Forecasting COVID-19 Trends in 2021: Comparing RNNs and Time-Series Transformers

Haider Ali
haiderali@vt.edu
Virginia Tech, USA

Yusuf Elnady
yelnady@vt.edu
Virginia Tech, USA

ABSTRACT

There has been a lot of research work, effort, and publications related to COVID-19. After more than a year has passed, there is a lot of data available to analyze and predict what the future holds for us. In this project, we have used a large dataset that is constantly updated to predict the future trends of COVID 19 in the year 2021. We have focused mainly on the forecasting of the number of confirmed cases, deaths, and administrated vaccines in the chosen five states of the United States. We used a variety of different deep learning models to compare the results such as GRU, LSTM, and Time-Series Transformers for forecasting. In evaluation, we have also shown the efficacy of these deep learning models by comparing the results with baseline models such as Prophet models and the SARIMA model. We used metrics such as Mean Absolute Percentage Error (MAPE) and R2 Score for the evaluation of these models. Out of all these models, Time-Series Transformers outperformed.

CCS CONCEPTS

• **Time-Series Analysis**; • **Sequential Modeling**; • **Transformers**; • **Deep Learning**; • **RNN models**; • **Temporal Correlations**; • **Attention Mechanism**; • **Natural Language Generation**;

KEYWORDS

COVID-19, Time-series Dataset, RNNs, Pandemic, Deaths Detection, Transformers, Transmission Rate, LSTMs and GRUs, Clustering, Self Attention, Time-series Forecasting, EDA, Real-Valued Data, Processing Sequential Data, Model Evaluation;

1 INTRODUCTION

COVID-19, a pandemic, has badly affected mankind. First originated in December 2019 in Wuhan China, it became a pandemic in no time. Every country has faced the consequences of COVID-19. It came to the country with one person and in a couple of months, it not only caused several cases but also numerous deaths. Countries suffered from severe lockdowns and everything got shut down all of a sudden. Quality of life was disturbed at its peak. The sudden surge of cases caused the scarcity of crucial emergency equipment like ventilators and oxygen not only in developing countries but also in developed countries.

This situation is still not under control after more than a year. More and more variants of COVID-19 are originating which are not only dangerous but also causing more deaths. These days, India is suffering a lot from new variants of COVID-19 and there is a huge scarcity of oxygen supply in India. If India could have predicted the surge of cases earlier, it would have been in a much less critical situation. They could have put lockdowns to stop the people from their religious festivals.

To take more informed actions about lockdowns and efficient emergency resources management, there is a need to predict the next COVID-19 waves in a certain area. There is a need to predict the number of cases and number of deaths. It's hard to see that even though we already have a vaccine in hand, we are still losing our people due to COVID-19. There is a need to predict when all of the people will be vaccinated in a certain area.

To make our predictions more meaningful, we just focused on five states of the United States that is to cover two of the highest affected states, the lowest affected states, and one intermediate state. The five states are New York, California, Virginia, Northern Mariana Islands, and the Virgin Islands. We have focused mainly on the forecasting of the number of confirmed cases, deaths, and vaccines. We are mapping this problem to a univariate or multivariate time series forecasting problem which we tried to solve using state-of-the-art Deep Learning models. Previously, Ayris et al. [1] has worked on a similar problem but they haven't leveraged the state-of-the-art methods of Deep Learning, which made this research novel. Deep Learning has the power of capturing the hidden trends necessary for forecasting. We used Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU), and Time Series Transformers (TST). We also evaluated these models using the baselines: Facebook Prophet Model and Seasonal Autoregressive Integrated Moving Average (SARIMA). We used metrics such as Mean Absolute Percentage Error (MAPE) and R2 Score for the evaluation of these models which is common in regression tasks. In this project, we show that out of all these models, the Time-Series Transformers perform best on the account of the number of parameters. We also provide all of our code at this GitHub repository HaiderAli and Elnady [6].

2 RELATED WORK

2.1 Use of DL in COVID-19 Problems

There is a medical image processing domain where machine learning techniques were used for COVID-19. CNN was applied on COVID-19 patient's chest X-ray images to overcome the shortage of radiologists for early detection in Ozturk et al. [14]. There are several sets of papers using CNN models and transfer learning techniques on CT scans for detecting and tracking COVID-19 [8]. Apart from the vision domain, machine learning and deep learning techniques have been various times used in time series problems of COVID-19. There is a need to predict the number of cases and deaths for efficient resource management and planning. Zeroual et al. [20] used five deep learning models including LSTMs and GRUs for a task similar to us. But they did not use the latest techniques like time series transformers, which in our case lead to better results as compared to LSTMs and GRUs. They also did not consider the different states of the United States. They did on confirmed and recovered cases of six different countries. However, we focused

on the forecasting of cases, deaths, and vaccines of five different states of the USA. They also used MAPE as one of their evaluation metrics. Machine learning techniques like SVM and Exponential smoothing were used to predict COVID-19 future in Rustam et al. [16]. Our work uses more sophisticated models and more data to address this issue, showing the efficacy of our work. In Chimmula and Zhang [3], LSTM was used to predict COVID-19 spread in three countries including the USA. Stacked Autoencoders were also used for forecasting on China dataset in Hu et al. [9].

2.2 Use of Deep Learning in other Time Series Problems

There are several deep learning algorithms proposed by 2021 survey paper Torres et al. [18]: Deep feed-forward neural network (DFFNN), RNNs, LSTM, GRU, Bidirectional RNN, Deep RNN or stacked RNN, CNN, and Temporal Convolutional Networks. These architectures have been used in many time series applications including the health domain. Cardiovascular disease problems were being solved using DFFNN in Bui et al. [2]. CNN was also used to monitor the sleep stage in Liu et al. [11] for detecting premature problems as ventricular contractions or to forecast the Sepsis. In da Silva et al. [5], LSTM architecture was used for the early detection of patient health deterioration on time series data. We are also using deep learning models such as LSTM and Transformers for health-related time series problems. None of the work previously tackled this problem with transformers, which makes our work unique.

3 PROBLEM STATEMENT

We focus on data related to COVID-19 in the United States. Based on the datasets we collected, we consider the following tasks as a primary basis for our project:

- Forecasting the expected number of confirmed cases in the United States (per state).
- Forecasting the expected number of deaths in the United States (per state).
- Forecasting the expected number of vaccinated people cases in the United States (per state).
- Comparing a variety of deep learning models on a set of time-series forecasting tasks.
- Predicting when all people of a particular state will be vaccinated.
- Reporting useful hidden patterns in the dataset for the sake of future use.

4 DATASETS

We consider two datasets for our work. The first one is for the cases and deaths, and the second one is for the vaccines.

4.1 Confirmed Cases and Deaths Dataset

We have used a dataset by New York Times [13]. It is a time-series data that spans 16 months starting from January 2020 to May 5, 2021, and is updated daily. It had around 418 data points till April 22, 2021. This dataset contains the following state-wise information of the USA:

- Date: Observation date in mm/dd/yyyy
- State: State of the USA
- Cases: Cumulative counts of coronavirus cases till that date
- Deaths: Cumulative counts of coronavirus deaths till that date
- FIPS codes: standard geographic identifier

FIPS code is a standard geographic identifier which makes it easier for an analyst to combine this data with other data sets, and to visualize the dataset as shown in the figures 5 and 6. A sample picture of the dataset can be seen in Figure 1. There was a similar dataset on Kaggle [17] but that dataset is not being constantly updated and that was only till February 2021. That is why we chose this dataset maintained by New York Times.
After converting into sliding window chunks (more details in the pipeline section), We split this dataset into train and test datasets with the ratio of 80:20. The split is not randomly done, rather the first 80% of the days is the training data, and the last 20% days are the ones we use and evaluate on them.

date	state	fips	cases	deaths
1/21/20	Washington	53	1	0
1/22/20	Washington	53	1	0
1/23/20	Washington	53	1	0
1/24/20	Illinois	17	1	0
1/24/20	Washington	53	1	0
1/25/20	California	6	1	0
1/25/20	Illinois	17	1	0
1/25/20	Washington	53	1	0
1/26/20	Arizona	4	1	0
1/26/20	California	6	2	0

Figure 1: The structure of the cases and deaths dataset

4.2 Vaccinations Dataset

We have used a dataset [12] by 'Our World In Data' for state-level vaccinations. This time-series data starts from January 13, 2021, till May 5, 2021. It is constantly being updated daily, and contains the following features:

- Date
- State Name
- Daily count of vaccinations

This dataset is quite smaller since it only contains data of three months and a half. We decided to reduce the size of the sliding window to increase the number of samples, and to split the dataset into training and test datasets using the ration 70:30, to have enough days to evaluate them.

5 EXPLORATORY DATA ANALYSIS AND SOME USEFUL HIDDEN TRENDS

5.1 Reason for Choosing 5 States

There was a total of 55 states and territories. Out of those, we selected five of the states. We have taken the two least important, one intermediate, and two most important states. We define the importance of the state as the state where the number of deaths is very high. The top 2 were California and New York. We chose the intermediate state like Virginia. The lower 2 states were the

Northern Mariana Islands and the Virgin Islands. The reason for choosing these states is to add diversity to our time series models.

We sorted the states by the number of deaths in ascending order. The top 10 in that list is shown in Figure 2. The bottom 10 in that list is shown in Figure 3. The selected states are shown in Figure 4. In these figures, the first column shows the state, the second column shows the total number of cases till April 22, 2021, and the total number of deaths till April 22, 2021.

	state	cases	deaths
0	Northern Mariana Islands	162	2
1	Virgin Islands	3068	27
2	Guam	8878	137
3	Vermont	22325	243
4	Alaska	66370	318
5	Hawaii	31977	473
6	Wyoming	57613	705
7	Maine	58868	769
8	District of Columbia	47040	1098
9	New Hampshire	92911	1274

Figure 2: Cases and Deaths of low death states, sorted by Deaths

45	Michigan	900423	18239
46	Ohio	1060119	19033
47	Georgia	1065319	19258
48	Illinois	1316497	24056
49	New Jersey	990580	25282
50	Pennsylvania	1128144	25934
51	Florida	2191030	34695
52	Texas	2868207	49984
53	New York	2016244	51299
54	California	3727713	61261

Figure 3: Cases and Deaths of high death states, sorted by Deaths

	state	cases	deaths
Northern Mariana Islands		162	2
Virgin Islands		3068	27
Virginia		650981	10653
New York		2016244	51299
California		3727713	61261

Figure 4: The cases and deaths of selected 5 States

5.2 Cases and Deaths Analysis

Figure 5 shows the cumulative cases of different states in the USA until April 2021. We can see the dark red color in California, Florida, New York, and Texas for the cases. We also have noticed the high number of deaths in California and Texas as shown in Figure 6. This figure shows the cumulative deaths until April 2021.

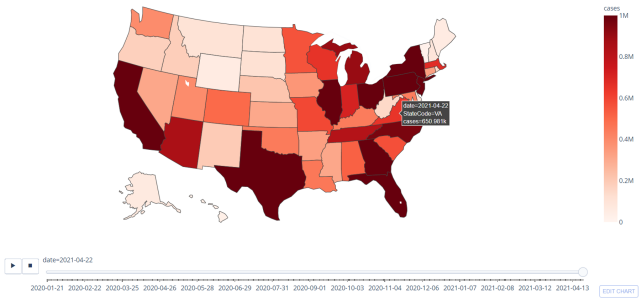

Figure 5: Cumulative Cases for whole USA

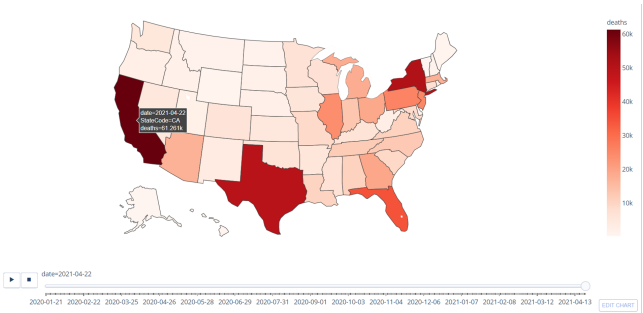

Figure 6: Cumulative Deaths for whole USA

5.3 Vaccinations Analysis

For 5 of the states, daily vaccinations can be seen in Figure 7. It increased first but then there is a decrease for 3 of the states: California, New York, and Virginia. The possible reason may be due

State	Vaccinated People	Total People	% Vaccinated
Northern Virginia Islands	22745	57216	40%
Virgin Islands	34148	106631	32%
Virginia	4.13 M	8.536 M	48%
New York	9.52 M	19.45 M	49%
California	19.79 M	39.51 M	50%

Table 1: Percentage of People Vaccinated

to some blood clot cases after people got the J&J vaccine recently, which may have created fear in people. Figure 9 shows the total number of vaccination doses received by people in 5 of the states. Figure 8 shows the total number of people who have received at least one dose of vaccine. More than 40 % of the people in these 5 states are vaccinated as shown in Table 1. There is still a need to vaccinate the remaining 60% of the population in these states. This proves the need of predicting when people will be vaccinated as we can see in Figure 7 that there is a decline in daily vaccines these days.

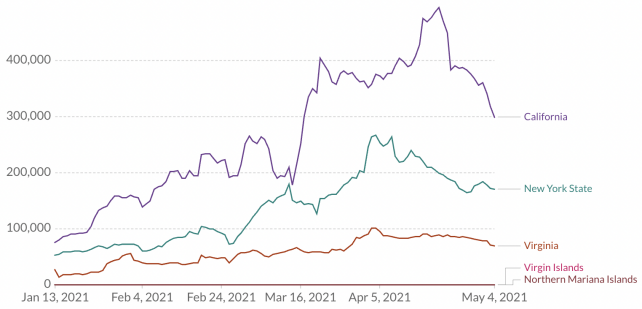

Figure 7: Daily Vaccinations of 5 States

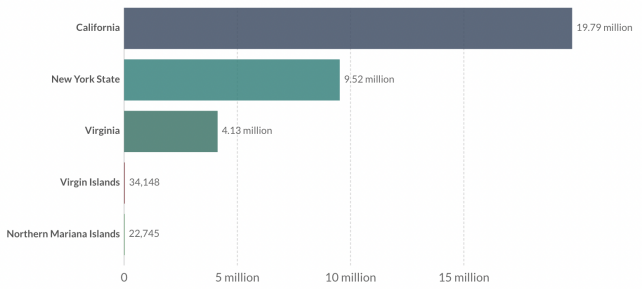

Figure 8: Total People Vaccinated of 5 States

5.4 Hidden Trends

We have observed a decline in the number of people getting the vaccination after April 15, which clearly showed that they are afraid of the blood clot cases originated from J&J vaccine. Since we observed a linearly increasing function of people getting the

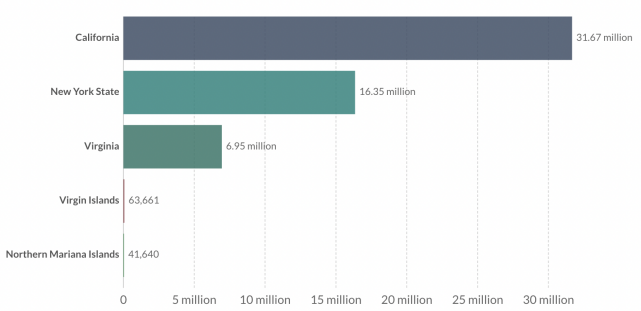

Figure 9: Total Vaccinations of 5 States till May 5th, 2021

vaccines in the first four months (Jan-Apr 2021), we observed more people getting the vaccines in the previous two months (March and April) instead of January and February. This can be seen in Figure 7. We also observed in Figure 11 that New York had peak cases at the end of December but soon after people start getting vaccines, the cases are constantly declining. A similar trend can also be seen in Virginia as shown in Figure 10. There is not much to observe in the Northern Mariana Islands and the Virgin Islands since these states were the least affected by COVID-19.

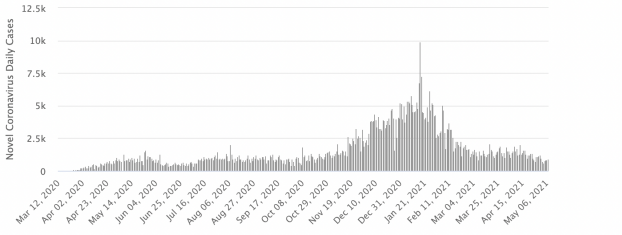

Figure 10: Virginia Daily Cases

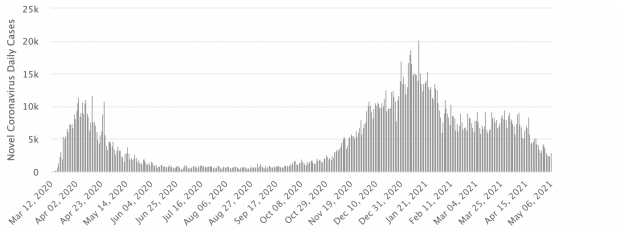

Figure 11: New York Daily Cases

6 DEEP LEARNING MODELS

There are many algorithms that we experiment with to see which provide more accurate results. Among the initial models, is to use the Prophet model which is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality. Then, we also bring ideas from the NLP community by trying to use LSTMs and tune the neural network to work best with our dataset, which is almost daily

updated and varies insanely due to the rapid mutation of COVID-19. In some cases, this option doesn't work well in capturing long-term temporal correlation, so we use the state-of-the-art models to process the sequential data, which is the transformers. Transformers are mostly used in NLP tasks such as text summarization and machine translation problems, but nowadays these models have been also applied to computer vision and time-series (real-valued) data. Building a time-series transformer may seem a challenging task since everything is real-valued data without any vocabulary or corpus. In this section, we discuss the three models we built for our COVID-19 forecasting task. In all of them, we used the Adam optimizer with a learning rate of 0.0001. Note that deep learning models are eager for data, so we decided to split our data into sliding window chunks with a stride of 1. In the cases and deaths dataset, we use a window size = 40, but in the vaccines dataset, we use a smaller window size that is 30 to increase the number of data points.

6.1 Long Short Term Memory (LSTM)

The first model we use in forecasting is LSTM, which is an enhanced version of RNN, that should be able to capture long-term temporal correlations in the data [7]. We may refer to RNNs or LSTMs alternatively to address the same meaning. We formulate our model as a seq2seq task in a setting that we are given a sequence of previous timesteps and we want to predict the next timesteps. The timesteps in our datasets are always the day of the year. The architecture of our LSTM model is shown in figure 12. Specifically, we preprocess our datasets to be in the format of [Batch Size, Number of Time Steps, Number of Features], then we have built three LSTMs in Sequence to serve as our encoder blocks.

Although our datasets range in their length because the vaccines have just started a few months ago, we fixed the hyperparameters across the datasets. The number of features depends on the dataset we use whether (cases + deaths) or (vaccines).

For training in all of our models, we use the MSE loss described below in the Evaluation Metrics. However, for comparing results across different models we decided to evaluate on the non-normalized data, therefore we chose MAPE and R2 score because our dataset has values that range from zero to millions, and MSE is not a good choice (gives big values) when data is not normalized.

The decoder part is considered to be only a one-layer neural network that maps back the hidden_size to the number of features. The output activation function can be Tanh or Sigmoid depending on whether data is normalized between -1 and +1, or zero and +1 respectively.

6.2 Gated Recurrent Unit (GRU)

Gated Recurrent Unit or GRU is another version of RNN cells that aims to solve the vanishing gradient problem, and it only consists of two gates [4]. We use the same methodology and identical hyperparameter values described in the previous subsection to evaluate our datasets using GRUs.

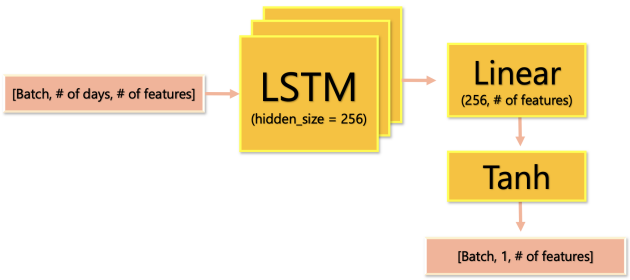

Figure 12: LSTM Model Architecture

6.3 Transformer Models

6.3.1 *Background - Vanilla Transformer.*

The original transformer architecture is proposed in the paper "Attention is all you need" [19]. The vanilla transformer is designed as a seq2seq problem for neural machine translation (NMT) tasks in the Natural Language Processing (NLP) domain. Transformers have proved to be the SOTA model for seq2seq problems, and recently it has been applied to different domains such as the computer vision field. Transformers are better than RNNs. They totally avoid recursion and can be parallelized (training is faster, and they can learn long-term temporal correlation relationships between words for very long sentences. Transformer models apply a lot of tricks: Positional Encoding or Embedding, Self-Attention, Multi-Head Attention, Masked Attention, Residual Connections, Layer Norm, Positional Feed-Forward Network, No RNN cells at all [19].

The transformer consists of Encoder and Decoder Blocks. In a typical NMT task from English to French, The first encoder block receives the English sentence, and the first decoder block receives the French sentence. But in Natural Language Generation task, we only have the English sentence which is our sample, and we want the model to learn how to reproduce it again and forecast the next timesteps. That is the targets are just the inputs shifted to the left. The challenge is applying transformer models on real-valued (time-series) data.

6.3.2 *Time-Series Transformers (TST).*

The original transformer described above cannot be applied directly to real-valued data which is the dominant type of our datasets. We modify that vanilla transformer and enhance it to generalize it to time-series tasks. We only use the encoder blocks of the transformer, and we ignore the decoder blocks at all.

The encoder blocks of the original transformer consist of Self Attention and Positional Feed-Forward Network, but we further improve the encoder blocks and use an architecture similar to GPT2 [15]. Our encoder blocks use the Masked-Self Attention option to prevent each time step from attending to future samples, and only pays attention to the previous ones.

The input embedding layer is replaced by a single-layer neural network that maps the data dimensions from the number of features to the hidden dimension that is expected by the encoder blocks. We train the confirmed cases and deaths together, so in this case, we have two features, but in vaccinations, it is a univariate time series dataset, so the number of features is one. As transformers don't use

any RNN cells, it needs a way to identify the positions of timesteps, as a result, we use the original positional encoding [19] although there are other options such as time2vec and positional embedding, positional encoding provides acceptable results. Softmax activation of the output layer is replaced by a Sigmoid or Tanh activation function based on the data normalization range. The softmax layer is removed because we want to produce real-valued numbers, not probabilities as we don't have vocabulary or corpus in our case. The overall architecture is shown in figure 13.

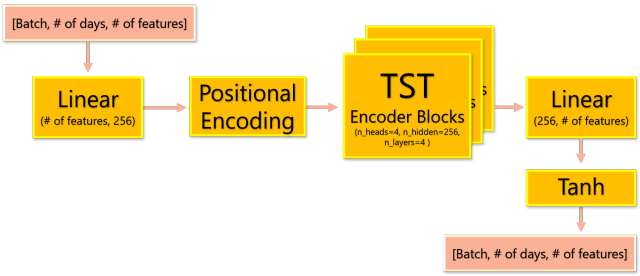

Figure 13: Our Time Series Transformers Model Architecture

7 BASELINE MODELS

7.1 Facebook Prophet Model

Prophet is Facebook's modular regression-based model. Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data and shifts in the trend, and typically handles outliers well. It provides automated forecasts which can be tuned as well. We can use human-interpretable parameters to improve our forecast by adding our domain knowledge. This was also designed keeping in mind the person whose experience with forecasting is very less. This is one of the reasons that this model serves as a good place for a baseline model. We can also see a seasonal trend in COVID-19 peaks if we observe the yearly data. Prophet handles seasonal trends well. It also can handle the multiplicative trends, which also suits well for COVID-19 peaks and dips. Due to these reasons, prophet serves as a good baseline to test deep learning models. Following are a few of the many hyperparameters we leveraged for our models in the experiments. We will explain shortly which parameters gave better results to us.

- Weekly seasonality: Can be 'auto', True, False. We usually made it True as it gave better results.
- Yearly seasonality: Can be 'auto', True, False. We usually made it True as it gave better results.
- Daily seasonality: Can be 'auto', True, False. We usually did not make it True as False or auto gave better results.
- Seasonality mode: 'additive' (default) or 'multiplicative'. Additive suited best for us since whenever there is an increase or decrease, it is more additive. Additive also gives a smoother model, which led to low values of MSE and MAPE values.
- Changepoint range: Proportion of history in which trend change points will be estimated. We tried various values between 0 and 1 to obtain the optimal value.

7.2 SARIMA Model

Seasonal Autoregressive Integrated Moving Average (SARIMA) model is an extension of the ARIMA model which also incorporates the seasonality for univariate time series forecasting. It has both components: autoregressive and moving average. The integrated component indicates differencing, which handles trends. We did not use ARIMA since we observed some seasonality in our data, therefore we used the seasonal version of the ARIMA model (SARIMA). ARIMA had three hyperparameters due to its three components of autoregression, differencing, and moving average. SARIMA has four new seasonal parameters together with the parameters of ARIMA. One of them defines the period of seasonality. We define SARIMA with these trend parameters (p,d,q) and seasonal parameters (P, D, Q)m:

- p: Trend autoregression order.
- d: Trend difference order.
- q: Trend moving average order.
- P: Seasonal autoregressive order.
- D: Seasonal difference order.
- Q: Seasonal moving average order.
- m: The number of time steps for a single seasonal period.

We did extensive fine-tuning of these parameters to obtain the optimal parameters. For m, three weeks, two weeks, etc. come to be optimal.

8 PIPELINE

We follow a set of defined steps in training and testing our deep learning models:

- Converting datasets from daily basis into cumulative to speed the training and make it easier.
- Normalizing the values using Min-Max Scaler Normalization (-1.0 to +1.0), because our values can range from zero to millions, which makes it impossible to train the neural networks and may run into exploding gradient issues.
- Dividing the data into sliding windows chunks each of size 40 or 30 depending on the dataset in hand, with a stride of 1 (one day).
- Splitting the dataset into training and testing datasets.
- Feeding the preprocessed data into the neural network model (LSTM, GRU, or TST)
- Training the neural network using Adam optimizer of a learning rate = 0.0001 and an MSE loss. We fix the number of training epochs to be 100.
- Testing the results and forecasting the future of the next days.
- Inverting the normalization and converting cumulative values into a daily basis.

9 EVALUATION

9.1 Evaluation Metrics

There are several evaluation metrics available for comparing different time series models. We are using Mean absolute percentage error (MAPE) and Coefficient of Determination (R2 Score) to be the most effective in our case. We preferred MAPE over Mean Squared Error (MSE) or Root Mean Squared Error (RMSE) or Mean Absolute Error (MAE) because it is the normalized version and more representative of the relative difference. Following are the definitions of all of these metrics.

- **Mean absolute percentage error (MAPE)**: It is the normalized version of Mean Square Error Metric (MSE). It is the sum of the normalized absolute differences between actual and forecasted values divided by total values. It is normalized by actual values. It is also used as a loss function for regression problems and in model evaluation, because of its very intuitive interpretation in terms of relative error. The lower MAPE indicates the better model. We show its value to be between 0 and 1, with 0 MAPE value being the ideal model. Let A_t be actual values, F_t be forecasted values, and n be the number of values. MAPE is given by the following equation:

$$\text{MAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right|$$

- **Coefficient of Determination (R2 Score)**: It is one minus (the sum of squares of residuals divided by the total sum of squares). It provides a measure of how well-observed outcomes are replicated by the model, based on the proportion of total variation of outcomes explained by the model. This is usually called R-squared. The higher the better. The R2 value of 1 is the representation of perfect fit. Let SS_{res} be the sum of squares of residuals and SS_{tot} be the total sum of squares.

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

- **Mean Squared Error (MSE)**: It is the sum of the squared differences between actual and forecasted values divided by total values. The lower MSE indicates the better model. Let A_t be actual values, F_t be forecasted values, and n be the number of values. MSE is given by the following equation:

$$\text{MSE} = \frac{1}{n} \sum_{t=1}^{n} (A_t - F_t)^2$$

9.2 Test & Forecast Functions

We build two different methods with different implementations for the testing and forecasting tasks. In testing, we use the split window chunks and feed one by one to the model, then we compare the forecasted timesteps of each chunk and test the results against the real values, and calculate the r2 score and mean absolute percentage error. However, We use different steps in forecasting the future days in tasks as predicting when all people of a particular state will be vaccinated or when the COVID-19 will end. For forecasting, we decided to stuck with the time-series transformers since it provides the best results as shown in the next section. We train the transformer on all days without any split into train and test datasets, so it can capture all patterns in the past few months until today. Then for predicting, we feed the transformer the last "window_size" days and start forecasting the next days in an auto-regressive way. Because we don't have the real values, so we depend on the last predicted day (timestep) and append to the data, and forecast the next day using the new data point. Note that we have already provided the weights for the trained models, so you can access our model by loading them directly and start forecasting.

10 EXPERIMENTS & RESULTS

10.1 Models Comparison

In the next subsections, we compare three deep learning models along with two baselines. As we will show, the transformer models outperform the other models. However, in some cases where there's a limited amount of data, we may find the LSTM to work better since the transformer is eager for large datasets. We compare our results against five states, and evaluate the last 20% or 30% days using MAPE and R2 score metrics. We also provide some graphs of how each model fits the datasets differently. We note that it's easy for training to converge and train fast in large population states such as New York and California since there's an existing temporal correlation across the days. However, for states such as the Northern Mariana Islands and the Virgin Islands, all of the models don't seem to capture the trend very well. We compared against the baselines for 2 states only, then we choose the DL models to forecast the results for the remaining states.

10.1.1 ***Cases and Deaths data for 2 states.*** Figure 14 shows the comparison of five models for cases and deaths of New York and Virginia. Transformers clearly show lower MAPE scores than LSTM and GRU. The lower the MAPE score the better is the model.

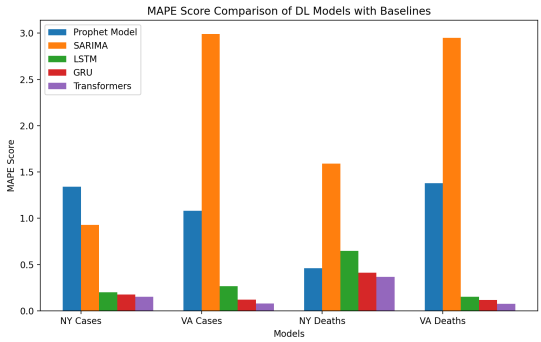

Figure 14: Comparison of MAPE score for NY Cases, VA Cases, NY Deaths and VA Deaths

Figure 15 shows the comparison of five models for cases and deaths of New York and Virginia. Transformers clearly show the R2 Score to be greater than LSTM and GRU. Baselines gave a negative R2 score. The higher the R2 score the better is the model.

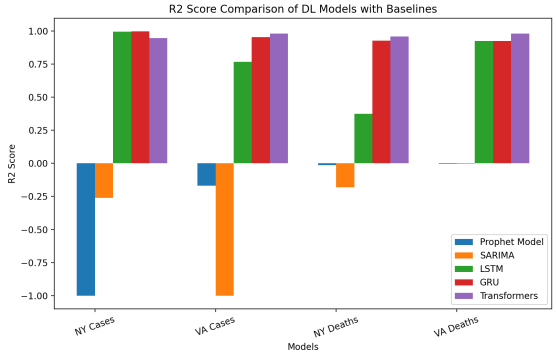

Figure 15: Comparison of R2 score for NY Cases, VA Cases, NY Deaths and VA Deaths

10.1.2 ***Vaccines data for 2 states.*** Figure 16 shows the comparison of five models for vaccines of New York (NY) and Virginia (VA). Transformers clearly show an R2 Score greater than LSTM. Baselines gave a negative R2 score. The higher the R2 score the better is the model.

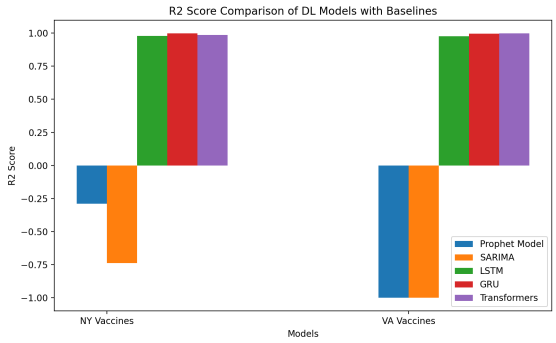

Figure 16: Comparison of R2 score for NY Vaccines and VA Vaccines

Figure 17 shows the comparison of five models for vaccines of New York (NY) and Virginia (VA). Prophet model shows better results here since the lower the MAPE score the better is the model. But if we combine these results with the R2 Score in Figure 16, Transformers is the clear option for us since it has a comparable MAPE score and high R2 score. LSTM in this case did not work well. You can get access to the notebooks to find more details about our numbers and for other states as well.

10.2 Forecasting the Number of Confirmed Cases

Figure 18 shows the forecasting using three of the deep learning algorithms as compared to the real test dataset. Y-axis in this graph shows the cumulative cases in Virginia. The X-axis shows the last two months: March and April 2021. Figure 19 also compares three algorithms by forecasting on New York's test dataset date ranges. Y-axis shows the daily cases of New York. Figure 20 also compares three algorithms by forecasting on California's test dataset date ranges. Y-axis shows the daily cases of California. Transformers is the best option here as it also tried to capture the peaks, which is crucial in COVID-19 forecasting.

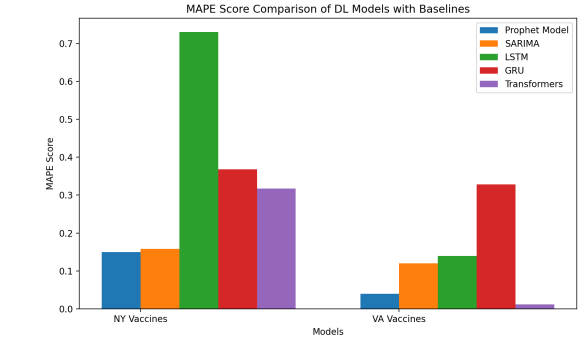

Figure 17: Comparison of MAPE score for NY Vaccines and VA Vaccines

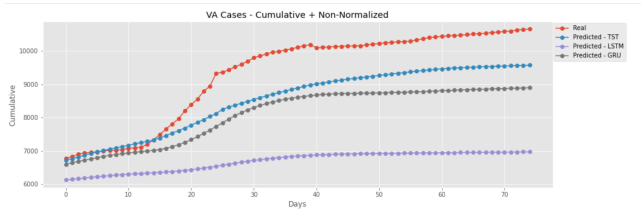

Figure 18: Forecasting of Cumulative Cases of Virginia

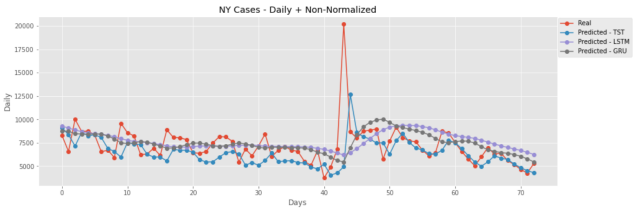

Figure 19: Forecasting of Daily Cases of Virginia

10.3 Forecasting the Number of Deaths

Figure 21 shows the forecasting using three of the deep learning algorithms as compared to the real test dataset of deaths. Y-axis in this graph shows the daily deaths in California. The X-axis shows the last two months: March and April 2021. Figure 22 also compares three algorithms by forecasting on New York's test dataset date ranges. Y-axis shows the daily deaths of New York. Figure 23 compares three algorithms by forecasting on California's test dataset date ranges. Y-axis shows the cumulative deaths of Virginia. Figure

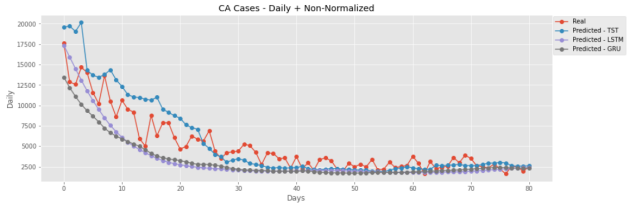

Figure 20: Forecasting of Daily Cases of California

24 is the forecasting of normalized cumulative deaths. Transformers clearly is the best option here as it also tried to capture the peaks, which is crucial in COVID-19 forecasting.

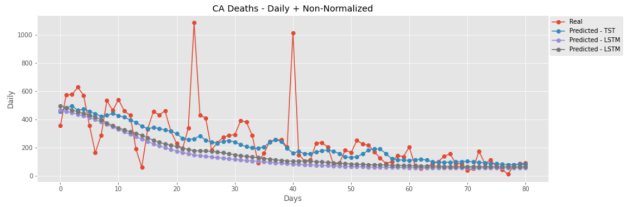

Figure 21: Forecasting of Daily Deaths of Virginia

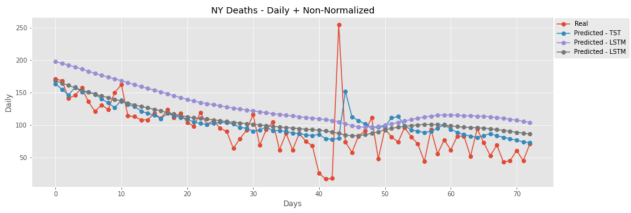

Figure 22: Forecasting of Daily Deaths of New York

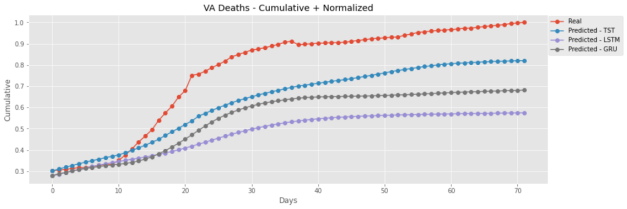

Figure 23: Forecasting of Cumulative Deaths of Virginia

10.4 Forecasting the Number of Administrated Vaccine Doses

Figure 26 shows the forecasting using three of the deep learning algorithms as compared to the real test dataset of vaccines. Y-axis in this graph shows the cumulative vaccines in New York. The X-axis shows the last 25 days. Figure 25 also compares three algorithms by forecasting on New York's test dataset date ranges. Y-axis shows

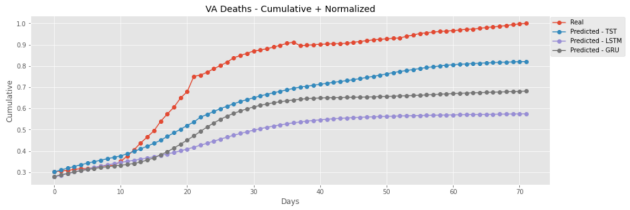

Figure 24: Forecasting of Normalized Cumulative Deaths of Virginia

the normalized cumulative vaccines of New York. Figure 28 compares three algorithms by forecasting on Virginia's test dataset date ranges. Y-axis shows the cumulative vaccines of Virginia. Figure 25 is the forecasting of normalized cumulative vaccines. Transformers and GRU both are the best options as they are closer to the real test dataset.

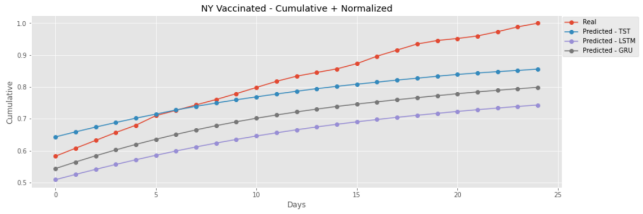

Figure 25: Forecasting of Normalized Cumulative Vaccines of New York

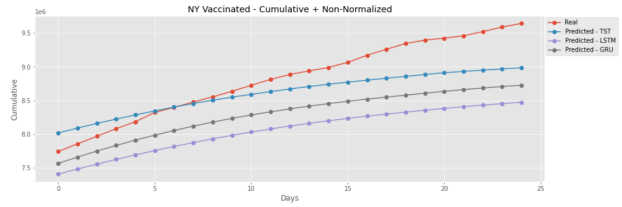

Figure 26: Forecasting of Cumulative Vaccines of New York

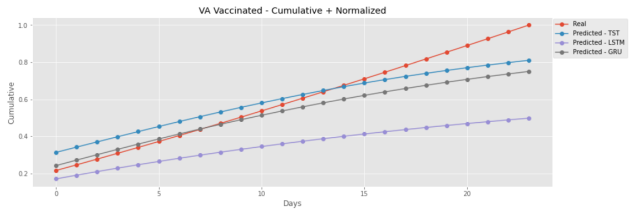

Figure 27: Forecasting of Normalized Cumulative Vaccines of Vaccine

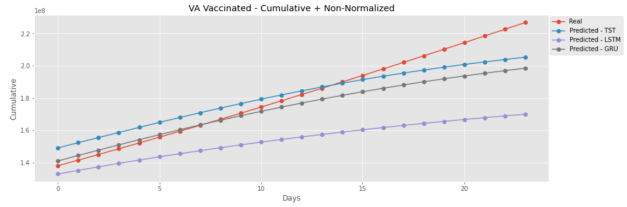

Figure 28: Forecasting of Cumulative Vaccines of Virginia

10.5 When a State is Fully Vaccinated?

First, we get the population per state using the Infoplease site [10]. To answer this interesting question, we use the forecasting function explained in the Evaluation section. In New York, the population is 19,453,561, and the number of vaccinated people until May 5th is 9,642,901. We asked the model to forecast the next 100 days, and it shows that during these 100 days there will be a total of 9,855,832 people. By adding the two numbers of vaccinated people we get a total of 19,498,733 which is even more than the population size of New York. Therefore, our analysis predicts that in less than 100 days, all people in New York will be vaccinated. The predicted number of vaccinated people per day in New York is shown in figure 29.
We did the same computation for Virginia. The number of currently vaccinated people is 4,194,146 out of 8,535,519. Our forecast shows that in 115 days, we will hit a total of 8,644,000 vaccinated people in Virginia, which is even higher than the Virginia population. The predicted number of vaccinated people per day in Virginia is shown in figure 30. You can find more of these results on our GitHub [6]. Our model is generalizable to work on any state, just by providing its name.

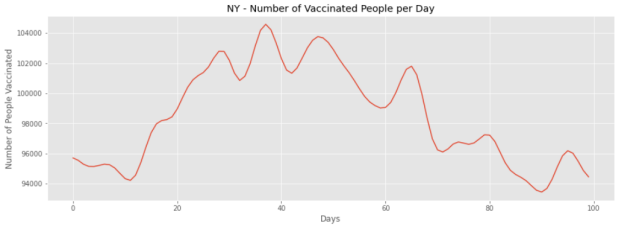

Figure 29: Predicted number of vaccinated people per day in New York

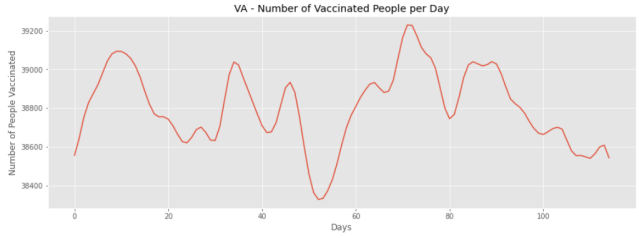

Figure 30: Predicted number of vaccinated people per day in Virginia

11 FUTURE WORK

The currently implemented transformer uses the original encoder blocks from the original "attention is all you need" paper [19]. Although it works, there are many advances to this attention layer, that improves further the implementation and makes it more faster and memory efficient. For example, Informer model [21] that use lower computing resources, train faster, and solve memory issues for long sequence time-series forecasting. Also, we didn't have time to do some experiments on using RNN units (LSTM, or GRU) with the attention layer, as it may decrease the number of the parameters while providing acceptable results. Note that our current transformer model uses a lot of memory in its computations. We also want to experiment with training one transformer model on all data points combined from all states together using the idea of controlled generation by appending the name of the state at the beginning of each data point, so that the model can learn to attend this state name with the measurements of this state.

12 CONCLUSION

In this project, we studied and analyzed the COVID-19 data for the United States per state. We formulated our problem as a forecasting task to predict the future number of confirmed cases, deaths, and vaccinated people. We discussed many sequential deep learning models and compared them to some baseline models that are used frequently in time-series forecasting. We built a new model called the time-series transformer and proved that it gives better results than the RNN (LSTM and GRU) models. We also forecasted when all people of a particular state will be vaccinated. We are giving our code at this GitHub repository [6].

REFERENCES

[1] Devante Ayris, Kye Horbury, Blake Williams, Mitchell Blackney, Celine Shi Hui See, and Syed Afaq Ali Shah. 2020. Deep Learning Models for Early Detection and Prediction of the spread of Novel Coronavirus (COVID-19). *arXiv preprint arXiv:2008.01170* (2020).

[2] C Bui, N Pham, A Vo, A Tran, A Nguyen, and T Le. 2017. Time series forecasting for healthcare diagnosis and prognostics with the focus on cardiovascular diseases. In *International Conference on the Development of Biomedical Engineering in Vietnam*. Springer, 809–818.

[3] Vinay Kumar Reddy Chimmula and Lei Zhang. 2020. Time series forecasting of COVID-19 transmission in Canada using LSTM networks. *Chaos, Solitons Fractals* 135 (2020), 109864. https://doi.org/10.1016/j.chaos.2020.109864

[4] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. arXiv:1412.3555 [cs.NE]

[5] Denise Bandeira da Silva, Diogo Schmidt, Cristiano André da Costa, Rodrigo da Rosa Righi, and Björn Eskofier. 2021. DeepSigns: A predictive model based on Deep Learning for the early detection of patient health deterioration. *Expert Systems with Applications* 165 (2021), 113905.

[6] HaiderAli and Yusuf Elnady. 2021. Forecasting COVID-19 Trends in 2021: Comparing RNNs and Time-Series Transformers. https://github.com/haider4445/Covid19ForecastUsingDL

[7] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[8] Michael J Horry, Subrata Chakraborty, Manoranjan Paul, Anwaar Ulhaq, Biswajeet Pradhan, Manas Saha, and Nagesh Shukla. 2020. COVID-19 detection through transfer learning using multimodal imaging data. *IEEE Access* 8 (2020), 149808–149824.

[9] Zixin Hu, Qiyang Ge, Shudi Li, Li Jin, and Momiao Xiong. 2020. Artificial intelligence forecasting of covid-19 in china. *arXiv preprint arXiv:2002.07112* (2020).

[10] Infoplease. [n.d.]. State Population by Rank. https://www.infoplease.com/us/states/state-population-by-rank

[11] Yixiu Liu, Yujuan Huang, Jianyi Wang, Li Liu, and Jiajia Luo. 2018. Detecting premature ventricular contraction in children with deep learning. *Journal of*

Shanghai Jiaotong University (Science) 23, 1 (2018), 66–73.

[12] Edouard Mathieu. 2021. State-by-state data on COVID-19 vaccinations in the United States. https://ourworldindata.org/us-states-vaccinations

[13] Nytimes. 2021. Coronavirus (Covid-19) Data in the United States. https://github.com/nytimes/covid-19-data

[14] Tulin Ozturk, Muhammed Talo, Eylul Azra Yildirim, Ulas Baran Baloglu, Ozal Yildirim, and U Rajendra Acharya. 2020. Automated detection of COVID-19 cases using deep neural networks with X-ray images. *Computers in biology and medicine* 121 (2020), 103792.

[15] A. Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

[16] Furqan Rustam, Aijaz Ahmad Reshi, Arif Mehmood, Saleem Ullah, Byung-Won On, Waqar Aslam, and Gyu Sang Choi. 2020. COVID-19 future forecasting using supervised machine learning models. *IEEE access* 8 (2020), 101489–101499.

[17] Srk. 2021. Novel Corona Virus 2019 Dataset. https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset

[18] José F Torres, Dalil Hadjout, Abderrazak Sebaa, Francisco Martinez-Alvarez, and Alicia Troncoso. 2021. Deep Learning for Time Series Forecasting: A Survey. *Big Data* 9, 1 (2021), 3–21.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017).

[20] Abdelhafid Zeroual, Fouzi Harrou, Abdelkader Dairi, and Ying Sun. 2020. Deep learning methods for forecasting COVID-19 time-Series data: A Comparative study. *Chaos, Solitons Fractals* 140 (2020), 110121. https://doi.org/10.1016/j.chaos.2020.110121

[21] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. arXiv:2012.07436 [cs.LG]