CUNY DATA 698 Spring 2021. Capstone: Senior Research Project
Authors: Santosh Cheruku, Angel Claudio, John K. Hancock, John Suh, and Subhalaxmi Rout

A Comparison of COVID-19 Time Series Models Forecasting Cases, Hospitalizations, and Deaths for New York

Santosh Cheruku
Angel Claudio
John K. Hancock
John Suh
Subhalaxmi Rout

CUNY DATA 698 Spring 2021. Capstone: Senior Research Project
Authors: Santosh Cheruku, Angel Claudio, John K. Hancock, John Suh, and Subhalaxmi Rout

TABLE OF CONTENTS

Section	**Page No.**
ABSTRACT	1
INTRODUCTION	1
LITERATURE REVIEW	2
DATA AND METHODS	6
COMPARATIVE RESULTS	20
DISCUSSION	21
CONCLUSION	22

CUNY DATA 698 Spring 2021. Capstone: Senior Research Project

Authors: Santosh Cheruku, Angel Claudio, John K. Hancock, John Suh, and Subhalaxmi Rout

ABSTRACT

The earliest case of Severe Acute Respiratory Syndrome Coronavirus 2, later to be identified as COVID-19, was discovered in December 2019. The virus, which is spread through aerosol transmission of human respiratory droplets, would be discovered in the U.S. three months later. U.S. public health care officials were unprepared for the exponential rise in cases, hospitalizations, and deaths. The severity of the crisis could have been mitigated with better data collection and forecasting methods. Looking at data from the first three months in New York, this paper compared the efficacy of nine different time series models to see which one was the most accurate at forecasting cases, hospitalizations, and deaths. Our analysis showed that the Seasonal Auto Regressive Integrated Moving Average("SARIMA") model showed the most consistency and accuracy for predictions of cases, hospitalizations, and deaths. We were able to show the efficaciousness of time series models to predict future needs during a pandemic. We recommend further study into using time series models to forecast pandemics. Specifically, we recommend developing a time series models just for pandemics.

INTRODUCTION

In December 2019, the World Health Organization ("WHO") reported the earliest onset of symptoms of a new highly contagious disease in Wuhan, China. By January 2020, the WHO confirmed human to human transmission of this new virus. This disease which is caused by the virus Severe Acute Respiratory Syndrome Coronavirus 2 or SARS-CoV-2 became known globally as COVID-19. The virus is spread through aerosol transmission of human respiratory droplets.

By March of 2020 the WHO declared the virus a pandemic. The United States became the epicenter of the pandemic by the end of March 2020 with New York reporting the highest number of cases in the world with over 12,000 cases reported in early April 2020. The exponential rise of cases created a strain on hospital resources, and the number of deaths also grew at an alarming rate.

The severity and rapid transmission of COVID-19 caught government officials off guard. There were shortages of Personal Protection Equipment ("PPE") for frontline health care workers, hospitals were over run, and shipping containers had to be used for the over flow of the deceased. The crisis could have been mitigated with better data collection and time series models that would have informed government officials of the potential impact.

The objective of this project is to compare multiple time series models, from the naïve to the more advanced, to determine which model can accurately predict cases, hospitalizations, and deaths from this pandemic. The best model(s) could be used by public health officials to prepare for future pandemics.

Can a time series model created from data taken from the early months of the pandemic provide accurate forecasts for a future period in 2020? As an example, could time series data taken from March through May of 2020 predict cases, hospitalizations, and deaths in June 2020?

LITERATURE REVIEW

Our research found an abundance of articles that used time series models to forecast the spread of covid-19. We saw that from the start, in March 2020, when the World Health Organization ("WHO") declared COVID-19 a global pandemic, researchers deployed an array of time series models to forecast the spread of the virus. These efforts were largely done to inform the general public of the need to enact mitigation measures to stop the spread. The articles discussed in this review provide an exemplar of the articles that we researched. We begin our review by comparing the findings from very basic time series models and finishing with a look at the more advanced ones.

Authors Haytham H. Elmousalami and Aboul Ella Hassanien's research focused on the day level spread of the virus [1]. Their methodology used naive time series models, Moving Average ("MA"), Weighted Moving Average ("WMA"), which take averages or weighted averages of past observations to forecast future cases. Such models are easy to create and even easier to report to the public. Most media outlets include a moving average component when reporting about the spread. The other model that the authors used was Single Exponential Smoothing ("SES"), another naïve model. In this instance, instead of weighing past observations equally, SES uses functions to exponentially decrease the weighting of past observations.
The results of Elmousalami and Hassanien's models showed that the SES model had the highest accuracy for confirmed cases, recovered cases, and deaths based on the evaluations of Mean Absolute Deviation ("MAD"), Mean Square Error ("MSE), Root Mean Square Error ("RMSE"), and Mean Absolute Percentage Error ("MAPE"). In contrast to Elmousalami and Hassanien, authors Vasilis Papastefanopoulos, Pantelis Linardatos and Sotiris Kotsiantis used a higher class of time series models [2], Auto Regressive Integrated Moving Average ("ARIMA"), Holt-Winters additive model ("HWAM"), Trigonometric seasonal formulation Box-Cox transformation ARIMA errors and trend component ("TBAT"), Facebook's Prophet, Deep AR, a probabilistic forecasting with Auto-Regressive Recurrent Networks, and N-Beats, a neural basis expansion analysis for interpretable time series forecasting.

Papastefanopoulos, Linardatos, and Kotsiantis' models consisted of linear regression and deep learning neural networks. Instead of applying averages or weights or exponentially decreasing weights, these models primarily make predictions by using either a regression of past observations or by using a system of inter-connected nodes that learns from past observations.

Also, in contrast to Elmousalami and Hassanien, the authors Papastefanopoulos, Linardatos, and Kotsiantis did not find a "one-size-fits-all" model. Their findings showed that based, on RMSE measures, the ARIMA and TBAT models performed best in most of the countries while achieving second best in the other two. They found "traditional statistical methods such as such ARIMA and TBAT overall prevail over deep learning counterparts such as DeepAR, and N-BEATS—an outcome which, due to the lack of large amounts of data [2]." (emphasis added)

Authors Vinay Kumar, Reddy Chimmula, and Lei Zhang focused solely on using a single deep learning network, Long short term Memory ("LSTM") [3], a non-linear approach that uses a Recurrent Neural Network ("RNN") to forecast trends. In an RNN, output from the last step is fed as input to the current step. This is somewhat similar to Elmousalami and Hassanien's approach where the weights applied to past observations are manipulated to make forecasts. The difference is that in LSTM networks can retain long term information which is useful if there are lags of unknown duration between important time gaps.

Using data collected in Canada until March 31, 2020, Kumar, Chimmula, and Zhang's methodology was to use sequential networks to extract the patterns from a time series dataset. The rationale for this approach was that the linear approach often neglects the temporal components in the data. "They depend upon regression without non- linear functions and failed to capture the dynamics of transmission of infectious diseases like novel corona virus. Statistical models such as Auto Regressive Integrated Moving Average (ARIMA), Moving Average (MA), Auto Regressive (AR) methods overwhelmingly depends on assumptions and such models are difficult for forecasting real-time transmission rates." [3] In contrast to Papastefanopoulos, Linardatos, and Kotsiantis, they showed that the RMSE of the LSTM had the highest accuracy.

Authors Abdelhafid Zeroual et al. [4] differed from the previously discussed studies by comparing the five most advanced models, Recurrent Neural Network ("RNN"), Long short-term memory ("LSTM"), Bi-directional LSTM ("Bi-LSTM"), Gated recurrent units ("GRUs") and Variational AutoEncoder ("VAE") to forecast cases and recovered cases across six countries. The authors cited the models' ability to "handling temporal dependencies in time series data, distribution-free learning models, and their flexibility in modeling nonlinear features." [4]

Using RMSE as their primary performance metric, the authors found that VAE outperformed the other models for confirmed and recovered cases. This study was one of the first times that VAE has been used to model COVID-19 cases. The authors offer a reason as to why VAE out-performed the other advanced models. "[T]he capacity of the VAE in dealing with small data compared to the other recurrent models (RNN, L STM, Bi-L STM, and GRU) which may need more lengthy data to extract relevant variability in time series data [4]." (emphasis added) Poor performance of advanced models due to the lack of data was the same issue that Papastefanopoulos, Linardatos, and Kotsiantise experienced with the performance of DeepAR and N-BEATS. However, we did not see this issue with Kumar, Chimmula, and Zhang's LSTM model.

The articles discussed and compared in our review are indicative of the articles that we researched for the project. Each one chose a particular class of models to compare, used various datasets, compared the results using standard metrics, and summarized their findings. What we did not find was one paper that compared all classes of models, from the naive to the more advanced, in one study over one single dataset over the same period of time in order to determine which model(s) performed the best. That is the objective of this paper.

DATA AND METHODS

The research problem that this project attempts to solve is to find the most efficacious time series model that could have provided health officials with the most accurate predictions of the number of COVID-19 cases, hospitalizations, and deaths in the early days of the pandemic. To make this determination, we will compare the results of multiple time series models ranging from the very basic to the most advanced.

To start, we reviewed data collected by authoritative sources, the Centers for Disease Control ("CDC") and Johns Hopkins University, for the time period from March to June 2020. After review of these and other sources, we made a selection of the best data source for analysis. We explored and evaluated the data making notes of key points.

We proceeded to build nine time series models of various classes, from the most basic to the more advanced, Seven-day Rolling Average, Simple Exponential Smoothing, Holt Winters, ARIMA, SARIMA, Facebook Prophet, XGBOOST, Neural Networks, and Long Short Term Memory ("LSTM"). Each model will be written in Python, and will be evaluated using standard metrics, Root Mean Square Error ("RMSE"), Mean Square Error ("MSE"), Mean Absolute Error ("MAE"), and Mean Absolute Percentage Error ("MAPE"). The performance of each model on the same dataset over the same time period and the same geographical location provided a solution to our research problem.

Data Sources

Our team investigated three data sets as candidates for modeling future effects of COVID-19 in areas resembling the state of New York. The first of the three was the Centers for Disease Control and Prevention (CDC) [15]. The CDC data, while extensive, did not have any attributes for location, which was vital for us to create a model for our main area of interest (New York) and then trying to contrast it with a similar location (e.g., Florida, Texas, California).

The next data source we vetted was the Johns Hopkins University Center for Systems Science and Engineering [16]. The data set was a strong contender for modeling, but we found the volume of the data (424 features) did not come without the need for a lot of data manipulation. This included filtering out hundreds of unnecessary columns, a pivot execution in both directions for values and columns needing restructure, multiple data cleansing tasks including imputations, and finally we would still be left with a heavy data source for extracting and using for computations.

This finally left us with our selected data set, The COVID Tracking Project at the Atlantic [17]. The data was readily available for us to extract via a web API and since it is a compilation of data brought together by the Atlantic team of data scientists and analysts, the data for the most part was clean and manageable. The COVID Tracking Project at the Atlantic supplies a public web API for data consumption. We were interested in region specific data sets and fortunately had the option to supply the query string of the API URL with any choice name of a US state. Unfortunately, the hosting of this resource had a schedule to be taken down on May 1, 2021. The data itself had a manageable structure and clean data. There were many features excluded as a result of not being applicable such as "date modified" or having one value for every observation - for example, a variable with exclusively "NA" values. (*See Appendix A, Data Sources and Collection.*)

Exploratory Data Analysis

The first case of COVID-19 in the state of New York was discovered on March 2, 2020 and by March 19, 2020, cases surged to over 7,000. This sudden and exponential rise in cases necessitated public health mitigation orders. On March 20, 2020, Governor Andrew Cuomo issued a stay at home order for the state [18], and on April 15, 2020, Governor Cuomo issued a mandate requiring the wearing of masks in public [19]. (*See Appendix B, Section II Exploratory Data Analysis.*) The mitigation measures worked. Thirty days after the lockdown and thirty days after the mask mandate, the number of cases, hospitalizations, and deaths all trended downward. These measures were the most effective action taken to fight the virus until a vaccine could be developed. This also shows the importance of gathering data and predicting the spread of the virus.

The final note in our exploration was that the state ceased reporting daily increases in hospitalizations effective June 2, 2020. This had an impact on some of our metrics.

Basic Models

The first class of time series models that we examined are foundational time series models which are often used as benchmarks for other time series models. These include Seven-day Rolling Average, Simple Exponential Smoothing, and Holt Winters.

Model One: Seven day Rolling Average

The most basic time series forecasting method is the rolling average method where the forecast is an average of all previous observations. The forecast is considered a flat forecast. The equation for rolling average is as follows:

$$MA_t = \frac{x_t + x_{t-1} + x_{t-2} + x_{t-3} + x_{M-(t-1)}}{M}$$

This final 7 day rolling average was used to make forecasts for the next 30 days. The 7-day rolling average, RMSE, MSE, MAPE, and MAE are reported in Table 1 below.

Table 1: Seven Day Rolling Average

Metric	Cases	Hospitalizations	Deaths
7-day Rolling Avg.	1322.143	180.857	73.429
RMSE	600.771	172.286	44.852
MSE	360,926.03	29,682.463	2,011.679
MAPE	86.70	Undefined	243.06
MAE	566.47	167.36	41.76

See Appendix B, III. Naive Models, Model 1: Seven Day Rolling Average predictions.

Model Two: Simple Exponential Smoothing

Single Exponential Smoothing, SES , also known as Simple Exponential Smoothing, is a time series forecasting method for data without a trend or seasonality. It requires a single parameter, called alpha, also called the smoothing factor. This parameter controls the rate at which the influence of the observations at prior time steps decay exponentially. Alpha is often set to a value between 0 and 1. Large values mean that the model is influenced mostly by the most recent past observations, whereas smaller values mean more of the history is considered when making a prediction. Forecasts are calculated using exponentially decreasing weighted averages of past observations. The smallest weights are associated with the oldest observations.

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha)y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \cdots,$$

As with the rolling average, the forecast is the last observed value, but unlike rolling average more weights are assigned to the more recent value. We tested different levels of alpha including allowing the stats model package to determine the optimal alpha. The best results, including the best alpha are reported in Table 2.

Table 2: Simple Exponential Smoothing

Metric	Cases	Hospitalizations	Deaths
α	0.995	0.792	0.8
RMSE	408.32	182.356	31.82
MSE	166,464	33,253.689	1012.51
MAPE	0.58	Undefined	1.90
MAE	369.71	177.70	27.64

See Appendix B: III. Naive Models, Model 2: Simple Exponential Smoothing

Model Three: Holt Winters

The Holt-Winters seasonal method comprises the forecast equation and three smoothing equations — one for the level, one for the trend and one for the seasonal component. There are two variations to this method that differ in the nature of the seasonal component. The additive method is preferred when the seasonal variations are roughly constant through the series, while the multiplicative method is preferred when the seasonal variations are changing proportional to the level of the series. The equation for the component form for the additive method is:

$$\hat{y}_{t+h|t} = \ell_t + hb_t + s_{t+h-m(k+1)}$$

$$\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$$

$$b_t = \beta * (\ell_t - \ell_{t-1}) + (1 - \beta *)b_{t-1}$$

$$s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}$$

Table 3: Holt Winters

Metric	Cases	Hospitalizations	Deaths
Alpha Level	.2	.2	.2
RMSE	185.48	62.95	66.86
MSE	34,402.83	3,962.70	4,470.26
MAPE	0.2136	Undefined	3.5490
MAE	151.95	53.73	60.49

See Appendix B, III. Naive Models, Model 3: Holt Winters

Autocorrelation and Moving Average Time Series Models

The next class of time series models that we examined seek to capture the autocorrelations of a time series. In addition to the models below, we looked at Auto Regressive Model (“AR”), Moving Average Model (“MA”), and Auto Regressive Moving Average Model (“ARMA”). See Appendices C and D for more information about these models.

Model Four: Auto Regressive Integrated Moving Average (“ARIMA”)

The ARIMA model is a good choice for data that has a moving mean or in other words, where data is non-stationary [20]. We say it is integrated because we are predicting the differences in one time stamp to a previous one. We do this in hopes of identifying a constant, which will be used as a mean to make the data stationary. Once an appropriate time series model has been fit, it may be used to generate forecasts of future observations. The expression for ARIMA is as follows

$$\left(1-\sum_{i=1}^{p}\varphi iL^{i}\right)(1-L)^{d}X_{t}=\delta+\left(1-\sum_{i=1}^{q}\theta iL^{i}\right)\varepsilon_{t}$$ where $L$is the lag operator.

In ARIMA we consider the following:

$$ARIMA=(p,d,q)$$

- p: Trend autoregression order (this is the AR order)
- d: Trend difference order (this is the Integrated order)
- q: Trend moving average order (this is the MA order)

As in the stationarity, the ACF and PACF of an ARIMA process are determined by the AR and MA components, respectively.

Below are the final ARIMA equations created and the metrics for the model:

Feature	ARIMA(p,d,q)	Final Equations
Cases	ARIMA (2,1,2)	$y_t = 12.8608 + 1.2211y_{t-1} + -0.8223y_{t-2} + -1.3266\varepsilon_{t-1} + 0.7512\varepsilon_{t-2}$
Hospitalizations	ARIMA (2,1,0)	$y_t = 2.1224 + -0.1951y_{t-1} + -0.0938y_{t-2}$
Deaths	ARIMA (2,1,2)	$y_t = 0.3477 + 0.9478y_{t-1} + -0.1156y_{t-2} + -1.5163\epsilon_{t-1} + 0.67732\epsilon_{t-2}$

Table 4: ARIMA

Metric	Cases	Hospitalizations	Deaths
RMSE	765.70	42.37	33.74
MSE	586,296.49	1,795.22	1,138.39
MAPE	0.98	Undefined	0.9
MAE	740.96	15.18	29.53

See Appendix B, II. Trend and Seasonal Time Series Models, Model 4: ARIMA

Model Five: Seasonal Auto Regressive Integrated Moving Average model ("SARIMA")

The SARIMA model is a variation of the ARIMA model that considers a timeseries that has seasonality. In ARIMA we consider the following:

$$ARIMA = (p, d, q)$$

However, with SARIMA the updated expression becomes:

$$SARIMA = (p, d, q)(P, D, Q)m$$

In the equation above we add to the ARIMA model a seasonality component. The first 3 arguments of this component represent the same arguments from the first tuple - to recap these values are:

1. The number of auto-regressive lags
2. The number of time steps to take the difference over.
3. The number of moving average lagged errors.
4.

But to add seasonality to these values we have added “m” to our equation, this represents the length of time for the observed seasonality. For example, if we were to see a spike in our data over a 12-month period, we would set the value to 12 since each time step in our data is 1 month long. Without differencing operations, the equation for SARIMA can be expressed as:

$$(1)\Phi(B^S)\phi(B)(x_t - \mu) = \theta(B^S)\theta(B)w_t[1].$$

Table 5: SARIMA

Metric	Cases	Hospitalizations	Deaths
RMSE	188.46	32.30	12.77
MSE	35,517.17	1043.29	163.07
MAPE	0.223	Undefined	0.392
MAE	155.43	13.94	3.36

See Appendix B, II. Trend and Seasonal Time Series Models, Model 4: SARIMA

In addition to these models, we looked at other Trend and Seasonal Time Series Models. See Appendices C and D for more details about Auto Regressive (“AR”), Moving Average (“MA”), and Auto Regressive Moving Average (“ARMA”) models.

Advanced Models

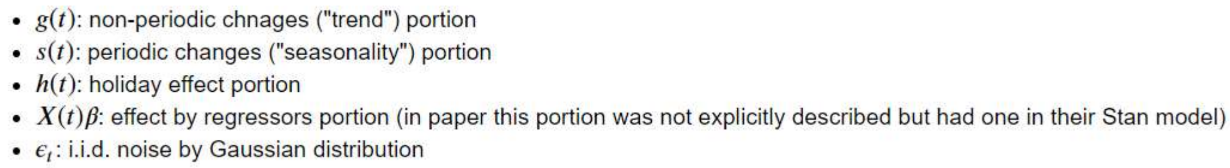
The next class of time series models that we evaluated are Facebook’s time series forecasting model, Prophet, and deep learning models, Neural Network (“NN”), XGBoost, and LSTM.

Model Six: Facebook's FBProphet Model

Developed by Facebook, FBProphet is an open-source library for univariate time series data forecasting. To help solve and alleviate this potential problem of not having enough analysts who could produce such highly complex and high quality forecasts without a deep knowledge and understanding of forecasting, Facebook created the FBProphet library for use in both Python and R. FBProphet implements what is referred to as an additive time series forecasting model, and the implementation supports trends, seasonality, and holidays. It tries to automatically create a time series data forecast based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. The math behind the model is as follows [21]:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

where :



- $g(t)$: non-periodic chnages ("trend") portion
- $s(t)$: periodic changes ("seasonality") portion
- $h(t)$: holiday effect portion
- $X(t)\beta$: effect by regressors portion (in paper this portion was not explicitly described but had one in their Stan model)
- ϵ_t: i.i.d. noise by Gaussian distribution

Respectively,

$$g(t) = (k + \mathbf{a}(t)^{\intercal}\boldsymbol{\delta})t + (m + \mathbf{a}(t)^{\intercal}\boldsymbol{\gamma})$$

Linear trend

$$g(t) = \frac{C(t)}{1 + \exp(-(k + \mathbf{a}(t)^{\intercal}\boldsymbol{\delta})(t - (m + \mathbf{a}(t)^{\intercal}\boldsymbol{\gamma})))}$$

Logistic trend

Trend portion

$$s(t) = \sum_{n=1}^{N} \left(a_n \cos\left(\frac{2\pi n t}{P}\right) + b_n \sin\left(\frac{2\pi n t}{P}\right) \right)$$

Seasonality portion

$$Z(t) = [\mathbf{1}(t \in D_1), \ldots, \mathbf{1}(t \in D_L)]$$

$$h(t) = Z(t)\boldsymbol{\kappa}.$$

Holiday effect portion

Facebook's Prophet model only works in Python version 3.7. Because of this, we had to separate this model out from Appendix B. For more information about this model, see Appendices E and F.

Table 6: Facebook Prophet Model

Metric	Cases	Hospitalizations	Deaths
RMSE	2939.30	823.50	129.54
MSE	8,639,484.49	678,152.25	16,780.61
MAPE	4.20	3.244	7.00
MAE	2,934.72	752.64	100.75

See Appendix E, The Prophet Model

The final class of models that we reviewed were deep learning neural networks, a system of inter-connected layers of nodes that takes input data and passes signals from one node to another and then from one layer to another through activation function. These signals are assigned weights that are adjusted as it learns. These networks are trained to produce a result through the output layer. Neural networks are designed to mimic how the human brain works.

Neural Network ("NN")

We first evaluated a feed forward neural network. Using Keras and a Sequential object, the structure consisted of the first layer being an input layer, followed by two hidden layers with 20 and 10 nodes and finally the output layer with only one node. Before we fit the model, we compiled it which we do using the mean absolute error loss function and the Adam optimizer with a learning rate of .001. After compilation, we fit the model on the training sets and validate on the test sets, and ran the network for 240 epochs. The model was trained on data from March to May 2020 and used to make predictions for June 2020.   The NN metrics for cases, hospitalization, and deaths are below:

Table 7: Neural Network ("NN")

Metric	Cases	Hospitalizations	Deaths
RMSE	272.10	431.11	142.59
MSE	74,042.14	185,555.55	23,429.21
MAE	202.34	429.40	152.00
MAPE	26.33	Undefined	761.48

See Appendix B, III. Advanced Models, Model 7: Neural Network

XGBoost

XGBoost is an open-source software library that provides a gradient boosting framework. "Gradient boosting refers to a class of ensemble machine learning algorithms that can be used for classification or regression predictive modeling problems. Ensembles are constructed from decision tree models. Trees are added one at a time to the ensemble and fit to correct the prediction errors made by prior models. This is a type of ensemble machine learning model referred to as boosting." [22]

We created an XGBoost regressor model with the following hyper parameters:

- n_estimators = 12
- max_depth=7
- eta=0.1
- subsample=0.7
- colsample_bytree=0.8
-

The n_estimators are the number of decision trees in the ensemble until no improvement in the RMSE can be found. "max_depth" is the maximum depth of each tree. "eta" is the learning rate. "subsample" is the number of samples (rows) used in each tree. "colsample_bytree" is the Number of features (columns) used in each tree. [23] We fit the model to the training data and made predictions for cases, hospitalizations, and deaths.

Table 8: XGBoost

Metric	Cases	Hospitalizations	Deaths
RMSE	385.43	183.22	43.82
MSE	148,555.63	354,816.40	1920.4
MAE	366.36	566.70	204.80
MAPE	54.68	Undefined	207.47

See Appendix B, III. Advanced Models, Model 8: XGBoost

Long Short Term Memory ("LSTM")

A Recurrent Neural Network ("RNN") has the same structure as a NN with the major difference that an RNN has loops in them that allows for information to persist. RNNs can connect shorter term memory retention to current tasks but struggle with longer term retentions. The LSTM model is an improvement over RNNs because they can connect long term previous information to the current task.

We add the LSTM layer and later add a few Dropout layers to prevent overfitting. We add the LSTM layer with the following arguments:

a. 50 units which is the dimensionality of the output space
b. "return_sequences" = True which determines whether to return the last output in the output sequence, or the full sequence
c. "input_shape" as the shape of our training set.
d. The layers included the input layer, three dense layers, and an output layer.
e. The model was compiled with the optimizer parameter set to "adam" and the loss function was "mean squared error".
f. The model was fit on the training data and predictions were made on cases, hospitalizations, and deaths.

Table 9: Long Short Term Memory ("LSTM")

Metric	Cases	Hospitalizations	Deaths
RMSE	507.27	144.70	39.34
MSE	257,331.84	20,935.26	1547.65
MAE	455.52	132.9	37.68
MAPE	64.96	Undefined	174.46

See Appendix B, III. Advanced Models, Model 9: LSTM

COMPARATIVE RESULTS

Tables 10, 11, and 12 compares the performance of our nine models based on RMSE, MSE, MAE, and MAPE.

Table 10: Positive Cases

Model	RMSE	MSE	MAE	MAPE
7-day Rolling Avg.	600.771	360,926.03	86.70	566.47
Simple Expo Smoothing	408.32	166,464	369.71	0.58
Holt Winters	**185.48**	**34,402.83**	**151.95**	**0.2136**
ARIMA	765.70	586,296.49	740.96	0.98
SARIMA	**188.46**	**35,517.17**	**155.43**	**0.223**
Facebook Prophet Model	2939.3	8,639,484.49	2,934.72	4.20
Neural Network	272.10	74,042.14	202.34	26.33
XGBoost	385.43	148,555.63	366.36	54.68
LSTM	507.27	257,331.84	455.52	64.96

Table 11: Hospitalizations

Model	RMSE	MSE	MAE	MAPE
7-day Rolling Avg.	172.286	29,682.463	167.36	Undefined
Simple Expo Smoothing	182.356	33,253.689	177.70	Undefined
Holt Winters	62.95	3962.70	53.73	Undefined
ARIMA	**42.37**	**1795.22**	**15.18**	**Undefined**
SARIMA	**32.30**	**1043.29**	**13.94**	**Undefined**
Facebook Prophet Model	823.5	678,152.25	752.64	3.244
Neural Network	431.11	185,555.55	429.40	Undefined
XGBoost	183.22	354,816.4	566.70	Undefined
LSTM	144.70	20,935.26	132.9	Undefined

Table 12: Deaths

Model	RMSE	MSE	MAE	MAPE
7-day Rolling Avg.	44.852	2011.68	41.76	243.06
Simple Expo Smoothing	**31.82**	**1012.51**	**27.64**	**1.90**
Holt Winters	66.86	4470.26	60.49	3.5490
ARIMA	**33.74**	**1138.39**	**29.53**	**0.9**
SARIMA	**12.77**	**163.07**	**3.36**	**0.392**
Facebook Prophet Model	129.54	16,780.61	100.75	7.00
Neural Network	142.59	23,429.21	152	761.48
XGBoost	43.82	1920.4	204.8	207.47
LSTM	39.34	1547.65	37.68	174.46

DISCUSSION

After comparing performance metrics, the model that showed the most consistent and highest accuracy for predicting cases, hospitalizations, and deaths was the SARIMA model. The Holt Winters model also performed well for forecasting cases. ARIMA did well for hospitalizations, and the SES model performed well for deaths. The advanced models, Facebook Prophet Model and Neural Network underperformed all other models, including the basic 7-day rolling average model. The LSTM model was the best advanced model on Hospitalizations and Deaths, and XGBoost was the best advanced model for predicting cases.

SARIMA models provide a seasonality component to ARIMA models. There is a strong seasonal component to COVID-19 given its exponential rise and decline in the first three months of the pandemic, and the SARIMA model captures that seasonality which is the reason why it out-performs the other models.

CONCLUSIONS

Our findings show that time series models are effective at forecasting a future time period for the COVID-19 pandemic. At the end of May 2020, public health officials could have used one of the time series models that we studied to predict cases, hospitalizations, and deaths for June 2020 with a high degree of accuracy. A major caveat is that the accuracy of time series models is dependent on full and accurate data collection. Our team reviewed three sources of data until we found one that transparently published its data and its methods for collection and cleaning.

Although we confirmed that a time series model can be used to predict one time period ahead, there was a wide variation in the performance metrics for each model. This may be due to the seasonal nature of the COVID-19 time series. We would need to further investigate why this is true. We also think that time series modeling of pandemics should be its own area of study. Most of the models that we saw were not built to model the spread of a virus specifically. Most were built to model financial transactions, sales, or usage. Pandemic time series modeling should be its own field of study.

References

1. Haytham H. Elmousalami and Aboul Ella Hassanien, "Day Level Forecasting for Coronavirus Disease (COVID-19) Spread: Analysis, Modeling and Recommendations", Scientific Research Group in Egypt (SRGE), Cairo, Egypt (March 15, 2020)

2. Papastefanopoulos, Vasilis, Linardatos, Pantelis, and Kotsiantis, Sotiris, "COVID-19: A Comparison of Time Series Methods to Forecast Percentage of Active Cases per Population" Applied Sciences. 10. 3880. 10.3390/app10113880 (May 2020)

3. Kumar,Vinay, Chimmula, Reddy, and Zhang ,Lei "Time series forecasting of COVID-19 transmission in Canada using LSTM networks", Chaos, Solitons and Fractals Nonlinear Science, and Nonequilibrium and Complex Phenomena (May 2020).

4. Zeroual, Abdelhafid, Harrouc ,Fouzi, Dairi, Abdelkader, and Sunc, Ying, "Deep learning methods for forecasting COVID-19 time-Series data: a Comparative study", Chaos, Solitons and Fractals Nonlinear Science, and Nonequilibrium and Complex Phenomena (July 15, 2020)

5. Kumar, Naresh & Susan, Seba, "COVID-19 Pandemic Prediction using Time Series Forecasting Models", The 11th ICCCNT 2020 conference.

6. Er, Başak, Emeç, Murat, and Ozcanhan, Mehmet, "Analysis of COVID-19 Data Using Arima Time Series Model", Conference: V. International Scientific And Vocational Studies Congress – Engineering (December 2020).

7. Mahmud, Sakib, Bangladesh COVID-19 Daily Cases Time Series Analysis using Facebook Prophet Model, Social Science Research Network

8. Ismail, Khan, Znati, Materwala, Turaev, "Tailoring time series models for forecasting coronavirus spread: Case studies of 187 countries", Computational and Structural Biotechnology Journal Volume 18, 2020, Pages 2972-3206 (September 2020).

9. Chaurasia, Vikas and Pal,Saurabh, "Application of machine learning time series analysis for prediction COVID-19 pandemic", Sociedade Brasileira de Engenharia Biomedica (October 2020).

10. Vijander Singh , Ramesh Chandra Poonia , Sandeep Kumar, Pranav Dass , Pankaj Agarwal , Vaibhav Bhatnagar & Linesh Raja, “Prediction of COVID-19 corona virus pandemic based on time series data using support vector machine”, Journal of Discrete Mathematical Sciences and Cryptography, 23:8, 1583-1597, DOI: 1080/09720529.2020.1784535.

11. Shah, Saloni; Mulahuwaish, Aos; Ghafoor, Kayhan; Maghdid, Halgurd S., “Prediction of Global Spread of Covid-19 Pandemic: A Review and Research Challenges.” TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.12824378.v1

12. Yi-Cheng Chen, Ping-En Lu , Cheng-Shang Chang, “A Time-Dependent SIR Model for COVID-19 with Undetectable Infected Persons”, IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, VOL. 7, NO. 4, (OCTOBER-DECEMBER 2020)

13. Ian Cooper, Argha Mondal , Chris G. Antonopoulos, “A SIR model assumption for the spread of COVID-19 in different communities”, Chaos, Solitons and Fractals 139 (2020)

14. Vinay Kumar, Reddy Chimmula, and Lei Zhan, “Time series forecasting of COVID-19 transmission in Canada using LSTM networks”, Chaos, Solitons and Fractals (May 2020)

15. Centers for Disease Control and Prevention, ‘United States COVID-19 Cases and Deaths by State over Time’, 2021. [Online]. Available: https://data.cdc.gov

16. Johns Hopkins University, “Coronavirus Resource Center”, 2021. [Online]. Available: https://coronavirus.jhu.edu/

17. The Atlantic, “The Covid Tracking Project”, 2021. [Online]. Available: https://covidtracking.com

18. New York State, “Governor Cuomo Issues Guidance on Essential Services Under The 'New York State on PAUSE' Executive Order”, March 20, 2020. [Online]. Available: https://www.governor.ny.gov/news/governor-cuomo-issues-guidance-essential-services-under-new-york-state-pause-executive-order

19. New York State, “Amid Ongoing COVID-19 Pandemic, Governor Cuomo Issues Executive Order Requiring All People in New York to Wear Masks or Face Coverings in Public”, April 15, 2020. [Online]. Available: https://www.governor.ny.gov/news/amid-ongoing-covid-19-pandemic-governor-cuomo-issues-executive-order-requiring-all-people-new

20. ritvikmath, “Time Series Talk : ARIMA Model”, *YouTube*, Jul. 11, 2019 [Video file]. Available: https://www.youtube.com/watch?v=3UmyHed0iYE [Accessed: Mar. 20, 2021]

21. Moto DEI, “Facebook Prophet”, August 22,2020. [Online]. Available: https://medium.com/swlh/facebook-prophet-426421f7e331

22. Jason Brownlee, “Basic Feature Engineering With Time Series Data in Python”, December 14, 2016. [Online]. Available: https://machinelearningmastery.com/basic-feature-engineering-time-series-data-python/

23. Jason Brownlee, ‘XGBoost for Regression’, March 12, 2021. [Online]. Available: https://machinelearningmastery.com/xgboost-for-regression/

24. “Compare time series predictions of COVID-19 deaths,” 2021. [Online]. Available: https://www.coursera.org/learn/compare-time-series-predictions-of-covid19-deaths/home/welcome

25. "COVID19 Data Visualization Using Python", 2021. Accessed on: March 20, 2021. [Online]. Available: https://www.coursera.org/learn/covid19-data-visualization-using-python/home/welcome

26. "Using a Keras Long Short-Term Memory (LSTM) Model to Predict Stock Prices', 2018. [Online]. Available: https://www.kdnuggets.com/2018/11/keras-long-short-term-memory-lstm-model-predict-stock-prices.html

Appendices

Appendix A	Data Sources and Collection
Appendix B	Exploratory Data Analysis and Model Building
Appendix C	Discussion of AR, MA, ARMA, ARIMA, and SARIMA models
Appendix D	AR, MA, ARMA, ARIMA, and SARIMA models
Appendix E	Facebook Prophet model
Appendix F	Discussion of Facebook Prophet model

Presentation
A Comparison of COVID 19 Time Series Models for New York

Technology
Programming Language: Python (versions 3.7 and 3.8)
IDE: Jupyter Notebook, PyCharm, Spyder
Libraries: Pandas, Numpy, matplotlib, seaborn, io, warnings, datetime
Collaboration: Github
Communication: Slack, Zoom, Google meet