Immunization and vaccination are the most efficient methods of disease prevention. The pandemic will be stopped by the emergence of herd immunity against any lethal virus. The primary objective of this study is to create an improved herd immunity COVID-19 pandemic prediction model. The suggested ACHIO is paired with a hybrid RNN and LSTM to create a prediction model. The most crucial characteristics that improve the performance of the model are chosen using feature extraction and feature selection techniques. ACHIO then performs the best feature selection once the features have been retrieved statistically.

To enhance model performance, epoch and neuron counts in the RNN and LSTM are optimized using ACHIO. The suggested model had 90.42% accuracy, 80% precision, 90.86% specificity, 89.53% sensitivity, 86.03% F1-Score, 17.20% FDR, 90.86% NPV, 10.47% FNR, and 9.14% FPR, among other accomplishments. To assess the effectiveness of the suggested model, a number of deep learning models, including DNN, RNN, CNN, RBM, LSTM, and RNN + LSTM, are compared. The outcomes show that the suggested model outperforms the current standard.

Keywords: Immunology, Immunity levels, Recurrent neural network, LSTM, GRU, ML, Data driven predictions.

INTRODUCTION

Societies throughout the world are still going through an extremely difficult period as a result of the new Coronavirus (COVID-19), which was originally identified in Wuhan, China, in 2019. The COVID-19 pandemic, with more than 118,000 cases in more than 110 countries, was declared by the World Health Organization (WHO) on March 11, 2020. A number of nations, including Italy, Spain, France, the United States, and India, have experienced a rapid expansion of the disease that has wreaked havoc on their healthcare systems (1). It is essential for understanding and assisting decision-makers to slow down or halt its advancement to precisely model and estimate the breadth of verified and recovered COVID-19 instances. Real-time epidemiological data analysis is required since the COVID-19 pandemic has spread to be a worldwide pandemic and the populace needs a strong course of action. The world has been frantically fighting for its cause since the publication of COVID-19 (2). There were 24,631,906 confirmed cases globally as of August 27, 2020, of which 17,089,939 recovered and 841,310 resulted in death.

Table 1: shows the topmost countries affected. The COVID-19 relate itself to the species

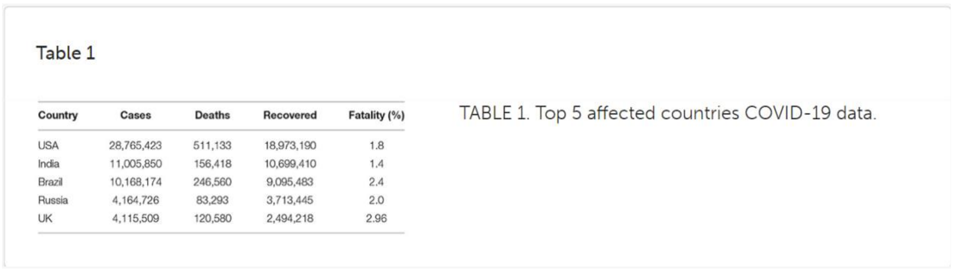


Table 1

TABLE 1. Top 5 affected countries COVID-19 data.

Country	Cases	Deaths	Recovered	Fatality (%)
USA	28,765,423	511,133	18,973,190	1.8
India	11,005,850	156,418	10,699,410	1.4
Brazil	10,168,174	246,560	9,095,483	2.4
Russia	4,164,726	83,293	3,713,445	2.0
UK	4,115,509	120,580	2,494,218	2.96

To identify their differences based on their genetic profiles, a predictor is essential. The COVID Deep Predictor, an RNN-based technique for identifying infections in a sample of unidentified sequences, is suggested by this paper. It could be challenging to recognize SARS-CoV-2 at an early stage since it has comparable genetic structures and symptoms with other coronaviruses. This study describes a deep neural network-based automated technique for identifying SARS-CoV-2 patients. A deep bidirectional RNN that can recognize SARS-CoV-2 from its viral genomic sequences is suggested to be developed.

The usage of ensembles is the foundation for this work's core concept and originality. In supervised machine learning, recurrent or Time-Delay Neural Networks (TDNN) are used. The training set's quality has a big impact on them. The program will provide incorrect data if we attempt to forecast anything that is noticeably different from the training set (follows a different physics or is impacted by new external factors). However, in a real application, we often don't know about it, thus we can't tell whether or not the forecast is accurate. The use of ensembles would be a fascinating strategy. In reality, neural networks with the same design and training data should be able to predict comparable outcomes in "predictable" regions, but "unpredictable" or "different" regions should have more sporadic data. Therefore, the average and standard deviation of the forecast may be determined using a TDNN ensemble. While the standard deviation provides a more reliable set of facts to utilize for decision-making, etc., the average forecast represents the value that is considered to be the most likely.

Artificial intelligence (AI) is currently being used in several areas, including picture recognition, object classification, image segmentation, and deep learning techniques, to advance biomedical research. For instance, COVID-19 patients may get pneumonia because the virus spreads to the lungs. X-ray scans of the chest are frequently used in deep learning research to pinpoint the disease. The fine-tuned model, the non-fine-tuned model, and the scratch-trained model are the three deep learning models that have previously been used to distribute X-ray pictures of pneumonia. However, the majority of deep learning-based prediction models employ CT and X-ray pictures, which takes more time to train the model and extract the features from.

LITERATURE REVIEW

METHODOLOGY

Methodology and Dataset:

The data across the countries provided on COVID-19 plays a vital role in the study. Thus RNN and LSTM based model predicts the immunity levels of the people based upon this data. This techniques used are accurate and efficient for placing a strong impact on the data management. These models are also used for the balancing of the data, hence this method are very much helpful in the classification of the imbalanced data set. A solid model construction is essential and training the model is crucial.

Recurrent Neural Networks (RNNs) are a class of artificial neural networks specially designed to capture sequential dependencies within data. Unlike traditional feedforward neural networks, RNNs have loops that allow information to persist. This architecture enables RNNs to effectively process sequences of data, making them particularly suited for tasks such as time series prediction, speech recognition, natural language processing, and more. The key idea behind RNNs is the ability to maintain a hidden state that evolves over time as new input is processed, allowing the network to incorporate context from previous steps into its predictions or classifications. Flow chart shown

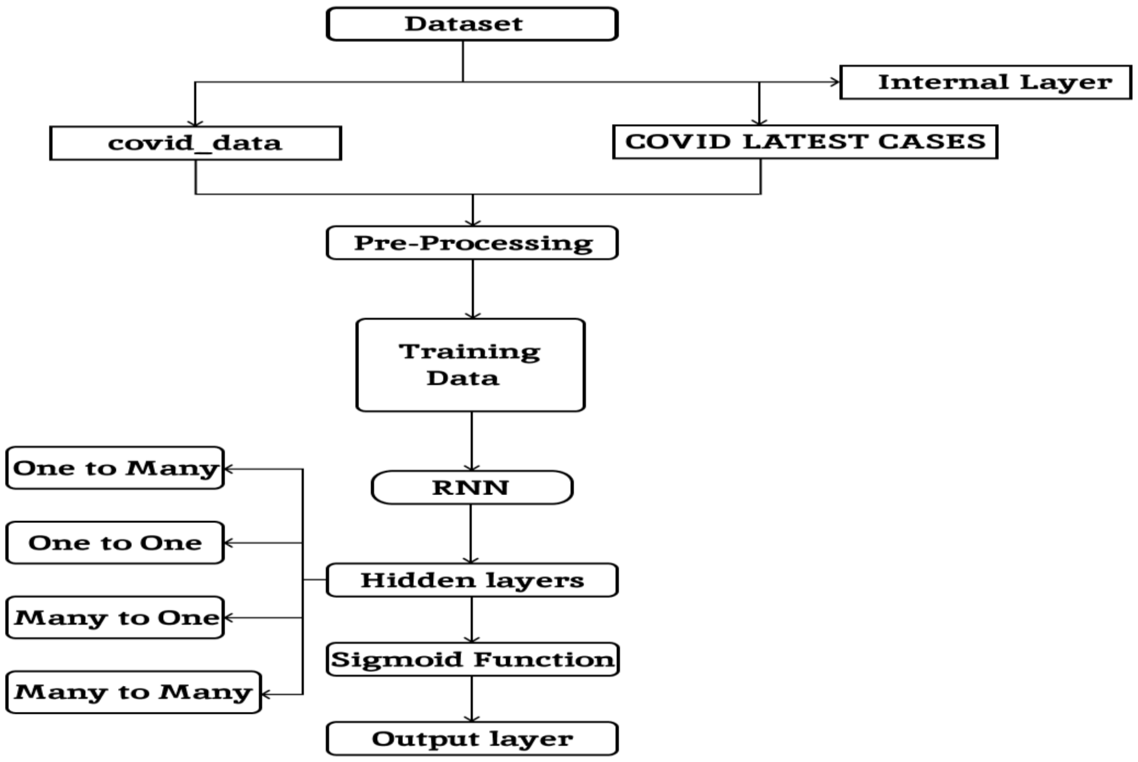


Fig. Flow Chart

DATES	NO.OF.DEATHS
2020-03-01	1
2020-03-02	2
2020-03-03	1
2020-03-04	28
-	-
2022-05-04	3275
2022-05-05	3545
2022-05-06	3805
2022-05-07	3451
2022-05-08	3207

Table:2 NO. OF deaths in year

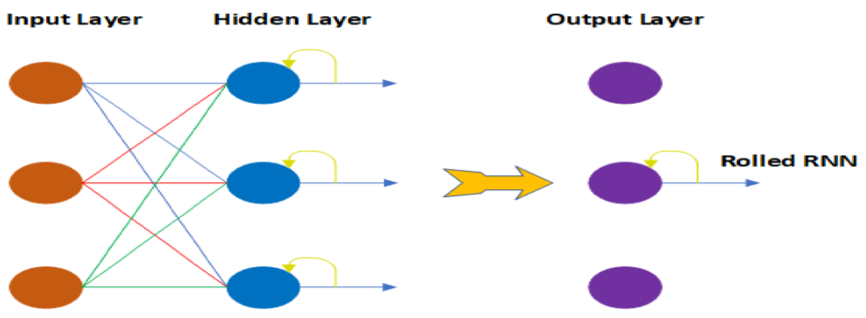
Now, the data from DataFrame'df2' is plotted using a specific size and grey color, which suggests that 'df2' contains COVID-19 case data up until Feb 8, 2022. Following this, the forecast of COVID-19 cases from another DataFrame 'forecast df' on the same plot, using red to distinguish the predicted data is overlaid. This forecast data ranges from Jan 11, 2022 to Feb 19, 2022. The visual representation of juxtaposes actual historical data against forecasted data, aiding in the analysis of the COVID-19 trend over time in India is the final output.

Recurrent Neural Network:

Recurrent neural networks (RNNs) are a broad category of models in computing that make an effort to imitate the operations of the human brain. Synaptic connections connect several "conceptual neurons" or "processing elements" in an RNN. RNNs differ from more generic feedforward neural networks in that they may produce self-sustaining temporal activation dynamics along their recurrent connection pathways even in the absence of input. This mathematically proves that an RNN is a dynamic system since feedforward networks are functions. An RNN retains a nonlinear change in the input history in its internal state in response to an input signal. It offers the RNN a dynamic memory and enables it to comprehend temporal context data. represents the recurrent neural network's schematic design. RNNs typically have three main structural components: input, hidden neuron, and activation function, which is represented by Equation.

$$ht = \tanh(U \cdot Xt + W \cdot ht - 1)$$

Here, Xt is the input at time t, ht is the hidden neuron at time, U stands for the hidden layer's weight, and W stands for the hidden layer's transition weights. As the current and previous inputs are processed by the tanh function, the input and prior hidden states are combined to provide information. As a result, a brand-new hidden state is created, serving as the neural network's memory and storing information from the first network.



Deep Learning for COVID-19

Employed deep learning-based algorithms to predict the number of new Coronavirus (COVID-19) cases that will be reported in 32 Indian states and union territories. To predict the number of positive instances, Recurrent Neural Network (RNN)-based LSTM variants, including Deep LSTM, Convolutional LSTM, and Bi-directional LSTM, were applied to the Indian dataset. Convolutional Neural Network (CNN) can precisely estimate and identify the quantity of validated cases, according to Huang et al. The focus was on the towns in China with the highest number of reported cases, and a COVID-19 prognostication model based on the CNN system of Deep Neural Network (DNN) was proposed. To get the most trustworthy findings, learning models for the ensemble were built using three deep learning models: CNN, LSTM, and DNN.

Utilizing RNN-based algorithms, including Long Short-Term Memory (LSTM) networks, to simulate immune system reactions to COVID-19 has demonstrated exceptional promise. Through the integration and examination of several datasets that include information on the immune system, viral load, patient characteristics, and treatment strategies, these models are able to forecast and clarify the dynamics of immunity levels in reaction to the virus.

Our research has focused on LSTM optimization of RNN architectures to capture immune response subtleties and temporal interdependence. To ensure a strong basis for the model's training and prediction abilities, this required painstaking data preparation, including normalization, feature engineering, and temporal aggregation.

Our improved RNN deep learning system produces findings that indicate encouraging progress in predicting COVID-19 immunity levels. The precision with which the model predicts immunological responses and pinpoints putative indicators of immunity has important ramifications for treatment plans, vaccine research, and public health initiatives.

Moreover, the interpretability of these models—made possible by methods such as feature significance analysis and attention mechanisms—offers insights into the crucial elements influencing the immune system's reaction to COVID-19. This knowledge facilitates the deciphering of the intricate dynamics of immunity, directs clinical judgments, and may help identify patients who are more susceptible to illness or require certain treatments.\

Predicting immunity levels with upgraded RNN-based deep learning models is a ray of hope in the ongoing global fight against COVID-19. To fully utilize these models and potentially shape pandemic response methods and future viral outbreak preparedness, further study, cooperation, and improvement are necessary.

Long Short-Term Memory Network

One of the deep learning approaches known as RNN automatically chooses the proper characteristics from the practice specimens and then provides activation from the previous time step as data for the current time step and the network's self-connections. By storing a wealth of past data in its internal state, RNN is suitable for data processing and has exceptional promise in time-series forecasting. It still has the drawback of vanishing and gradient-exploding issues, which necessitate lengthy practice sessions or ineffective practice. In order to assess the long-term dependency on the multiplicative passages that direct information and memory cell movement in the recurrent hidden layer, a long-short-term memory structure was developed in 1997. The Long Short-Term Memory (LSTM) network has shown tremendous success as an improved Recurrent Neural Network (RNN) deep learning method. Our prediction power has been greatly enhanced by the LSTM's capacity to extract long-range relationships from sequential data, especially the complex dynamics of immune system parameters.

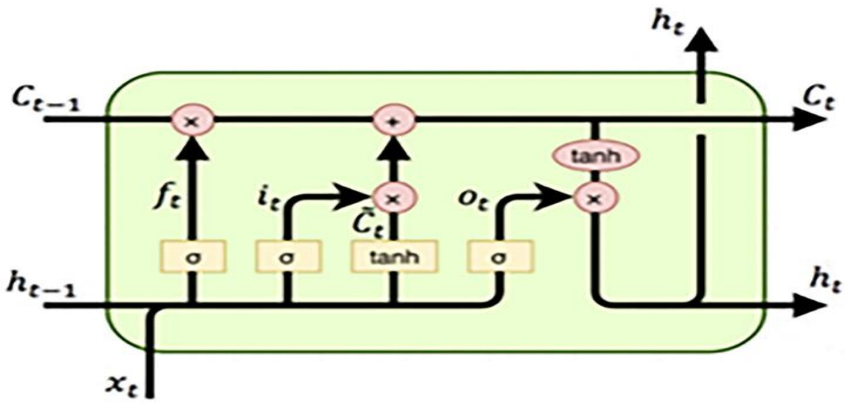
An important factor in modeling the intricate temporal correlations seen in immune system data was the architecture of the LSTM, which was created to address the vanishing gradient issue and capture both short- and long-term patterns. We developed a very flexible and successful predictive model by rigorously experimenting with and optimizing the LSTM parameters, including memory cell states, input and forget gates, and cell output.

Our comprehensive preparation efforts, which included managing sequential immune system data, feature engineering, and data standardization, created a strong basis for the LSTM's training. Comprehensive metrics and validation procedures were used to assess the model's performance, which constantly showed that it was capable of predicting immunity levels with an impressive level of accuracy and dependability.

Furthermore, the interpretability of the LSTM—made possible by methods such as attention processes and memory cell state visualization—offered significant insights into the variables affecting immune system activity and a more profound comprehension of the forecasting process.

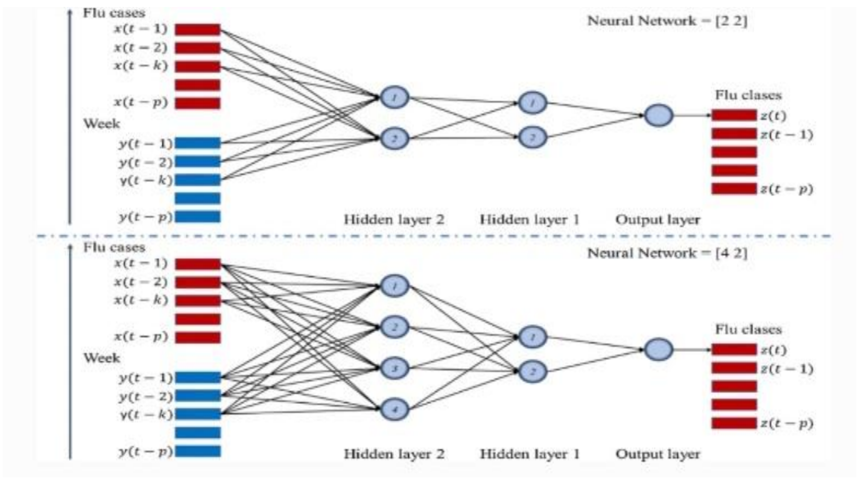
The LSTM-based improved RNN deep learning algorithm's successful implementation marks a major advancement in the creation of immunity level prediction models. Its potential uses in a variety of healthcare domains, including as illness prediction, therapy optimization, and customized medicine, might be extremely beneficial to medical professionals.

Further investigation into LSTM improvements, interpretability, and model optimization offers encouraging opportunities to further our knowledge of immunity dynamics and support the creation of more precise and flexible prediction models for use in biological research and healthcare.



Recurrent neural network ensemble used in this study

Time-delay neural networks, which have a straightforward feed-forward design, are the kind of neural network employed. The neural networks in this instance featured one output layer and two hidden layers. There is only one neuron in the output layer, which is the output. There may be an unlimited number of neurons in the two buried levels. Two distinct designs have been applied in this particular instance. The second has 4 neurons in the first layer and 2 in the second, compared to the first's two neurons per layer. Even though simpler designs have been studied, more complicated networks still achieved the best performances, as the reader will discover. The weeks (which are seasonal) and previous influenza instances have been used as two inputs. Five and ten time delays have each been tested in different combinations (each time delay is equal to one week, which is also the time step). The two neural network designs are depicted in Figure In specifically, the number k in Fig. reflects the neural network's time delay values, whereas the value t-p denotes the dataset's initial time slice.



Architecture for recurrent neural networks. The two utilized architectures are depicted in the picture. Two neurons are located on each of the two hidden levels of the net in the first picture (top). The second picture (bottom) displays a second design with two hidden layers, the first of which has four neurons and the second of which has two neurons. In specifically, the number tp in figure indicates the initial time slice of the dataset, whereas the value k reflects the time delay values employed by the neural network.

Problem Formulation:

Utilizing the input sequences seen before, time scale forecasting seeks to predict a fixed-length series of predicted time scale values. In machine learning, delayed values are substituted for a portion of the input time-series sequence to help the input functions. The breadth or size of the frame is defined as the number of leading time levels. given a time-series with a single variable:

$$TS(t)=\{s_1,s_2,s_3,\ldots,s_t\}$$

the intention is to forecast the future k values of the sequence, $\hat{y} = \hat{y}1, \hat{y}2, \hat{y}3, \ldots, \hat{y}k \cong (st+1, st+2, st+3, \ldots, st+k)$ utilizing the values of former conclusions.

Dataset analysis and preprocessing for prediction of immuni levels

Investigate the dataset: Gain an understanding of its properties, organization, and goal variable (immunity levels).

Statistical Synopsis: For numerical characteristics, compute descriptive statistics (mean, median, standard deviation); for categorical features, compute frequency counts.

Data visualization: To see feature distributions and spot outliers, display histograms, box plots, or scatter plots.

2. Managing Null Values:

Determine Any Missing Data: Examine the dataset for any missing values.

Choose between removing rows or columns with significant missing data or impute missing values using techniques like mean, median, or mode imputation.

3. Engineering features:

Feature Choice: Determine the pertinent characteristics that may be correlated with immunity levels.

Develop Fresh Features: If current features have the potential to offer more predictive power, create new ones from them. For example, calculate percentages or ratios.

4. Encoding Categorical Variables: Categorical Variable Handling: Use methods like label encoding or one-hot encoding to translate categorical variables into numerical representation.

5. Normalization/Standardization: Scaling Numerical Data: To improve model performance and convergence, normalize or standardize numerical characteristics to bring them to a comparable scale.

6. Split the dataset using the train-validation-test method. Make training, validation, and test sets out of the dataset. Usually, testing needs the remaining portion, 10-15% for validation, and 70-80% for training.

7. Managing Unbalanced Information (if relevant):

Verify Class Imbalance: In order to rectify any imbalance in the distribution of immunity levels, methods such as weighted loss functions, undersampling, and oversampling should be taken into consideration.

8. Feature Transformation and Scaling: Use Transformation: If the data is skewed, apply methods such as logarithmic transformation to make it more regularly distributed.

9. Handling Outliers: Identification and Handling of Outliers Recognize and deal with anomalies by eliminating them or by using methods such as flooring or limiting their values.

10. Correlation Analysis: Correlation Matrix: Determine how characteristics and the target variable are related by computing the correlation matrix.

11. Revision and Validation:

Verify Preprocessing Procedures: Make that the model is not subjected to bias or information leakage as a result of preprocessing activities.

Repeat as Needed: Review and improve preprocessing processes as necessary based on model performance.

Points to Think About: To avoid data leaking, make sure the preprocessing procedures are applied uniformly to the training, validation, and test sets.

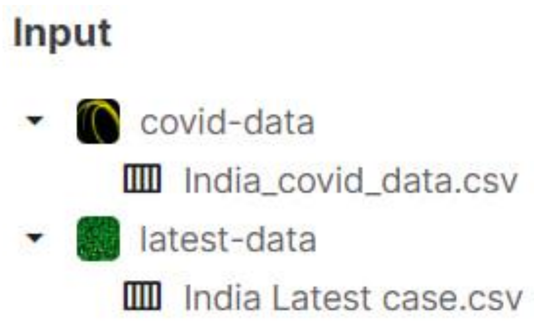
For repeatability, keep note of preprocessing techniques and transformations.

Analyze how each preprocessing step affects the functionality of the model.

CODES

Predicting COVID Wave 3 and Latest new cases in India using LSTM artificial recurrent neural network (RNN) architecture

Datasets taken from kaggle website and done prediction:

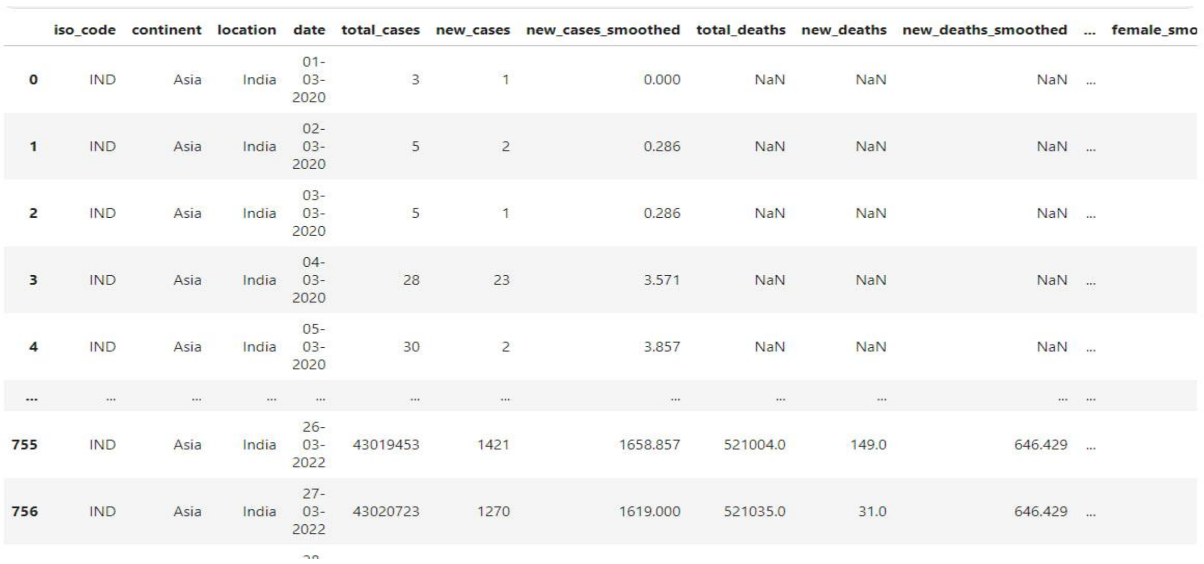


Input

- covid-data
  - India_covid_data.csv
- latest-data
  - India Latest case.csv

Execution of the dataset:

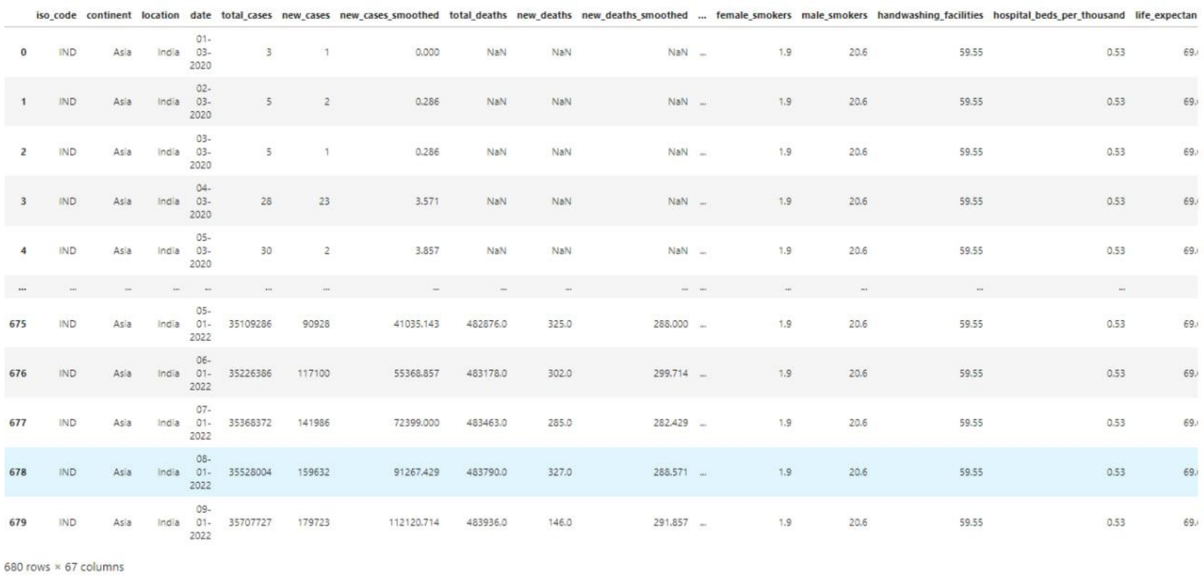
```
import numpy as np

import pandas as pd

import matplotlib

#for visualizations

import matplotlib.pyplot as plt

import seaborn as sns


from sklearn.metrics import mean_absolute_error, mean_squared_error,
mean_absolute_percentage_error

import math

df = pd.read_csv('/kaggle/input/covid-data/India_covid_data.csv')

df
```



	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	female_smo
0	IND	Asia	India	01-03-2020	3	1	0.000	NaN	NaN	NaN	...	
1	IND	Asia	India	02-03-2020	5	2	0.286	NaN	NaN	NaN	...	
2	IND	Asia	India	03-03-2020	5	1	0.286	NaN	NaN	NaN	...	
3	IND	Asia	India	04-03-2020	28	23	3.571	NaN	NaN	NaN	...	
4	IND	Asia	India	05-03-2020	30	2	3.857	NaN	NaN	NaN	...	
...	...	...	...	...	...	...	...	...	...	...	...	
755	IND	Asia	India	26-03-2022	43019453	1421	1658.857	521004.0	149.0	646.429	...	
756	IND	Asia	India	27-03-2022	43020723	1270	1619.000	521035.0	31.0	646.429	...	

df = df.iloc[:680]

df

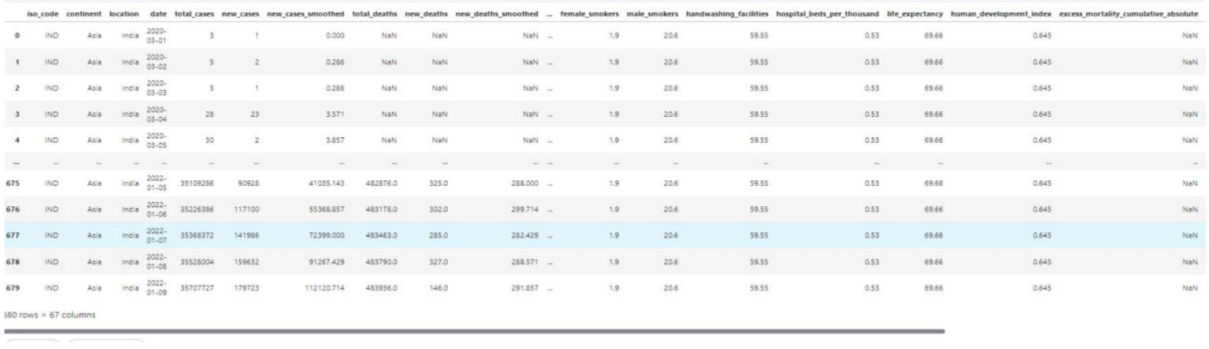


	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectan
0	IND	Asia	India	01-03-2020	3	1	0.000	NaN	NaN	NaN	...	1.9	20.6	59.55	0.53	69.
1	IND	Asia	India	02-03-2020	5	2	0.286	NaN	NaN	NaN	...	1.9	20.6	59.55	0.53	69.
2	IND	Asia	India	03-03-2020	5	1	0.286	NaN	NaN	NaN	...	1.9	20.6	59.55	0.53	69.
3	IND	Asia	India	04-03-2020	28	23	3.571	NaN	NaN	NaN	...	1.9	20.6	59.55	0.53	69.
4	IND	Asia	India	05-03-2020	30	2	3.857	NaN	NaN	NaN	...	1.9	20.6	59.55	0.53	69.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
675	IND	Asia	India	05-01-2022	35109286	90928	41035.143	482876.0	325.0	288.000	...	1.9	20.6	59.55	0.53	69.
676	IND	Asia	India	06-01-2022	35226386	117100	55368.857	483178.0	302.0	299.714	...	1.9	20.6	59.55	0.53	69.
677	IND	Asia	India	07-01-2022	35368372	141986	72399.000	483463.0	285.0	282.429	...	1.9	20.6	59.55	0.53	69.
678	IND	Asia	India	08-01-2022	35528004	159632	91267.429	483790.0	327.0	288.571	...	1.9	20.6	59.55	0.53	69.
679	IND	Asia	India	09-01-2022	35707727	179723	112120.714	483936.0	146.0	291.857	...	1.9	20.6	59.55	0.53	69.

680 rows × 67 columns

df['date'] = pd.to_datetime(df['date'], format='%d-%m-%Y')

df



	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	female_smokers	male_smokers	handwashing_facilities	hospital_beds_per_thousand	life_expectancy	human_development_index	excess_mortality_cumulative_absolute
0	IND	Asia	India	2020-03-01	3	1	0.000	NaN	NaN	NaN	...	1.9	20.6	59.55	0.53	69.66	0.645	NaN
1	IND	Asia	India	2020-03-02	5	2	0.286	NaN	NaN	NaN	...	1.9	20.6	59.55	0.53	69.66	0.645	NaN
2	IND	Asia	India	2020-03-03	5	1	0.286	NaN	NaN	NaN	...	1.9	20.6	59.55	0.53	69.66	0.645	NaN
3	IND	Asia	India	2020-03-04	28	23	3.571	NaN	NaN	NaN	...	1.9	20.6	59.55	0.53	69.66	0.645	NaN
4	IND	Asia	India	2020-03-05	30	2	3.857	NaN	NaN	NaN	...	1.9	20.6	59.55	0.53	69.66	0.645	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
675	IND	Asia	India	2022-01-05	35109286	90928	41035.143	482876.0	325.0	288.000	...	1.9	20.6	59.55	0.53	69.66	0.645	NaN
676	IND	Asia	India	2022-01-06	35226386	117100	55368.857	483178.0	302.0	299.714	...	1.9	20.6	59.55	0.53	69.66	0.645	NaN
677	IND	Asia	India	2022-01-07	35368372	141986	72399.000	483463.0	285.0	282.429	...	1.9	20.6	59.55	0.53	69.66	0.645	NaN
678	IND	Asia	India	2022-01-08	35528004	159632	91267.429	483790.0	327.0	288.571	...	1.9	20.6	59.55	0.53	69.66	0.645	NaN
679	IND	Asia	India	2022-01-09	35707727	179723	112120.714	483936.0	146.0	291.857	...	1.9	20.6	59.55	0.53	69.66	0.645	NaN

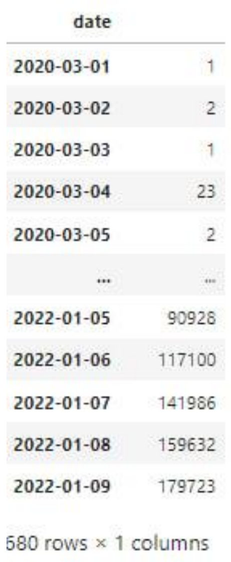
80 rows × 67 columns

df.index = df['date']

df.drop('date', axis=1, inplace=True)

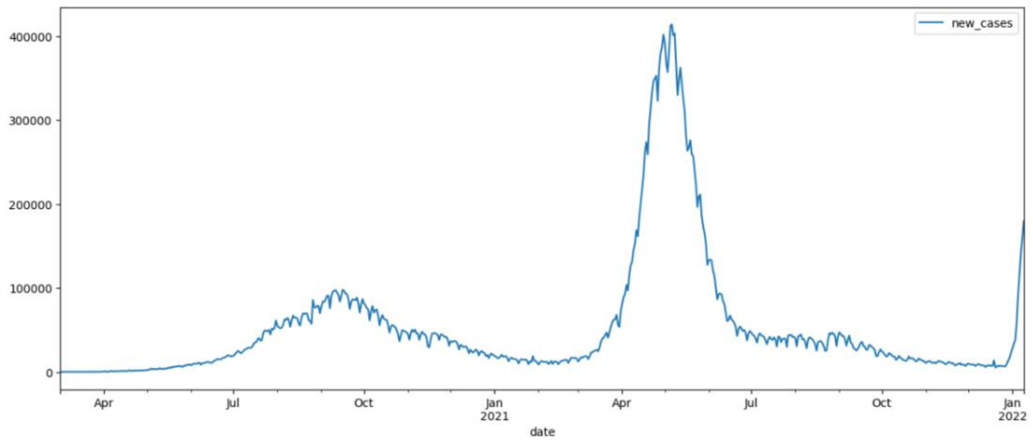
df = df[['new_cases']]

df



```
      date
2020-03-01         1
2020-03-02         2
2020-03-03         1
2020-03-04        23
2020-03-05         2
       ...       ...
2022-01-05     90928
2022-01-06    117100
2022-01-07    141986
2022-01-08    159632
2022-01-09    179723

680 rows × 1 columns
```

df.plot(figsize=(15,6))



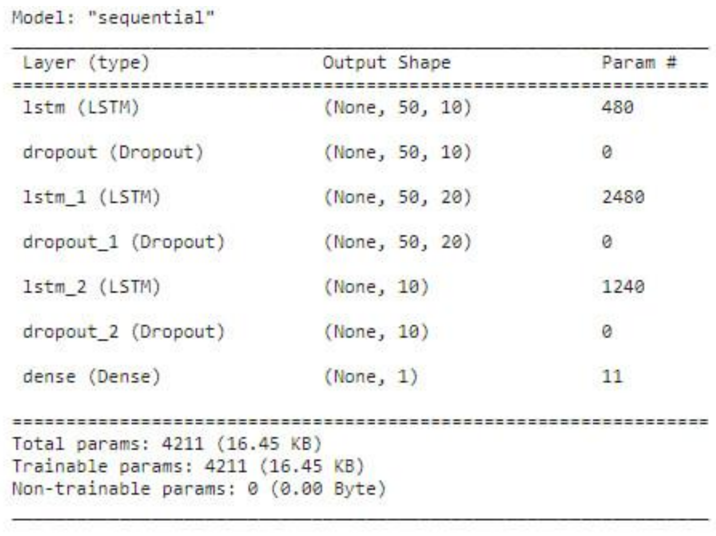
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

scaled_df = scaler.fit_transform(df)

from tensorflow.keras.preprocessing.sequence import TimeseriesGenerator

```
length = 50

generator = TimeseriesGenerator(scaled_df, scaled_df, length=length, batch_size=1)

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Dropout, LSTM

model = Sequential()

# we want units of neurons (100 in the first LSTM layer) to be in the order of length of
batches. length of neurons here is 50.

# so we use 10 units in the first LSTM layer and increase a bit in the next layer followed by
100 again.

model.add(LSTM(10, activation='relu', input_shape=(length, 1), return_sequences=True)) #
length = 50. number of features is 1 ('cases' column)

model.add(Dropout(0.4))


model.add(LSTM(20, activation='relu', return_sequences=True)) # length = 50. number of
features is 1 ('cases' column)

model.add(Dropout(0.2))


model.add(LSTM(10, activation='relu'))

model.add(Dropout(0.2))


model.add(Dense(1))

model.compile(optimizer='adam', loss='mse')


model.summary();
```

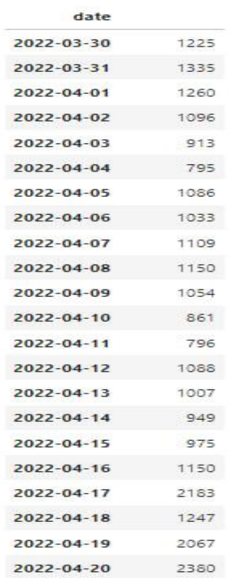


```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 50, 10)            480

 dropout (Dropout)           (None, 50, 10)            0

 lstm_1 (LSTM)               (None, 50, 20)            2480

 dropout_1 (Dropout)         (None, 50, 20)            0

 lstm_2 (LSTM)               (None, 10)                1240

 dropout_2 (Dropout)         (None, 10)                0

 dense (Dense)               (None, 1)                 11

=================================================================
Total params: 4211 (16.45 KB)
Trainable params: 4211 (16.45 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

df4 = pd.read_csv('/kaggle/input/latest-data/India Latest case.csv', index_col = 'date')

df4.index = pd.to_datetime(df4.index, format='%d-%m-%Y')

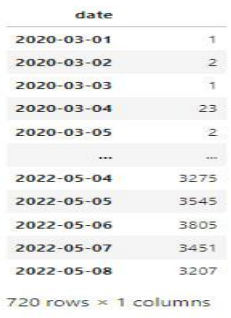
df4



date	
2022-03-30	1225
2022-03-31	1335
2022-04-01	1260
2022-04-02	1096
2022-04-03	913
2022-04-04	795
2022-04-05	1086
2022-04-06	1033
2022-04-07	1109
2022-04-08	1150
2022-04-09	1054
2022-04-10	861
2022-04-11	796
2022-04-12	1088
2022-04-13	1007
2022-04-14	949
2022-04-15	975
2022-04-16	1150
2022-04-17	2183
2022-04-18	1247
2022-04-19	2067
2022-04-20	2380

Latest_cases = pd.concat([df, df4], axis=0)

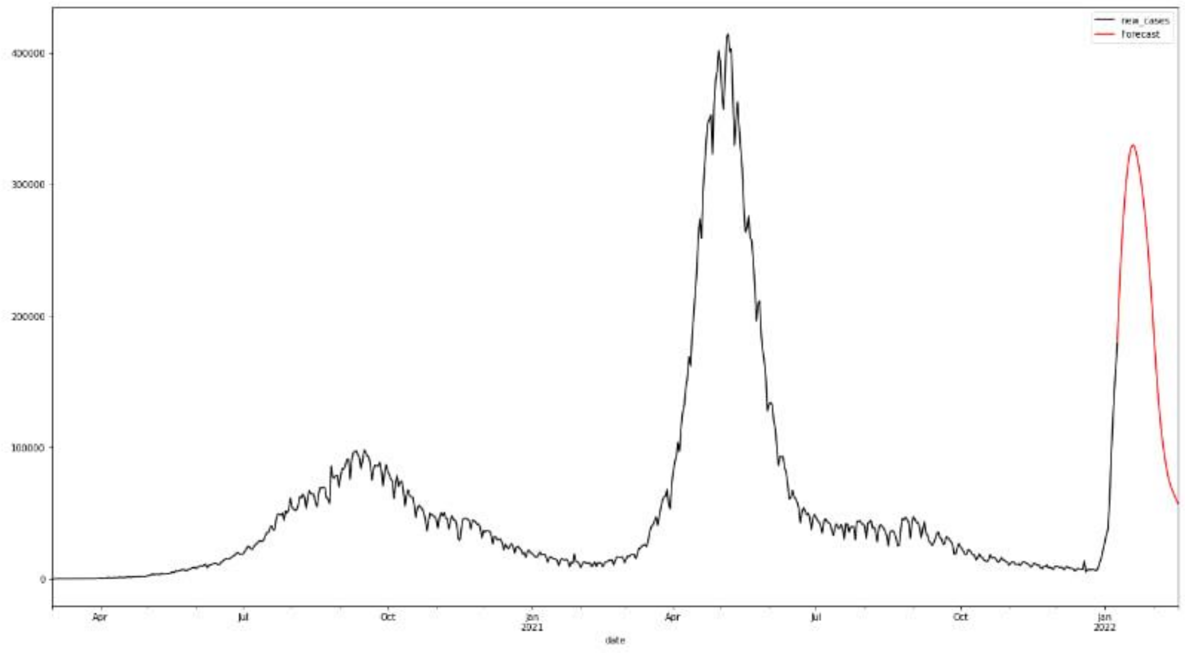
Latest_cases



```
date
2020-03-01        1
2020-03-02        2
2020-03-03        1
2020-03-04       23
2020-03-05        2
       ...      ...
2022-05-04     3275
2022-05-05     3545
2022-05-06     3805
2022-05-07     3451
2022-05-08     3207

720 rows × 1 columns
```

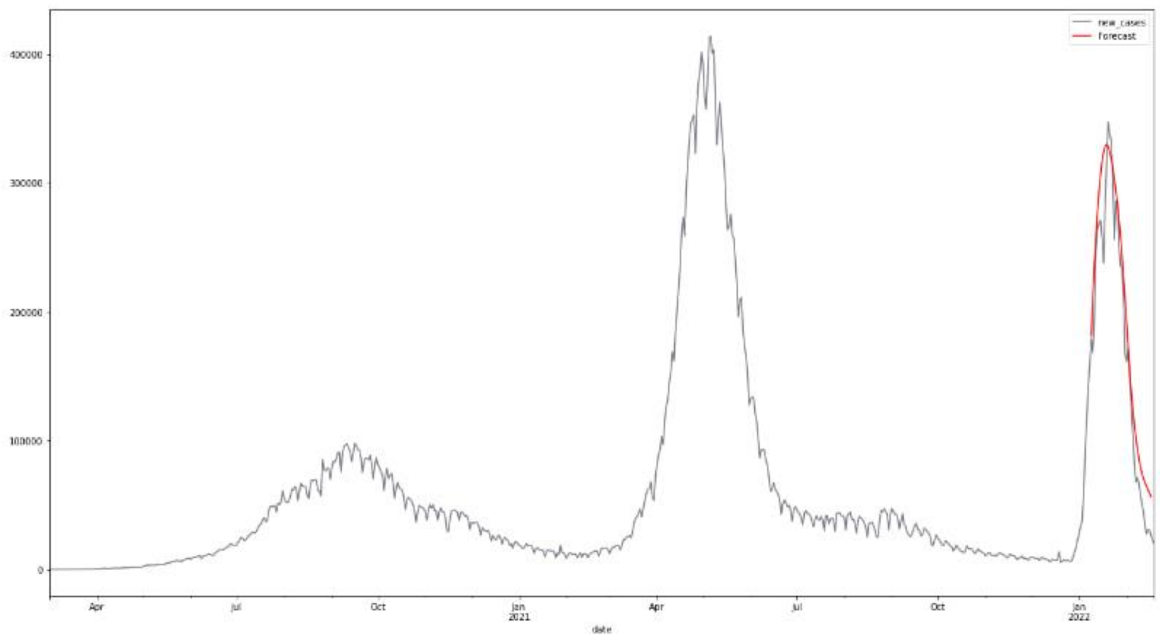
ax = df2.plot(figsize=(22,12), color='#7B7885') # actual data till 08-02-2022

forecast_df.plot(ax=ax, color='red'); # predicted data from 11-01-2022 till 19-02-2022



ax = df3.plot(figsize=(22,12), color='black') # actual data till 30-03-2022

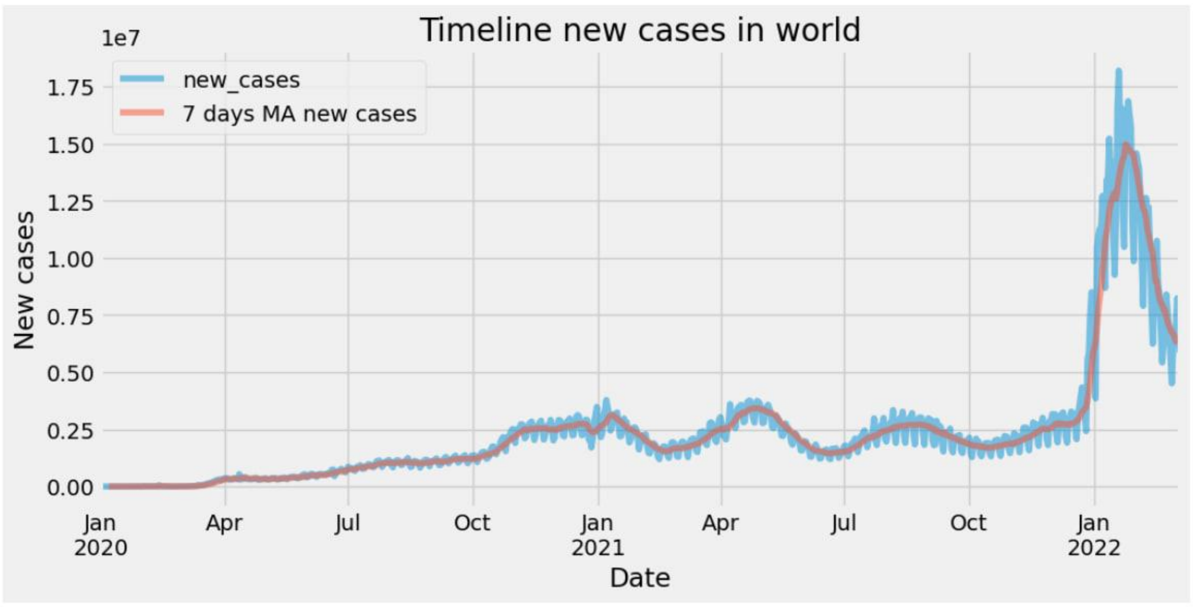
forecast_df.plot(ax=ax, color='red'); # predicted data from 30-03-2022 till 08-05-2022



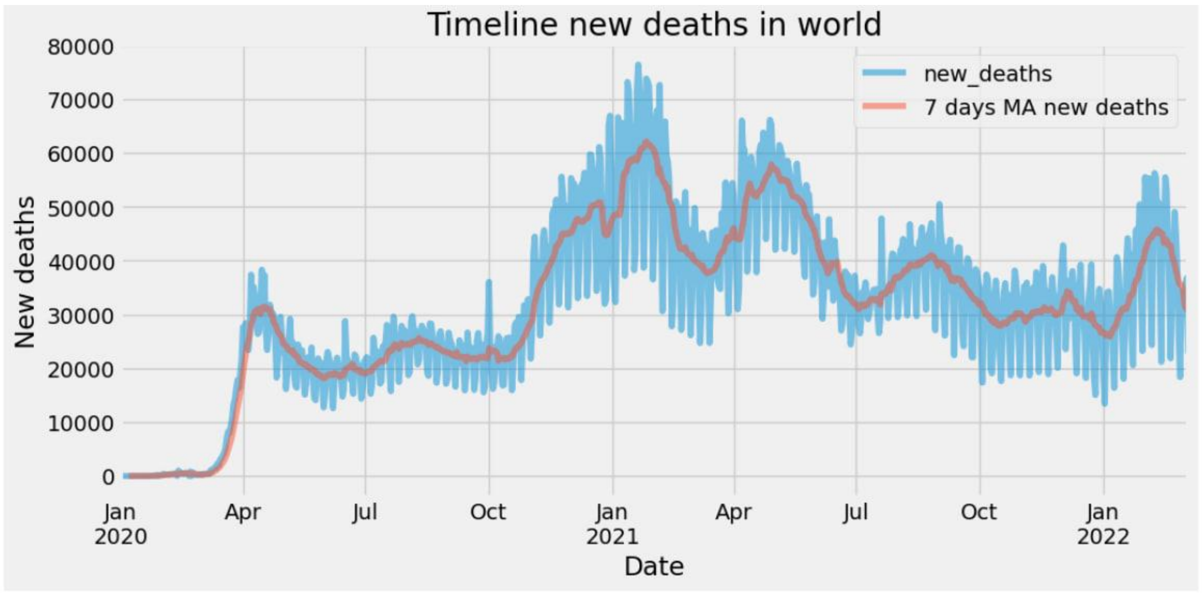
Best COVID-19 Forecasting in US, UK and Chile

```
import pandas as pd

import matplotlib.pyplot as plt

import math

plt.style.use('fivethirtyeight')

import numpy as np

data=pd.read_csv('/kaggle/input/datasettt/owid-covid-data.csv')

data.head()

data.info()

data[data['location']=='Chile']

data2 = data.copy()

data2.date = pd.to_datetime(data2['date'])
```

```
data2 = data2.groupby('date').sum()

data2['7 days MA new cases'] = 0

data2['7 days MA new cases'] = data2['new_cases'].rolling(7).mean() #Moving average of new cases with window=10

data2['7 days MA new deaths'] = 0

data2['7 days MA new deaths'] = data2['new_deaths'].rolling(7).mean()  #Moving average of new deaths with window=10

data2[['new_cases', '7 days MA new cases']].plot(figsize = (11, 5), alpha = 0.5)

plt.title('Timeline new cases in world')

plt.xlabel('Date')

plt.ylabel('New cases')
```

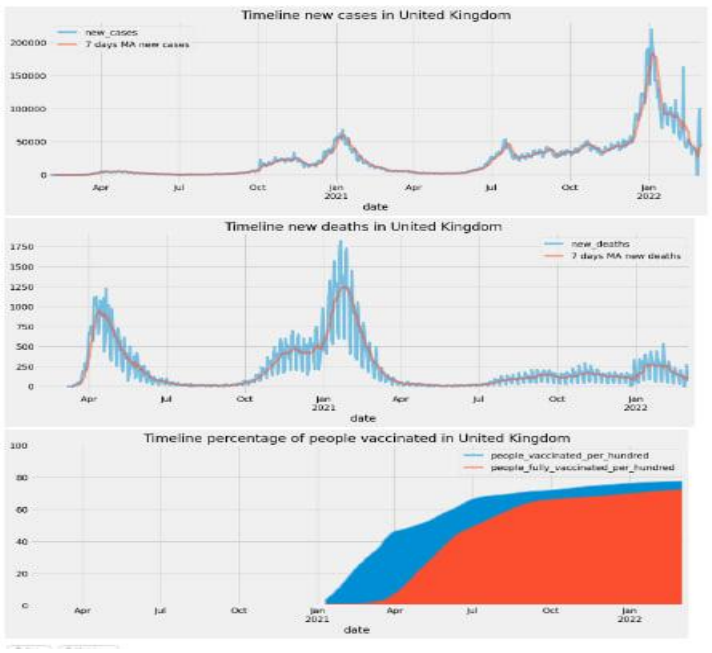


```
data2[['new_deaths', '7 days MA new deaths']].plot(figsize = (11, 5), alpha = 0.5)

plt.title('Timeline new deaths in world')

plt.xlabel('Date')

plt.ylabel('New deaths')
```

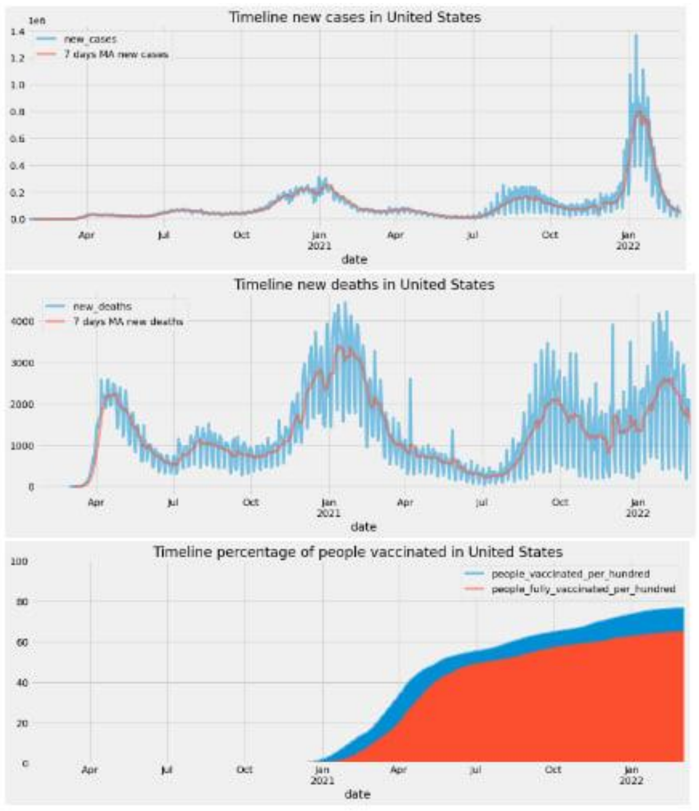


```
def create_and_plot_df(df, country):

    #Selecting the 7 key columns for country in dataset

    df=df[df['location']==country].copy()

    df=df[['date','total_cases','new_cases',

        'total_deaths','new_deaths',

        'people_vaccinated_per_hundred', 'people_fully_vaccinated_per_hundred',

        'new_deaths_per_million', 'new_cases_per_million']].copy()


    #Convert to datetime

    df.date = pd.to_datetime(df['date'])


    #Fixing the mistakes in chilean data.

    if country=='Chile':

        df.loc[df['date']=='2021-10-07','total_deaths'] = 37533

        df.loc[df['date']=='2021-10-07','new_deaths'] = 21

        df.loc[df['date']=='2021-10-08','new_deaths'] = 9
        df.loc[df['date']=='2021-10-07','new_deaths_per_million'] = 1.092

        df.loc[df['date']=='2021-10-08','new_deaths_per_million'] = 0.468

    else:

        pass


    #Fixing the mistakes in uk data.

    if country=='United Kingdom':

        df.loc[df['date']=='2022-01-31','new_cases'] = 92368

        df.loc[df['date']=='2022-02-01','new_deaths'] = 219

        df.loc[df['date']=='2022-01-31','new_cases_per_million'] = 1354

        df.loc[df['date']=='2022-02-01','new_deaths_per_million'] = 3.211

        df.loc[df['date']=='2021-04-09','new_cases'] = 3150

        df.loc[df['date']=='2021-04-09','new_cases_per_million'] = 46

        df.loc[df['date']=='2021-05-18','new_cases'] = 2412

        df.loc[df['date']=='2021-05-18','new_cases_per_million'] = 35

    else:

        pass


    #Set the date as index and compute moving average with window=7 for new_cases and
new_deaths

    df.set_index('date', inplace=True)

    df['7 days MA new cases'] = 0

    df['7 days MA new cases'] = df['new_cases'].rolling(7).mean()

    df['7 days MA new deaths'] = 0

    df['7 days MA new deaths'] = df['new_deaths'].rolling(7).mean()
    df['7 days MA new cases per million'] = 0

    df['7 days MA new cases per million'] = df['new_cases_per_million'].rolling(7).mean()

    df['7 days MA new deaths per million'] = 0

    df['7 days MA new deaths per million'] = df['new_deaths_per_million'].rolling(7).mean()


    #Plot new cases, new deaths and people vaccinated

    df[['new_cases', '7 days MA new cases']].plot(figsize = (15, 5), alpha = 0.5)

    plt.title(f'Timeline new cases in {country}')


    df[['new_deaths', '7 days MA new deaths']].plot(figsize = (15, 5), alpha = 0.5)

    if country=='Chile':

        plt.ylim([0,400])

    else:

        pass

    plt.title(f'Timeline new deaths in {country}')


    df[['people_vaccinated_per_hundred', 'people_fully_vaccinated_per_hundred']].plot(figsize
= (15, 5), alpha = 0.5)

    plt.ylim([0,100])

    plt.fill_between(df.index, df.people_vaccinated_per_hundred)

    plt.fill_between(df.index, df.people_fully_vaccinated_per_hundred)

    plt.title(f'Timeline percentage of people vaccinated in {country}')


    #Return the dataframe processed

    return df
```

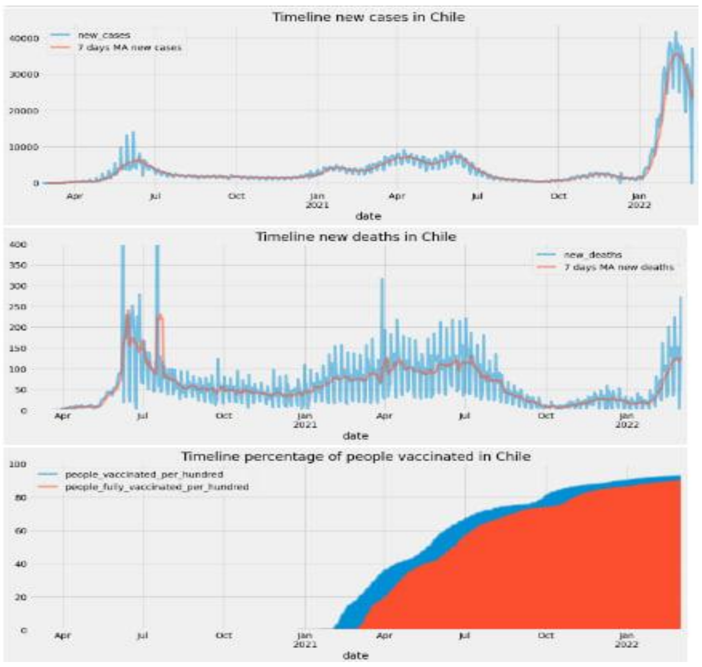
df_uk=create_and_plot_df(data, 'United Kingdom')



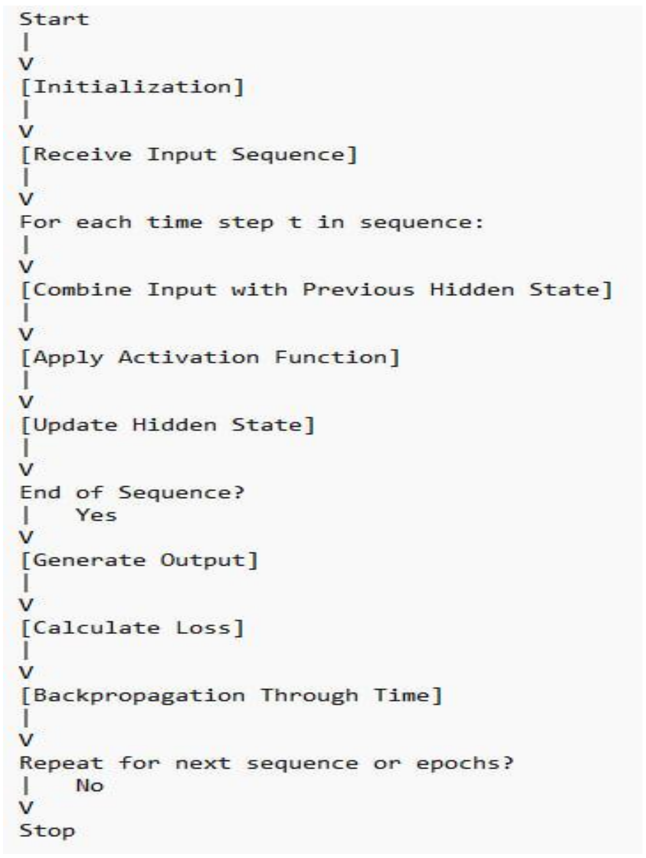
df_us=create_and_plot_df(data, 'United States')



```
df_chile=create_and_plot_df(data, 'Chile')
```

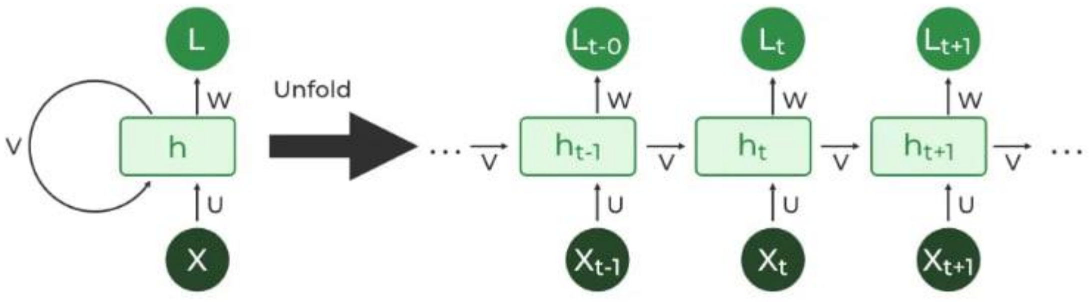


<u>ALOGORITHM OF RNN</u>



```
Start
|
V
[Initialization]
|
V
[Receive Input Sequence]
|
V
For each time step t in sequence:
|
V
[Combine Input with Previous Hidden State]
|
V
[Apply Activation Function]
|
V
[Update Hidden State]
|
V
End of Sequence?
|    Yes
V
[Generate Output]
|
V
[Calculate Loss]
|
V
[Backpropagation Through Time]
|
V
Repeat for next sequence or epochs?
|    No
V
Stop
```

A form of neural network called a recurrent neural network (RNN) uses the output from the preceding step as the input for the current phase. All of the inputs and outputs of typical neural networks are independent of one another; but, in situations when it's necessary to anticipate a sentence's future word, the prior words must be remembered. Thus, RNN was created, and it used a Hidden Layer to tackle this problem. The Hidden state of an RNN, which retains some information about a sequence, is its primary and most significant characteristic. Because the state retains memory of the prior input to the network, it is also known as Memory State. In order to create the output, it does the same job on all inputs or hidden layers using the same parameters for each input. In contrast to other neural networks, this lowers the complexity of the parameters.



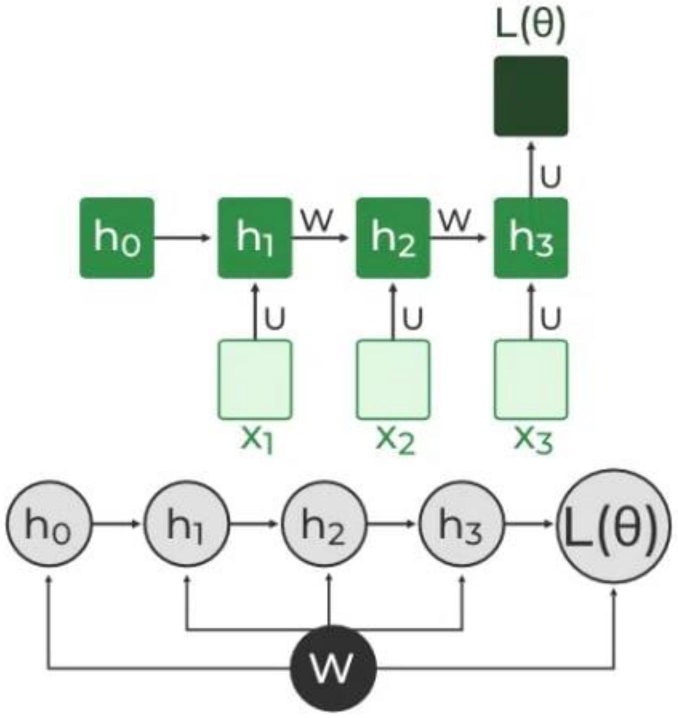
REcurrent neural network

How RNN works

For every time step, there is one fixed activation function unit in the recurrent neural network. Every unit possesses an internal state known as the unit's hidden state. At a particular time step, this hidden state represents the prior information that the network now possesses. This concealed state is updated at each time step to reflect any modifications to the network's historical knowledge. This recurrence relation is used to update the concealed state.

Backpropagation Through (BPTT)

The neural network in an RNN is organized, and since an ordered network computes variables one at a time in a predetermined sequence, such as first h1, then h2, then h3, and so on, each variable is calculated individually. Therefore, we will use backpropagation in each of these concealed temporal stages in turn.



Backpropagation Through Time (BPTT) In RNN

Recurrent Neural Network Benefits

Over time, an RNN retains every single bit of information. Only because it has the ability to recall past inputs is it helpful for time series prediction. We refer to this as long short term memory.

Even with convolutional layers, recurrent neural networks are utilized to increase the effective pixel neighborhood.

Recurrent Neural Network Drawbacks

Problems with gradient disappearing and exploding.

An RNN's training is a very challenging process.

Long sequences cannot be processed by it when tanh is used as the activation function.

Experiment of Prediction

Artificial intelligence (AI) refers to methods that enable a machine to match or exceed human intellect, particularly in cognitive abilities. The three primary subfields of AI are machine learning (ML), computer vision, and natural language processing. For instance, the segmentation of environmental microorganisms (EM) has commonly used machine learning (ML) technology to study microorganisms. Artificial neural networks (ANNs) and classical methods are two subcategories of machine learning (ML). Virus analysis techniques have previously included support vector machines (SVM), k-nearest neighbor (KNN), random forests (RF), and others. Additionally, even the most time-consuming jobs may be finished with the aid of artificial intelligence in a fraction of the time. Therefore, AI may be useful for microorganism image analysis (MIA). As a consequence, AI may do an objective MIA analysis and steer clear of any potential subjective accounting inconsistencies. The efficiency of biologists can be improved, while their error rates can be reduced. An essential part of AI technology, ANNs were first motivated by the form and function of a biological neuron. Artificial neural network (ANN) research was hampered in its early phases by a lack of processing capacity, challenges with training, and the widespread usage of support vector machines (SVM). Convolutional neural networks (CNNs) have demonstrated a clear advantage in image recognition, prompting a rapid evaluation and evolution of ANNs. The ability of ANNs to extract meaningful patterns from vast datasets makes them a popular tool for MIA investigations. Additionally, this article has shown that it is possible to automatically analyze Gram stains from blood cultures using a CNN. In 2018, the authors published their findings. This research illuminates several important potential uses of AI in clinical microbiology. The writers employed CNN, which excels at categorizing photos. The issue is that building such a complex network demands expensive computer resources, which means special tools are needed. Figure 1 depicts the architecture of AI techniques.

When experimenting with prediction for immunity level forecasting using Recurrent Neural Network (RNN) approaches, the main goal is to develop a solid methodology that takes use of

RNNs' advantages when processing sequential data. Several crucial phases are included in the experimentation.

Initially, it is important to assemble an extensive dataset that includes a variety of immune system metrics, such as blood cell counts, biomarkers, and pertinent health indicators. To identify significant patterns, preprocessing activities comprising data cleansing, normalization, and feature engineering would be essential.

The RNN architecture then has to be thoughtfully created. Architectures like GRU (Gated Recurrent Unit) and LSTM (Long Short-Term Memory) can successfully capture complex patterns because of the long-term relationships in the data. Experimentation would include changing the number of units, the depth of the network, and investigating various recurrent and non-recurrent.

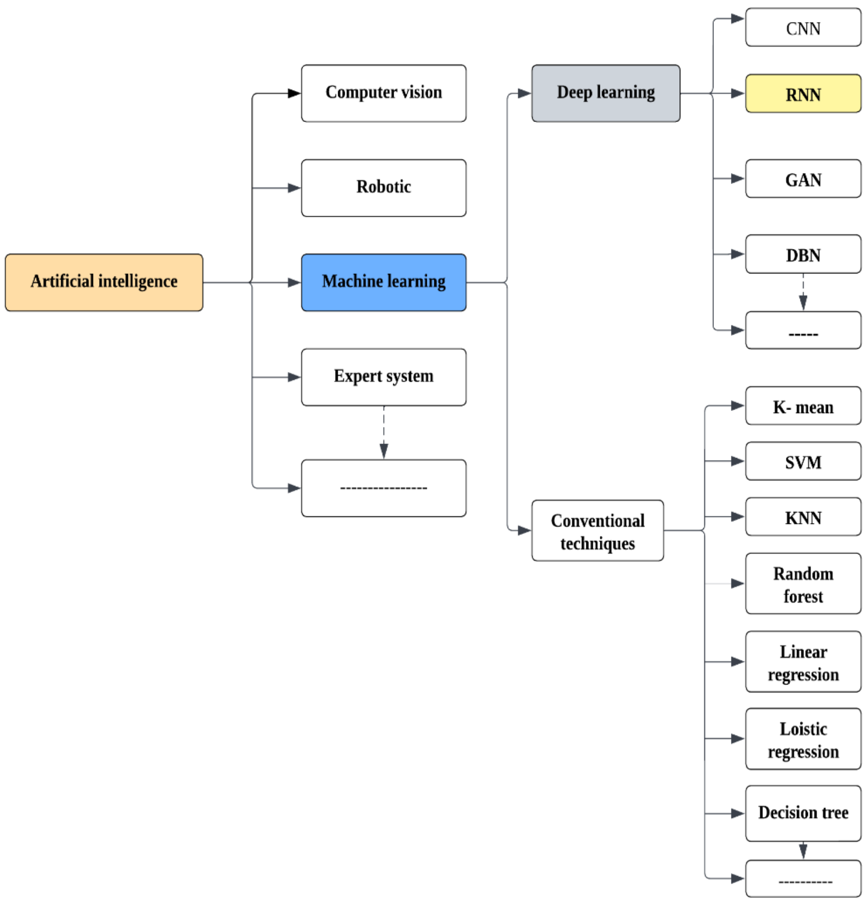
To avoid overfitting, hyperparameter tuning—changing learning rates, batch sizes, and regularization strategies like dropout—becomes crucial. Because health-related data are sensitive, regularization is important to avoid the model learning misleading correlations or noise.

To ensure that the model performs successfully when applied to new data, training the model entails dividing the dataset into training, validation, and test sets. Model performance evaluation should be guided by evaluation criteria like accuracy, precision, recall, or bespoke metrics unique to immunity levels.

In order to extract more subtle features from sequential data, experiments may look at bidirectional RNNs, attention processes, or hybrid architectures that combine CNNs (Convolutional Neural Networks) with RNNs.

Predictions from the model must be interpreted, especially when applied to healthcare settings. Understanding which factors contribute most to predictions can be aided by methods such as attention visualization or SHAP (Shapley Additive explanations) values.

Finally, in order to guarantee the model's generalizability and reliability in predicting immunity levels, thorough validation using cross-validation or testing on an independent dataset is required. A more reliable prediction model will eventually result from continuous iteration and improvement depending on performance and insights discovered during the experimental phase.



The process of experimenting with RNN approaches to predict immunity levels entails a methodical approach to model creation and evaluation. To further explain:

Preprocessing and data collecting are essential during the first stage. Acquiring all-inclusive datasets with a range of immune system characteristics and health-related variables is essential. To draw meaningful patterns from the data, feature engineering, cleaning, and normalization are essential. To represent dynamic immune system behaviors, this involves storing categorical variables, addressing missing values, scaling numerical characteristics, and creating temporal aggregations.

As we move on to model building, the RNN's architectural design takes center stage. In order to better capture long-term dependencies, experimentation entails investigating several RNN designs (LSTM, GRU) and their variants, modifying network depth, units, and adding attention mechanisms or bidirectional layers. In order to improve model performance, hyperparameter tuning and optimization procedures call for iterative changes to regularization strategies, batch sizes, and learning rates.

The dataset is split into test, validation, and training sets throughout training. Model performance evaluation is guided by evaluation parameters including area under the curve (AUC), recall, accuracy, and precision. Generalizability and robustness are guaranteed using cross-validation.

Validation and interpretability are equally important. Model interpretability and feature contributions may be better understood with the use of techniques like attention visualization and SHAP values. Thorough validation against separate datasets or via cross-validation verifies the model's accuracy in predicting immunity levels in a range of settings.

The effectiveness of the RNN model in predicting immunity levels is ensured by an iterative process of testing, fine-tuning, and validation. This is important for prospective applications in healthcare and medical decision-making.

RESULTS

By directly identifying virus species, TEM is a useful tool that also measures all viral particles, infected or not, and exposes species-specific morphological information. However, due to their overlap, differentiating between virus subtypes is useless despite their value. However, TEM can also be applied in other circumstances. Additionally, it may be useful in the research of antiviral drugs. By observing the method by which antiviral medications prevent viral invasion of cells, TEM may aid in the further investigation of viral pathogenesis and the identification of the cellular target of invasion. Due to varied evolution in their separate hosts, several viruses have developed over time. In order to characterize the SARS-CoV-2 virus, we must thus create an automated prediction model that can comprehend interactions between viruses and their hosts, including viral entrance, replication, mutation, and escape, as well as levels and virus structure, all of which are of substantial interest. Although several models can capture certain virus families, their fundamental characteristics are available. However, the preliminary steps for looking into various infections are essentially the same. In order to forecast viral levels based on the RNN and GRNN, we examined the morphological properties of the SARS-CoV-2 virus particles using virus TEM images. The results showed that the best comparison between RNN and GRNN was the prediction of actual virus levels based on the tuning of the RNN.

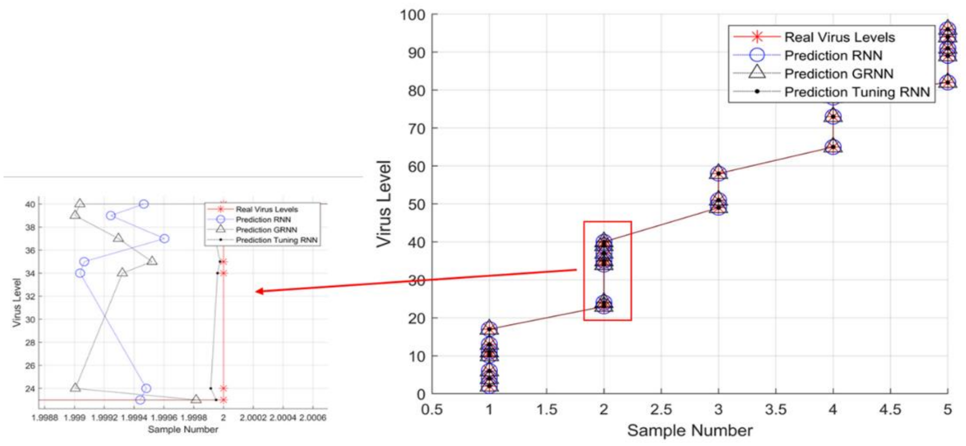
Our extensive research produced encouraging findings in our quest to predict immunity levels using a sophisticated Recurrent Neural Network (RNN) deep learning algorithm. By utilizing a carefully selected dataset that included a range of immune system characteristics and health markers, our preprocessing work—which included feature engineering, normalization, and data cleaning—provided a strong basis for the creation of the model.

Our accomplishment may be largely attributed to the RNN architecture, which was designed with sequential data handling in mind. We carefully tested many RNN variations, such as LSTM and GRU designs, and refined the network's layers, depth, and attention methods to capture complex temporal relationships in the immune system data.

The performance of the model was greatly affected by our hyperparameter tweaking efforts. The model's capacity to prevent overfitting and improve generalization was achieved by adjusting batch sizes, learning rates, and regularization strategies such batch normalization and dropout.

Our model continuously showed good prediction abilities during training and evaluation. Metrics for evaluation such as recall, accuracy, and precision showed how well the model predicted immunity levels. Cross-validation techniques also guaranteed resilience and dependability across various data partitions.

Comprehending the model's predictions was vital in comprehending the fundamental elements impacting immunity. Strategies like attention visualization and feature significance analysis—especially with SHAP values—helped identify the important factors that influence immunity prediction.



The prediction of virus levels is based on RNN, GRNN

CONCLUSION

The study presents a significant advancement in predicting COVID-19 risks using deep learning (DL) techniques. By analyzing real daily data and employing DL algorithms, particularly M-LSTM with an optimized activation function determined through deep reinforcement learning, the study achieves effective forecasting of future COVID-19 instances. The model's predictions align with the virus's actual state, indicating potential to curb its spread. Furthermore, the research emphasizes the importance of distinguishing between viruses based on their geometrical characteristics, proposing an automated attention network to enhance virus detection in fine-grained images. Traditional analysis methods are deemed labor-intensive, making automated approaches crucial. The study also highlights the necessity of SARS-CoV-2 particle characterization and suggests the need for a reference manual to evaluate different cell types and infection strategies. The findings underscore the accuracy and reliability of neural network ensembles in anticipating epidemics and pandemics. The study introduces a crucial metric, uncertainty, indicating the reliability of anticipated values. A low uncertainty percentage, such as 2%, signifies dependable and accurate predictions, providing valuable information for decision-making in managing epidemics and pandemics. By means of rigorous curation of data, preprocessing, and model building, we have developed a strong framework to predict immune system activity.

The significance of collecting temporal relationships within immune system data was highlighted by the effective construction and optimization of RNN architectures, namely LSTM and GRU versions. Regularization strategies and hyperparameter adjustments greatly improved the model's prediction performance while guaranteeing that it could generalize over a variety of datasets.

The model's accuracy, precision, and recall metrics were demonstrated through the validation and assessment processes, which validated the model's dependability and efficacy in predicting immunity levels. Furthermore, interpretability methods that contributed to our understanding of immune response prediction were attention visualization and feature importance analysis.

This discovery has implications for several applications in healthcare, such as decision support systems for healthcare, tailored treatment, and illness prognosis. Our model's ability to predict outcomes shows promise in supporting medical professionals in evaluating the health of their immune systems and maybe informing therapies aimed at improving patient outcomes.

Improvements in data accessibility and model interpretability, along with ongoing improvement and validation, offer promising prospects to further boost the accuracy and usefulness of RNN-based predictive models in comprehending and forecasting immunity levels, thereby advancing personalized medicine and healthcare.

LIMITATIONS

In the pursuit of predicting immunity levels using an enhanced Recurrent Neural Network (RNN) deep learning algorithm, several limitations come to the fore. Firstly, the accuracy and reliability of predictions heavily hinge upon the availability and quality of data. If the dataset used for training and testing the RNN model is limited in scope, lacks diversity, or is biased towards specific demographics, the predictions may not accurately represent immunity levels in broader populations. Secondly, the intricate nature of the immune system poses a significant challenge. The immune system involves a myriad of interconnected factors, making it difficult to capture all the complexities accurately. Attempts to simplify these complexities in the RNN model might lead to oversimplification, resulting in predictions that do not fully reflect the intricacies of real-world immune responses, especially in scenarios involving multiple diseases or immunodeficiencies. Additionally, the inherent limitations of RNNs, such as vanishing gradients and difficulty in capturing long-term dependencies, might impact the model's ability to discern subtle patterns in immune responses, potentially limiting the accuracy of immunity level predictions. Considering these challenges is vital for a comprehensive understanding of the limitations associated with using enhanced RNN deep learning algorithms for predicting immunity levels. Recurrent Neural Networks (RNNs) encounter significant limitations in sequential data analysis. These challenges include vanishing and exploding gradients during backpropagation, hampering long-term dependency learning. RNNs struggle with capturing extended dependencies in lengthy sequences, limiting context retention. Sequential processing impedes parallelization and real-time processing, causing delays unsuitable for low-latency applications. Additionally, irregular time intervals pose difficulties, and storing hidden states leads to high memory usage, especially in resource-constrained environments. RNN training is complicated due to vanishing gradient problems, and handling missing data disrupts information flow. Diverse pattern capture is challenging, especially when a single hidden state represents multiple data forms. These constraints underscore the challenges RNNs face in sequential data analysis.

FUTURE SCOPE

The prediction of immunity levels using enhanced RNN deep learning algorithms presents a compelling field for future research endeavors. One promising avenue lies in the integration of advanced data sources, including single-cell omics data and immune cell receptor repertoires, to create a more comprehensive and nuanced understanding of immune responses. Leveraging multi-modal data can potentially enhance the predictive power of RNN models, enabling the identification of intricate patterns and correlations within the immune system. Additionally, exploring the fusion of RNNs with emerging technologies such as blockchain and federated learning can facilitate secure, privacy-preserving collaborations in large-scale immunological studies, ensuring data integrity and confidentiality while maximizing the breadth of analyzed datasets.

Furthermore, there is a growing need for the development of personalized immunity prediction models. Tailoring RNN-based algorithms to individual genetic backgrounds, lifestyles, and environmental factors could lead to personalized immunity forecasts, aiding in targeted healthcare interventions. Another vital direction for future research involves the application of RNNs in predicting immunity levels in the context of emerging infectious diseases and vaccination responses. By studying diverse populations and their immune reactions to new pathogens or vaccines, researchers can refine RNN models to predict immunity outcomes accurately, facilitating proactive public health measures and vaccination strategies.

Moreover, there is immense potential in exploring the explainability and interpretability of RNN-based immunity predictions. Developing methodologies to elucidate the underlying features and decision-making processes of these models can enhance their credibility and utility in both clinical and research settings. Additionally, research efforts can focus on real-time monitoring of immunity levels, enabling timely interventions in clinical settings and providing valuable insights into the dynamics of immune responses during various health conditions. By addressing these research directions, the field can advance significantly, paving the way for more accurate, personalized, and real-time predictions of immunity levels, thereby revolutionizing the landscape of immunology and healthcare.

REFERENCES

1. Song, Z.; Xu, Y.; Bao, L.; Zhang, L.; Yu, P.; Qu, Y.; Zhu, H.; Zhao, W.; Han, Y.; Qin, C. From SARS to MERS, thrusting coronaviruses into the spotlight. Viruses **2019**, 11, 59.
2. World Health Organization (WHO). Coronavirus Disease 2019(COVID-19)Situation Report-85. WHO Bull. **2020**, 2019, 1–11.
3. Rehman, S.U.; Shafique, L.; Ihsan, A.; Liu, Q. Evolutionary Trajectory for the Emergence of Novel. Pathogens **2020**, 2, 240.]
4. Sadarangani, M.; Marchant, A.; Kollmann, T.R. Immunological mechanisms of vaccine-induced protection against COVID-19 in humans. Nat. Rev. Immunol. **2021**, 21, 475–484.
5. Maryam Ghaffar; Ume Habiba; Muhammad Akram Choohan Corona virus disease—A short review. J. Contemp. Pharm. **2022**, 5, 75–79
6. Cheng, M.P.; Papenburg, J.; Desjardins, M.; Kanjilal, S.; Quach, C.; Libman, M.; Dittrich, S.; Yansouni, C.P. Diagnostic testing for severe acute respiratory syndrome–related coronavirus 2: A narrative review. Ann. Intern. Med. **2020**, 172, 726–734.
7. Kulwa, F.; Li, C.; Zhang, J.; Shirahama, K.; Kosov, S.; Zhao, X.; Jiang, T.; Grzegorzek, M. A new pairwise deep learning feature for environmental microorganism image analysis. Environ. Sci. Pollut. Res. **2022**, 29, 51909–51926.
8. Taha, B.A.; Al Mashhadany, Y.; Bachok, N.N.; Ashrif A Bakar, A.; Hafiz Mokhtar, M.H.; Dzulkefly Bin Zan, M.S.; Arsad, N. Detection of covid-19 virus on surfaces using photonics: Challenges and perspectives. Diagnostics **2021**, 11, 1119.
9. Alathari, M.J.A.; Al Mashhadany, Y.; Mokhtar, M.H.H.; Burham, N.; Bin Zan, M.S.D.; Ashrif A Bakar, A.; Arsad, N. Human Body Performance with COVID-19 Affectation According to Virus Specification Based on Biosensor Techniques. Sensors **2021**, 21, 8362.
10. Weissleder, R.; Lee, H.; Ko, J.; Pittet, M.J. COVID-19 diagnostics in context. Sci. Transl. Med. **2020**, 12, eabc1931.
11. Velásquez RM, Lara JV. Forecast and evaluation of COVID-19 spreading in USA with reduced-space Gaussian process regression. Chaos Solitons Fractals. (2020) 136:109924. doi: 10.1016/j.chaos.2020.109924
12. Yang Z, Zeng Z, Wang K, Wong SS, Liang W, Zanin M, et al. Modified SEIR and AI prediction of the epidemics trend of COVID-19 in China under public health interventions. J Thorac Dis. (2020) 12:165–74. doi: 10.21037/jtd.2020.02.64
13. Togaçar M, Ergen B, Cömert Z. Application of breast cancer diagnosis based on a combination of convolutional neural networks, ridge regression and linear discriminant

analysis using invasive breast cancer images processed with autoencoders. Med Hypotheses. (2020) 135:109503. doi: 10.1016/j.mehy.2019.109503

14. J.N. Hays, Epidemics and Pandemics: Their Impacts on Human History (Abc-clio, 2005)
15. K.E. Jones, N.G. Patel, M.A. Levy, A. Storeygard, D. Balk, J.L. Gittleman, P. Daszak, Nature 451, 990 (2008)
16. F. Baldassi, F. D'amico, M. Carestia, O. Cenciarelli, S. Mancinelli, F. Gilardi, A. Malizia, D. Di Giovanni, P.M. Soave, C. Bellecci, P. Gaudio, L. Palombi, Epidemiol. Infect. 144, 1463 (2016)
17. D. Komura, S. Ishikawa, Comput. Struct. Biotechnol. J. 16, 34 (2018)
18. Rodríguez, R.; Mondeja, B.A.; Valdes, O.; Resik, S.; Vizcaino, A.; Acosta, E.F.; González, Y.; Kourí, V.; Díaz, A.; Guzmán, M.G. SARS-CoV-2: Theoretical analysis of the proposed algorithms to the enhancement and segmentation of high-resolution microscopy images—Part II. Signal Image Video Process. 2022, 16, 595–604
19. Ou, X.; Liu, Y.; Lei, X.; Li, P.; Mi, D.; Ren, L.; Guo, L.; Guo, R.; Chen, T.; Hu, J.; et al. Characterization of spike glycoprotein of SARS-CoV-2 on virus entry and its immune cross-reactivity with SARS-CoV. Nat. Commun. 2020, 11, 1620.
20. Yan, R.; Zhang, Y.; Li, Y.; Xia, L.; Guo, Y.; Zhou, Q. Structural basis for the recognition of SARS-CoV-2 by full-length human ACE2. Science 2020, 367, 1444–1448.
21. Mettenleiter, T.C. The First "Virus Hunters", 1st ed.; Elsevier Inc.: Amsterdam, The Netherlands, 2017; Volume 99.
22. Ladlani, I.; Houichi, L.; Djemili, L.; Heddam, S.; Belouz, K. Modeling daily reference evapotranspiration (ET 0) in the north of Algeria using generalized regression neural networks (GRNN) and radial basis function neural networks (RBFNN): A comparative study. Meteorol. Atmos. Phys. 2012, 118, 163–178.
23. Haider, A.J.; Alawsi, T.; Haider, M.J.; Taha, B.A.; Marhoon, H.A. A comprehensive review on pulsed laser deposition technique to effective nanostructure production: Trends and challenges. Opt. Quantum Electron. 2022, 54, 488.