SMU Data Science Review

Volume 3 | Number 1 Article 11

2020

Forecasting San Francisco Bay Area Rapid Transit (BART) Ridership

Swee K. Chew
Southern Methodist University (SMU), schew@smu.edu

Alec Lepe
Southern Methodist University (SMU), alepe@smu.edu

Aaron Tomkins
Southern Methodist University (SMU), atomkins@smu.edu

Peter Scheirer
Kaiser Permanente, peter.scheirer@kp.org

Recommended Citation

Chew, Swee K.; Lepe, Alec; Tomkins, Aaron; and Scheirer, Peter (2020) "Forecasting San Francisco Bay Area Rapid Transit (BART) Ridership," *SMU Data Science Review*: Vol. 3: No. 1, Article 11.
Available at: https://scholar.smu.edu/datasciencereview/vol3/iss1/11

Forecasting San Francisco Bay Area Rapid Transit (BART) Ridership

Swee K. Chew[1], Alec Lepe[1], Aaron Tomkins[1], and Peter Scheirer[2]

[1] Master of Science in Data Science
Southern Methodist University
Dallas, Texas USA
[2] Kaiser Permanente
Oakland, California USA
{schew,alepe,atomkins}@smu.edu
peter.scheirer@kp.org

Abstract. In this paper, we present a forecasting analysis of the San Francisco Bay Area Rapid Transit (BART) ridership data utilizing a number of different time series methods. BART is a major public transportation system in the Bay Area and it relies heavily on its riders' fares; having models that generate accurate ridership numbers better enables the agency to project revenue and help manage future expenses. For our time series modeling, we utilized autoregressive integrated moving average (ARIMA), deep neural networks (DNN), state space models, and long short-term memory (LSTM) to predict monthly ridership. As there is such a wide range of time series techniques being used in different applications today, we explore some of the most commonly-used methods to gain deeper insights into their strengths and weaknesses as it relates to our data set in particular. We apply a variety of novel transformations to our data set in an attempt to improve the forecast accuracy. One of our primary transformations was to decouple the time series into multiple different component series based on weekday and region. We then discover that different models have better performance on different weekday and regional series. While our transformations increased overall accuracy by roughly 550% across models, the decoupling of the series into multiple component series also allows for the possibility to fit different models to different series, and thus increase accuracy further. We, therefore, see this as a powerful transformation technique that can be applied to great effect when possible.

1 Introduction

Commutes in the Bay Area are notorious for being long and prone to traffic jams. Whether it be by car, bike, bus, ferry, heavy rail, light rail, cable car, or any combination of these modes of transportation, it is a daily struggle for thousands of people in the Bay Area to get to work, schools, and businesses in the region. There has also been a significant increase in rent and house prices in the Bay Area with the rise of Silicon Valley and tech companies in certain areas. Within the past few years, the usage of rideshare and carpooling apps has also increased [1]. This paper analyzes BART ridership data to better predict the ridership patterns of commuters across the Bay Area over time. The analysis will help provide strategies and recommendations to the BART agency for better public transit facilitation.

There are a wide variety of approaches to modeling time series data, and many come from different branches of statistical thought. Although some models can be specified in particular ways to reduce to or mimic other models, the interpretations and fitting procedures are very different across models. One of our primary motivations was to compare the results of different models on the same data set and while leveraging the unique qualities and data fitting procedures associated with each individual model. Of particular interest was whether nonparametric aproaches could dominate traditional statistical models (or vice versa), or whether an ensemble approach might be justified from a discovery that every model can achieve high accuracy when modeling different pieces of the same data set.

The main contributions of this paper are noted in this paragraph. We were very interested in seeing whether certain novel data transformations could improve forecast accuracy, and whether all models would benefit equally from such transformations. One of the primary transformations we perform decouples the time series into several different component series based on region and weekday. We will later see that different models perform better on different component series, even though the components are technically all from the same data set. This opens the way for using a variety of different models to help forecast one time series without explicitly averaging them in a traditional implementation of an ensemble.

To analyze and produce forecasts of the aforementioned BART data, we apply four different time series methodologies coming from different branches of statistics and machine learning. When fitting each model, we sometimes preprocessed the data in a way that was specific to the input of the model. However, we also performed some novel data cleaning and transformations of the series beforehand in an effort to help assist all models to discover the core patterns in the data and increase overall forecast accuracy. When choosing models to implement in this paper, we focused primarily on mainstream models, but also experimented with some deep learning applications. The models we chose are given below:

- Autoregressive Integrated Moving Average (ARIMA)
- Deep Neural Network (DNN)
- Bayesian Structural Time Series (State Space)
- Long Short-Term Memory (LSTM)

As a road map for this paper: after this introduction, Section 2 gives the background on BART. In Section 3, we present previous work done related to our topic. In Section 4, we describe the BART data set in details and the data preparations needed for the analysis and for reproducibility. In Section 5, we cover the four time series methods mentioned above and provide the framework used for the analysis. We introduce the custom goodness-of-fit metric that is used to assess the performance of the models and discuss the results and findings in Section 6. In Section 7, we include a discussion of ethical implications of our analysis. Lastly in Section 8, we draw the relevant conclusions and identify possible future areas of research.

2 Background

BART is a critical mode of transportation for many Bay Area residents serving Alameda, Contra Costa, San Francisco, and San Mateo counties in the San Francisco Bay Area Peninsula in California. It connects the two major cities, San Francisco and Oakland, with urban and suburban areas. Currently, it runs 5 routes along its entirely grade separated 121 miles throughout numerous cities and communities. And as of December 2019, it operates 48 stations in the four counties and has extensions planned for more stations and service into Santa Clara county. Based on the 2018 Fiscal Report, BART approximately carries 432,000 riders every weekday[1].

BART is heavily funded by the patrons it serves, with approximately 70% of the operational costs being covered by its fares, it has one of the highest fare box ratios in the United States. Better understanding how ridership has changed over time and will potentially change in the future is of great consequence to both BART and the numerous counties and communities it serves. By using models, visualizations and forecasting methods, we hope to forecast ridership more on a granular level to not only potentially better forecast BART fare revenues but also better understand travel patterns of BART patrons. This information will be extremely useful for BART in their future planning and their efforts to meet the demands of today and in the near future. Findings would also be valuable if they can be used along with the traffic data in helping reduce the traffic since people still need to get to their destination in one way or another and heavy traffic on the road could cause commuters to choose BART as their alternative.

Figure 1 shows the BART annual ridership from the fiscal year 2009-2018. After several years of strong annual ridership growth, the ridership began to decline in late FY16. Total ridership dropped to 124,171,000 in FY17, which is 3% below FY16 and it continued to drop another 3% in FY18[2].

[1] https://www.bart.gov/sites/default/files/docs/2018_BART%20Factsheet.pdf Last Accessed 9 October 2019.

[2] https://www.bart.gov/sites/default/files/docs/FINAL%20FY19%20SRTP_CIP.pdf Last Accessed 6 November 2019.

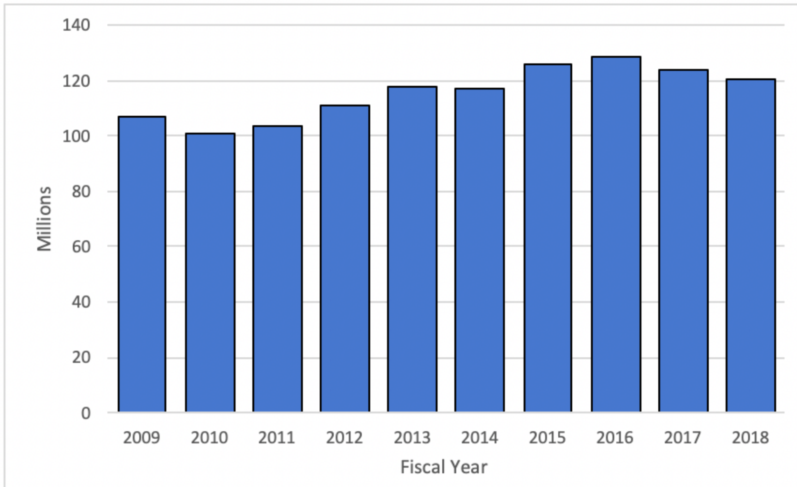

Fig. 1: BART Annual Ridership (FY09-FY18)

The decline in ridership could be due to many factors, which may include changes in demographics, services, telecommuting options, and the wide use of transportation network companies, such as Uber and Lyft. Nonetheless, in depth studies of the ridership patterns over the years and being able to forecast ridership will hugely benefit the BART agency with the operation optimization in term of evaluating whether to increase or decrease the number of train cars. Also, it will provide a better insight into how the ridership changes over time on a specific day of the week or station. For the purpose of this paper, we solely utilize the BART data set that is publicly available on the BART website.

3 Related Work

Based on our research, traditional time series and neutral network models have been used for forecasting. Some researchers explore the performance between various time series models including traditional and neural networks [2,3]. This particular paper discusses the use of neural networks to model seasonal and trend time series and compare the results to those from the Box-Jenkins seasonal ARIMA models. The finding suggests that neutral networks cannot capture seasonal and trend variations effectively with the unpreprocessed raw data and either deseasonalization or detrending can improve forecasting accuracy [4].

Hybrid models that combined classic time series and neural networks are also being studied since they demonstrate to improve forecasting accuracy than using either of the component models separately [5,6,7,8]. The conventional time series approaches perform well with the linear nature of a complex time series, whereas artificial neural network techniques are capable of capturing the nonlinear portion. Exploiting the strengths of each component, the hybrid models provide a more robust modeling framework producing more accurate forecasts.

However, when it comes to state space models and long short-term memory networks, researchers have used them for forecasting time series related work, independently from the conventional models. There have not been any comparisons, in term of model performance, to the classic time series models based on our research. Long short-term memory networks are widely used in predicting traffic flow due to its ability to learn more abstract representations in the non-linear traffic flow data and its intrinsic feature of capturing long-term dependencies in a sequential data [9,10,11].

Some studies have been done to look at forecasting ridership for some forms of transportation specifically. One analysis focuses on identifying the factors that influence the use of Tulsa Transit which serves the metropolitan area of Tulsa, Oklahoma [12]. Forecasting models are developed to predict monthly transit ridership using regression analysis (with autoregressive error correction), neural networks, and ARIMA models. The overall finding is that a simple combination of these forecasting methods yields greater accuracy than the individual models separately.

Another study looks at developing a ridership forecast model for each light rail transit station in the Madrid Metro network based on the joint use of Geographic Information Systems (GIS) and multiple regression models [13]. The GIS makes the use of distance-decay weighted regression possible, which applies the data nearer the stations a greater weighting in the model than those farther away. The results show that weighting the predictors according to the distance-decay functions improves accuracy, compare to the use of straight-line distance.

Both articles [12,13] also cite other studies that are done on forecasting ridership using multivariate regression. One of which is to develop a model to forecast ridership on alternative light rail and heavy rail extensions to the San Francisco BART system (Walters and Cervero, 2003). The method established statistical relationships between BART ridership and the characteristics of transit services and surrounding neighborhoods. Two regression equations were chosen to forecast ridership which includes predictors such as population–employment densities around stations, transit technologies, train frequencies and catchment population.

Unlike most of the models mentioned in the above literature on public transit ridership, our paper presents simple univariate time series models which could save time and resources and be useful when other predictor variables are not readily available. In addition, we propose novel data transformations prior to model fitting to improve the forecasting accuracy, compared to the conventional approach.

4 Data and Methodology

This section details data and methodology used for our paper—from the structure of the input data, to detailed descriptions of our transformations, to a visualization of the selected and transformed data.

4.1 Data Structure

The public BART ridership data[3] is comprised of ridership counts on both an hourly and daily basis in such a way that distinct day of week (eg Mondays and Tuesdays) can be generated. We collect the hourly ridership data from the 2011-2019 ridership reports. The exact attributes of each data set includes:

– Origin: Origin Station
– Destination: Destination Station
– Date: Date, Represented as YYYY-MM-DD
– Hour: Hour of Exits, Rounded Down
– Count: Count of Exits

Although we used some of the above attributes to organize the data into multiple time series, we did not use any additional attributes as exogenous regressor variables in all of our models since using them required such variables to be also forecasted into the future in order to make a forecast of our target series. Additionally, we also wanted to explore the differences between the models themselves, and some models (eg ARIMA) are not able to integrate regressor variables without changing the nature of the model.

4.2 Data Transformations

The original subject time series was comprised of hourly counts that were ordered sequentially and could then be rolled up by the day, week, and month across a span of several years (2011-2019). Since the focus of this paper is in long-term trends than hourly trends, we decided to aggregate the hourly counts into daily counts, but a daily time series was still on too small of a scale for our purposes. Therefore, to align with how BART management itself currently looks at ridership (namely, weekday averages for each month), we averaged the ridership for each weekday within every month. This resulted in a time series comprised of average ridership for each day of the week for every month across several years. Instead of each month having a single value associated with it, we therefore had a series where each month had seven sequential values associated with it (one average for each day of the week). For example, January 2012 would have one Monday associated with it that represents the average of all Mondays in the month. Similarly, it would have one Tuesday, and so on for the rest of the weekdays.

Part of the benefit of organizing the original time series into a time series of monthly averages was not only to align more closely with how ridership is currently considered within BART, but also to reduce some of the noise in the data that could hinder our models from capturing real, underlying trends at the risk of masking some of the natural variance. In addition, since many time series models already view their point forecasts as being means of a normal

[3] https://www.bart.gov/about/reports/ridership
Last Accessed 20 February 2020

(though unobservable) distribution anyway, this particular transformation is not inconsistent with the theoretical underpinnings of time series methodology.

In addition to the aforementioned transformations performed on the time series, we also paid special attention to anomaly days present in the series. Many anomalies that were filtered out were considered to be clear data errors and the strategy of removing them produced the same result as if we had imputed them (since our series was of averages already, removing them would have the same affect as mean imputation). However, we also removed a few days that were extremely high or low due to non-repeating, special events that occurred on those days.

We also paid special attention to Federal holidays. All 10 of these holidays, except Columbus Day, exhibited ridership lower than their corresponding day types. Thanksgiving in 2019, for instance, had approximately 65,000 riders while typical weekday ridership in 2019 is above 400,000. Days immediately before and after Thanksgiving and Christmas all consistently had lower ridership levels and were also filtered out. The justification for removing holidays was that other methods are often used to forecast such days separately in a time series. In fact, BART management forecasts holidays separately as well. The criteria used for removing outliers and a list of dates that were removed from the data set are documented in the Appendix, Section A.1.

After the above transformations, the resulting time series had a very strong cycle surrounding the seven days of the week, which risked masking the presence of other cycles (such as an annual cycle) as well as long-term trends. The spectral density of the time series also confirmed that the seven-day (weekly) cycle was the strongest seasonal pattern that dominated the time series. The strong weekday cycle can be seen in Figure 2.

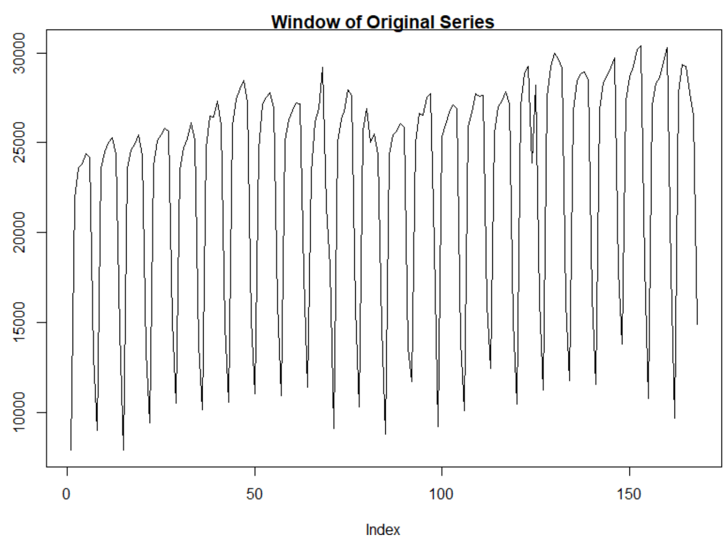

Fig. 2: Time Series of Monthly Averages of Weekdays

Since the weekday cycle obscured other patterns in the time series in some of our preliminary models, and because many time series models are actually unable to take into account more than one seasonal cycle at once [14], we performed one last modification to the data that split the overall time series into seven different time series (grouping the weekdays together). As part of justification for doing this, we fit models on both the overall time series and the decomposed time series and found that this last modification did improve forecast accuracy for many models. An example of what the time series for Mondays looked like after decoupling it from the other weekdays is shown in Figure 3.

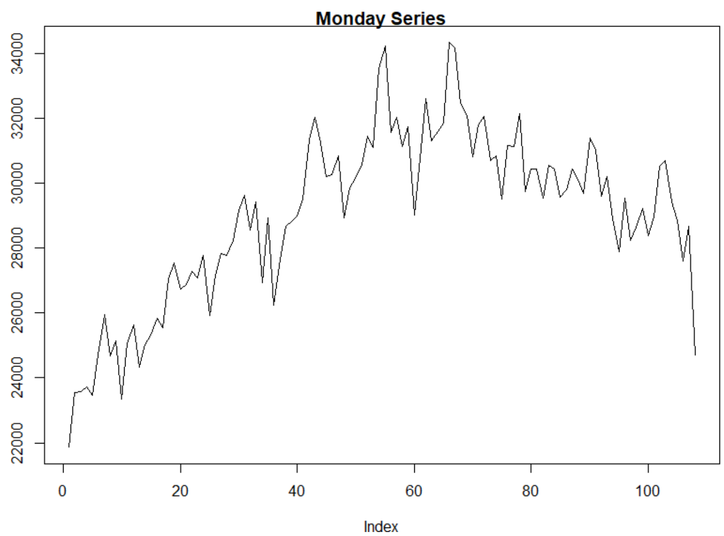

Fig. 3: Time Series of Monday Averages

4.3 Region Station Grouping

As Figure 4 [4] shows, the BART system has 48 stations. To consider each pairwise combination of passenger entry station to passenger exit station results in 2,304 combinations. Note that passengers are allowed to exit the same station from where they entered and are charged a flat excursion fare. Nonetheless, to reduce this number but still maintain useful inferences we condensed the 48 stations into 9 station regions. Each region share similar geographical locations, ridership levels, type of passenger (eg, "suburban vs urban"). See the Table A.4 in the Appendix for a list of stations in each particular region.

These 48 stations that were placed into 9 regions based on relative level of service and geographical location, were then combined into specific bidirectional pairs. For example, "Region 1 & 7" encompasses all entries within region 1 to all exits within region 7 and also all entries within region 7 to all exits within region 1. It does not include any "intra-region" travels such as an entry within region 1 but an exit outside region 7.

[4] https://www.bart.gov/schedules/developers/maps

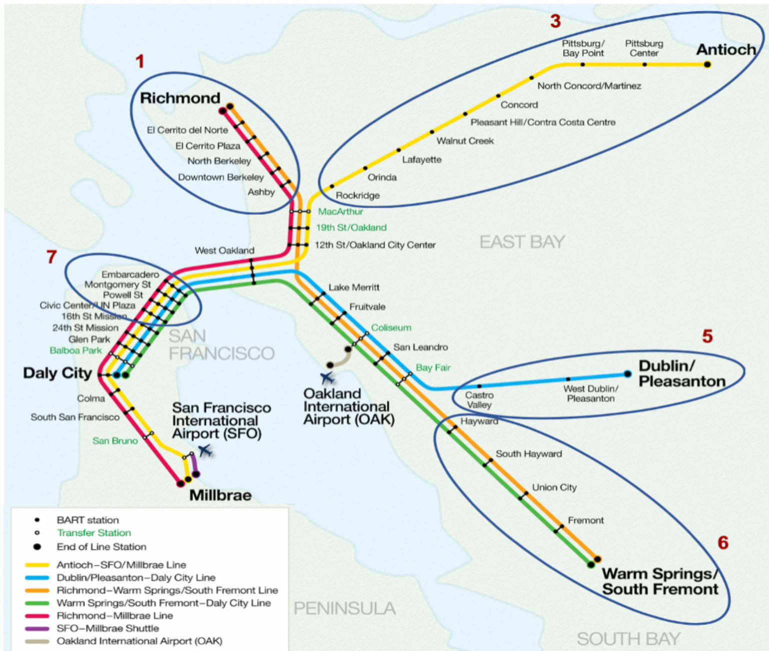

Fig. 4: BART Service Map with Regions of Interest Defined

Finally, we reduced the number of pairs to consider for later steps by selecting the pairs with the highest ridership levels. We decided to focus on regions 1, 3, 5, and 6 to and from region 7 as shown in Figure 4. These regions correspond to stations on the R Service Line, C and E Lines, L Line, and Lower A and S Lines to and from region 7 (which consists of stations in downtown San Francisco, from Embarcadero station to Civic Center station). The service lines can be referenced in Table A.4.

4.4 Notable changes to BART

Here is a list of dates in which major schedule or service changes took place:

- **June 22nd, 2003**: San Francisco International Airport (SFO) extension opens; new stations include South San Francisco, San Bruno, Millbrae, and SFO.
- **November 22nd, 2014**: Oakland International Airport (OAK) connector opens to allow for direct service from the Coliseum station to OAK.
- **March 25th, 2017**: Warm Springs/South Fremont Station Opening; Pushing the end of line for Alameda “A” line 4.6 miles to Warm Springs from Fremont
- **May 26th, 2018**: eBART Opening; Pushing the end of line for the Concord “C” line 10 miles to Antioch from Pittsburg Bay Point
- **Feb 11th, 2019**: Major schedule change, and scheduled run time and dwell tuning; Beginning of 5AM Weekday Start of Service

4.5 Data Preview

All the data we used for analysis grouped by region and day of week is shown in Figure 5. It summarizes nine years of BART ridership (2011-2019) grouped by regions and the seven days of the week.

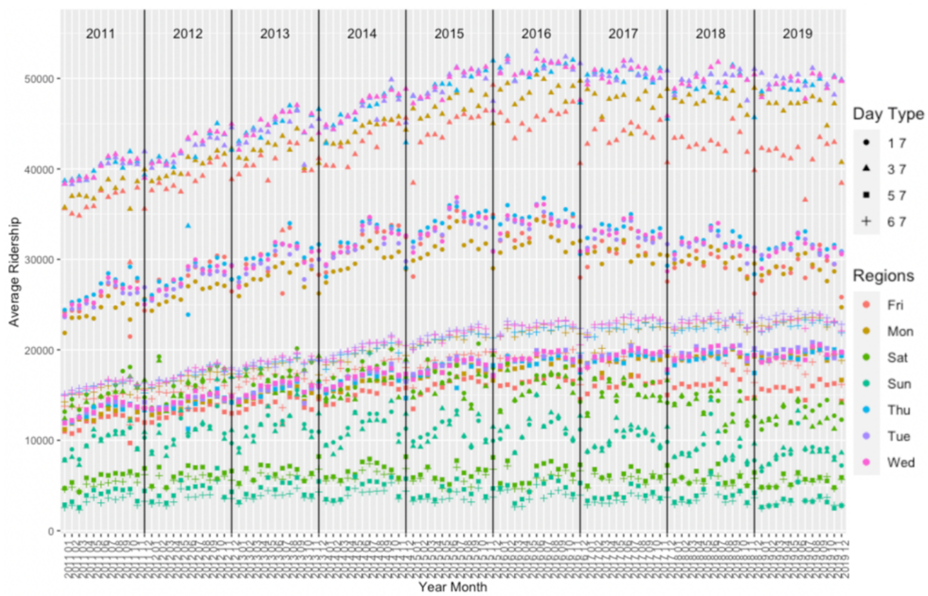

Fig. 5: Average BART Ridership by Region Grouping and Day Type

Figure 5 indicates that on weekends (Saturdays and Sundays), ridership was typically lower than on weekdays (Mondays through Fridays) in each regions pair. Friday ridership also appears to be noticeably lower than other weekday ridership and this pattern is most evident in region 3 & 7.

4.6 Methods

We used previously-described modified data from 2011-2018 as our training sample. Using these data, for each of the four algorithms, we first incorporated each calendar day sequentially (as is typically done in time series analyses). We refer to these models as the "base" models. Then for each of the four algorithms we fit a model for each day of the week separately, and we refer to them as the "day type" models.

For both types of models (base models and day type models), we then compared the forecasted 2019 monthly average ridership to the observed ridership in that same period to calculate our custom goodness of fit metric, the weighted average squared errors (WASE), as described in Section 6.1, to assess each model's performance.

5 Model Descriptions

This section details the framework for each of the time series method we used for forecasting BART monthly average ridership.

5.1 Autoregressive Integrated Moving Average (ARIMA)

The Box-Jenkins ARIMA model is one of the classic time series models used for non-stationary time series, where the ARMA part of the model assumes stationary and the "integrated" part of the model converts a non-stationary series into a stationary series by applying an initial differencing one or more times [15]. A seasonality term can be added to the model if the data contains a seasonal component. The seasonal order, s, can be any integer value greater than one, however, it is often the case that s = 4 is associated with quarterly data and s = 12 is generally associated with monthly data [15].

5.2 Deep Neural Network (DNN)

Neural networks are one of the most popular areas of machine learning currently being studied and implemented today. A deep neural network (also known as a deep feedforward network, DNN, or multi-layer perceptron) with no hidden layers can be made to be identical to any regressive model, and even an ARIMA model. In this sense, a neural network can be seen as a generalization of some traditional time series models. In fact, neural networks have been advocated as an alternative to traditional forecasting methods. When we expand the neural network by including one or more hidden layers, however, the model starts mapping non-linear relationships in the data. Intuitively, we can almost view some DNNs as non-linear versions of more traditional time series models [16].

5.3 Bayesian Structural Time Series (State Space)

The time series model we chose from a Bayesian methodology was a type of state space model called a Bayesian structural time series (also known as a dynamic linear model). Although these models can be set up in a way as to reduce to an ARMA model, the framework and fitting of the model is very different. Conceptually, we can view a state space model as a combination of different components of the series (represented as hidden state variables) that evolve according to their own system dynamics. The components can then be combined together to explain the observed time series; typically, components for trend, seasonality, and exogenous regressors are used to represent the hidden state variables.

Such models are usually estimated using Markov chain Monte Carlo (MCMC) methods, which iteratively steps around the posterior distribution of the parameter values that best explain the observed data; however, they can also be estimated through maximum likelihood [17]. Regardless of how the model is fit, the theoretical framework of such models is that of Bayes' theorem. That is, we can view the values and evolution of the state parameters as that which makes the observed data most likely. If we assume the errors of the observational and state processes are Gaussian, then the Bayesian updating actually simplifies into tractable mathematical formulas that can sequentially compute the posterior mean of each state parameter using a popular method known as Kalman filtering [17].

5.4 Long Short-term Memory (LSTM)

In addition to a deep neural network, our final model was a special type of recurrent neural network (RNN) called long short-term memory (LSTM). One of the main differences between RNNs and DNNs is that the inputs for RNNs are fed into the model sequentially, and then the network can propel forward what it learns at each time step into future time steps by maintaining the same shared hidden units across the entire network. Another difference is that DNNs treat relative time units as coefficients in a (non-linear) regressive type of context, whereas RNNs are structured to emphasize the sequential order in the data and thus can capitalize on the time-based patterns that naturally occur in a time series. Although we do not go into detail about the specific internals of LSTM in this paper, the primary reason LSTMs are often used in practice over normal RNNs is to overcome the problem of vanishing or exploding gradients that can occur when the network is trying to learn relationships over long sequences of data [18]. Since our time series is a fairly long sequence, a gated RNN like an LSTM was essential to use.

There are a variety of ways to construct an LSTM model for predicting a time series [19,20,21,22]. We intentionally used a fairly basic architecture with one just LSTM layer that fed its output at the last unit into a deep neural network, which then output a vector of an entire year of predictions. We did not stationarize the data beforehand, as we wanted to see if such a model could achieve high accuracy without this classical preprocessing step (which is normally associated with traditional ARMA models).

6 Results and Analysis

This section introduces the custom goodness-of-fit metric we used to evaluate the models performance and discuss the results and findings from the analysis.

6.1 Statistics Used to Assess Model Performance

In order to measure the effectiveness of the forecasting algorithms, we modified the average square error (ASE) equation. We designed our custom statistic, weighted average squared error (WASE), to penalize positive residuals more than negative residuals, at two times the rate. As with ASE, a lower WASE indicates a model with more suitably accurate fits.

$$Average\ Squared\ Error\ (ASE) = \frac{1}{n}\sum_{i=0}^{n}(x_i - \hat{x_i})^2 \tag{1}$$

$$Positive\ Residuals\ Statistic\ (PRS) = \frac{2}{n}\sum_{i=0}^{n}(x_i - \hat{x_i})^2\ when\ x_i > \hat{x_i} \tag{2}$$

$$Negative\ Residuals\ Statistic\ (NRS) = \frac{1}{n}\sum_{i=0}^{n}(x_i - \hat{x_i})^2\ when\ x_i < \hat{x_i} \tag{3}$$

$$Weighted\ Average\ Squared\ Error\ (WASE) = PRS + NRS \tag{4}$$

Where

- n represents the number of observations
- x_i represents the observed value
- $\hat{x_i}$ represents the predicted value

As defined, ASE places equal weight to situations when a model over-predicts as much as on situations when a model under-predicts. When a model over-predicts then by definition the fitted value exceeds the observed value, meaning contextually BART expected more riders than those who actually rode. BART would then schedule 9 cars on a given train where only 8 cars were needed. This scenario is pleasant for riders since there is more space on the cars for each rider than would otherwise be available, but for BART there was needless expense of that extra car. On the other hand, when a model under-predicts then by definition the fitted value fell short to the observed value, meaning contextually BART expected fewer riders than those who actually rode. BART would then schedule 9 cars on a given train where 10 cars were in fact needed. This scenario is unpleasant for riders since there is less space on the cars for each rider than would otherwise be available, but for BART there was a reduced expense operating that train.

We were motivated to identify a model that generally favors the experience of the average rider over the costs faced by BART thus we needed add a weight to the regular ASE. With ASE being a "lower is better" model statistic, we only needed to increase the weight for under-predictions (which are situations with riders who end up being too close to each other). In this paper we call under-predictions as "positive residuals" and our judgement supported a mere doubling of its weight (but other weights could have been also possible) from 1/n to 2/n. The weight for over-predictions was kept the same, at 1/n.

6.2 Modeling and Evaluation

Table 1 shows the WASE values, per 1,000,000 for the base models by regions pair. The prediction horizon is set to be 84 (7 days x 12 months), in order to forecast the monthly average ridership for 2019.

Table 1: Summary WASE Statistics for the Base Models

Regions	ARIMA	DNN	State Space	LSTM
1 & 7	12.0	16.7	2.4	269.8
3 & 7	21.9	19.4	6.1	995.0
5 & 7	4.3	4.3	1.4	214.5
6 & 7	7.9	7.5	1.6	1,689.9

* Results are per 1,000,000.

The results from Table 1 are plotted in Figure 6. Since the WASE values for the LSTM models are much higher, about 26 to 298 times higher than the average WASE values of the remaining three models, they are excluded from the plot for a better visualization.

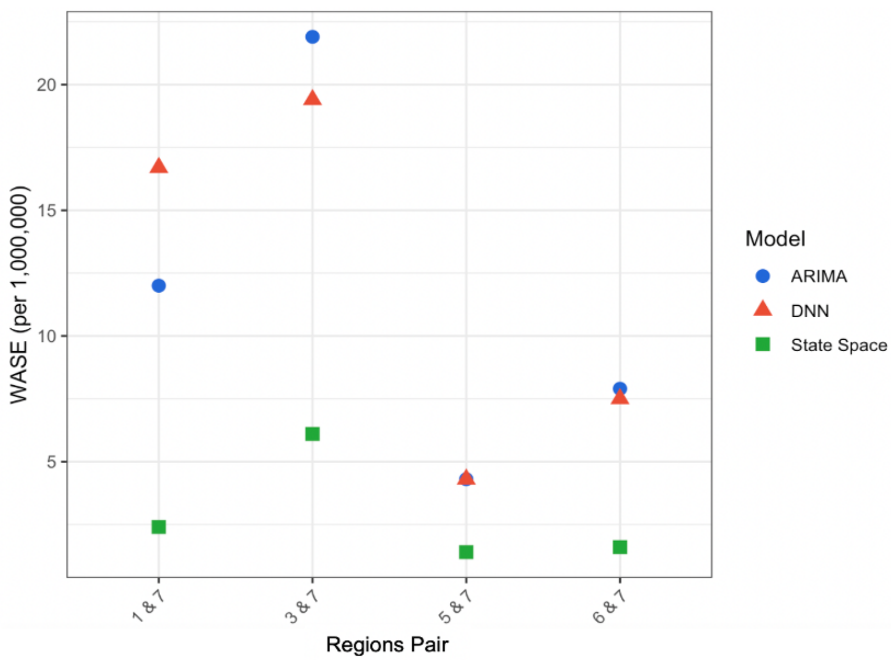

Fig. 6: WASE Results by Model Type by Regions Pair for the Base Models. LSTM results are excluded.

From Figure 6, we can see that the state space models have the lowest WASE, thus perform the best across regions pairs. The ARIMA and DNN performance varies based on the regions. For the regions pairs, 5 & 7 and 6 & 7, the WASE values for ARIMA and DNN are close to one another and are not too far from the state space. However, for the regions pairs, 1 & 7 and 3 & 7, there is a performance separation between the ARIMA and DNN models. In addition, the ARIMA performs better for 1 & 7 regions pair and the DNN does a better job with 3 & 7 regions pair.

Table 2: Summary WASE Statistics for the Day Type Models

Day Type	**Regions**	**ARIMA**	**DNN**	**State Space**	**LSTM**
Mondays	1 & 7	4.51	2.36	3.14	54.01
Mondays	3 & 7	6.62	8.88	10.22	586.29
Mondays	5 & 7	1.72	2.36	1.86	100.24
Mondays	6 & 7	3.85	2.61	2.39	121.64
Tuesdays	1 & 7	0.86	1.26	2.09	227.79
Tuesdays	3 & 7	2.14	4.80	4.61	91.90
Tuesdays	5 & 7	0.56	0.55	0.78	14.14
Tuesdays	6 & 7	0.87	0.61	1.03	198.50
Wednesdays	1 & 7	1.36	2.75	2.42	75.61
Wednesdays	3 & 7	1.47	5.18	6.63	680.63
Wednesdays	5 & 7	0.87	0.76	1.07	151.81
Wednesdays	6 & 7	0.91	0.99	0.91	135.83
Thursdays	1 & 7	1.87	1.46	1.41	212.61
Thursdays	3 & 7	5.75	3.00	2.88	300.70
Thursdays	5 & 7	1.45	0.45	0.36	94.41
Thursdays	6 & 7	1.60	0.48	0.37	63.91
Fridays	1 & 7	2.74	4.09	2.99	326.58
Fridays	3 & 7	7.20	6.62	6.75	375.11
Fridays	5 & 7	2.81	1.18	1.11	25.40
Fridays	6 & 7	3.42	0.96	1.51	32.01
Saturdays	1 & 7	3.72	3.78	1.76	12.04
Saturdays	3 & 7	2.32	3.19	2.39	14.76
Saturdays	5 & 7	0.88	1.24	0.85	6.10
Saturdays	6 & 7	0.49	0.79	0.60	7.36
Sundays	1 & 7	3.70	2.94	2.59	15.56
Sundays	3 & 7	2.17	2.27	7.52	44.26
Sundays	5 & 7	0.96	3.02	1.99	2.92
Sundays	6 & 7	0.42	1.30	1.74	4.73
All Day Types	1 & 7	2.68	2.66	2.34	132.03
All Day Types	3 & 7	3.95	4.85	5.86	299.09
All Day Types	5 & 7	1.32	1.37	1.15	56.43
All Day Types	6 & 7	1.65	1.10	1.22	80.57

* Results are per 1,000,000. *All Day Types* results are obtained by taking the average of WASE values from Monday to Sunday for each regions pair.

Next, we look at the day type models performance. Again, these models are from fitting decoupled time series of the base model by the day of the week. Table 2 shows the WASE values for the day type models by the day of the week and by regions pair. The prediction horizon is set to be 12 (as there are 12 months in a year), in order to forecast the monthly average ridership for 2019 for each day type. The LSTM models still have the highest WASE values among all the model types. However, it seems to perform better at forecasting the monthly average ridership for weekends than weekdays (the WASE values are smaller). The results from Table 2 are plotted in Figure 7. For the same reason that was described earlier, we exclude plotting the LSTM models for a better visualization.

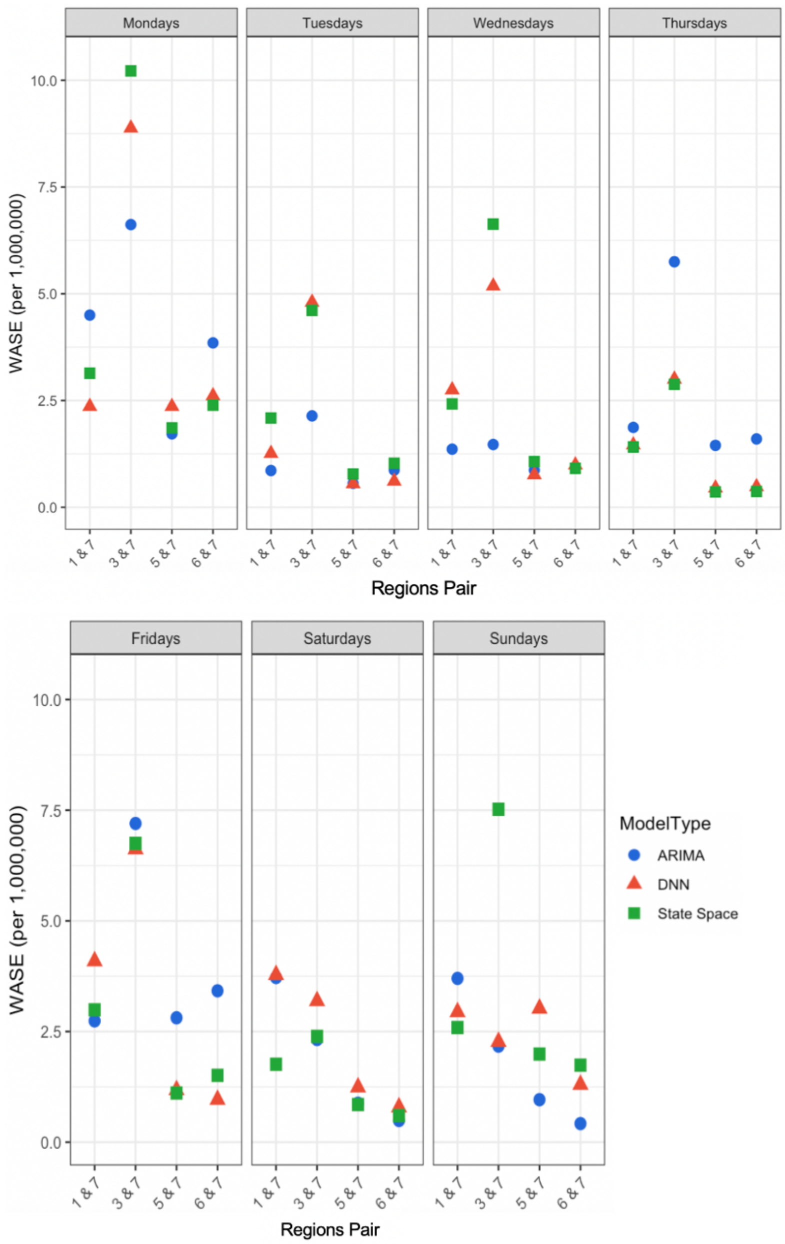

Fig. 7: WASE Results by Model by Regions Pair for Each Day of the Week. LSTM results are excluded.

Figure 7 shows that the performance varies based on the regions pairs and day types, there is no one single model type that outperforms the rest across all the combinations. For the regions pair 3 & 7, the WASE values are generally higher compared to other regions pairs for each day of the week, except Saturdays. This may be an indication of unusual ridership patterns for that specific regions pair. In general, the three models (ARIMA, DNN and State Space) perform better on the regions pairs 5 & 7 and 6 & 7, the WASE values are lower. In some cases, there is insignificant performance difference among the three (eg regions pair 5 & 7 for Tuesdays, 6 & 7 for Wednesdays and Saturdays). Also, the WASE values between the DNN and State Space models are overall closer to one another than the ARIMA's.

Next, we combined the day type models performance by averaging their WASE values and call the result *Aggregated Decomposed Models*. Table 3 below indicates the WASE values for the base models to the comparable aggregated decomposed models. The results from Table 3 are plotted in Figure 6. For the same reason that was described earlier, we exclude plotting the LSTM models. However, it is worth noting that regardless of the model type, the aggregated decomposed models perform better on the regions pairs 5 & 7 and 6 & 7 than the other two.

Table 3: Summary WASE Statistics for the Base Models and the Aggregated Day of the Week Models

Model Type	Regions	Base Model	Agg Decomposed Model
ARIMA	1 & 7	12.0	2.68
ARIMA	3 & 7	21.9	3.95
ARIMA	5 & 7	4.3	1.32
ARIMA	6 & 7	7.9	1.65
DNN	1 & 7	16.7	2.66
DNN	3 & 7	19.4	4.85
DNN	5 & 7	4.3	1.37
DNN	6 & 7	7.5	1.10
State Space	1 & 7	2.4	2.34
State Space	3 & 7	6.1	5.86
State Space	5 & 7	1.4	1.15
State Space	6 & 7	1.6	1.22
LSTM	1 & 7	269.8	132.03
LSTM	3 & 7	995.0	299.09
LSTM	5 & 7	214.5	56.43
LSTM	6 & 7	1689.9	80.57

* Results are per 1,000,000.

The aggregated decomposed models overall have a lower WASE than their respective base models, as shown in Figure 8, for each combination of model type and regions pair. For the ARIMA and neural network models, the differences are substantial but for the state space models the differences are trivial. Using the aggregated decomposed models, the WASE values are about 4.5 times lower on average for the ARIMA, 5.1 times for the DNN, and 1.1 times for the state space.

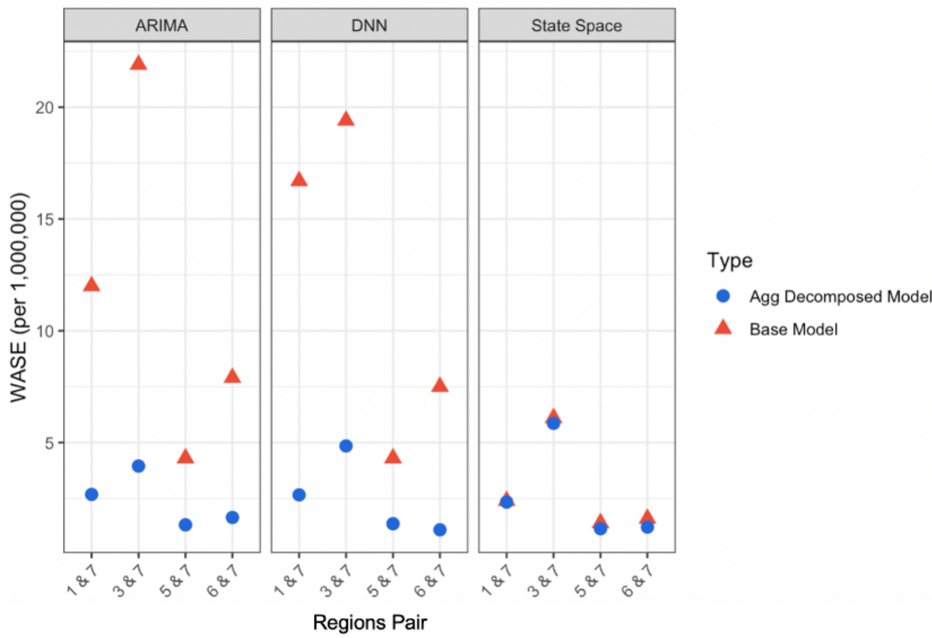

Fig. 8: Performance Comparison - WASE Results by Regions Pair for Each Model Type. LSTM results are excluded.

In order to illustrate that the forecasts using the aggregated decomposed model are better than the base model, we select one regions pair and plot the actual and predicted ridership. Figure 9 shows the predicted values for 2019 superimposed on the original time series with the actual monthly average values. The top graph shows the forecasts using the base model and the bottom one using the aggregated decomposed model. Both of them display the actual and predicted monthly average ridership for regions pair 1 & 7 using DNN. The actual and predicted values are more aligned with the aggregated decomposed models than the base model, indicating that the aggregated decomposed model has the higher forecasting accuracy.

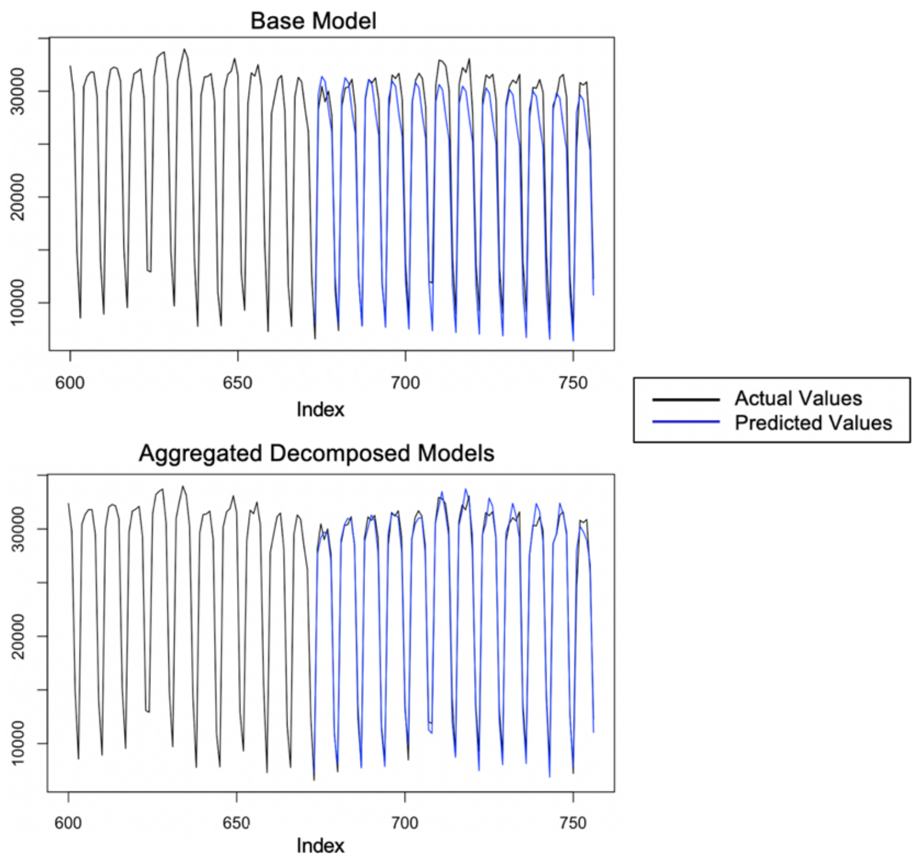

Fig. 9: Performance Comparison between the Base Model and the Aggregated Decomposed Model for Regions Pair 1 & 7 Using DNN

ARIMA Our analysis with the ARIMA included evaluating seasonality components from s = 0 to s = 12 on each data set with the aim to preserve the one with the best results based on our custom statistic. Not surprisingly, most seasonal orders were multiples of 12 denoting a repeatable monthly trend year over year. However, there were also some seasonal orders that were 0 or 7 denoting no seasonal trend or a trend that occurred approximately every 7 time periods (corresponding to the 7 day types in a week). Overall the seasonal order of s = 7 generated the best WASE statistic for the base model, for all region pairs, and of all day types grouped together. Putting this into context, these results suggest that BART ridership usually does not change dramatically year to year but mainly from day type to day type.

Overall, the ARIMA models performed moderately well when compared to other models. This relative performance may have been based on the fact that ARIMA models are a class of models that explain a given time series based on its own previous values. The ARIMA models are quite flexible in that they can represent different types of time series—pure autoregressive (AR), pure moving average (MA) and combined AR and MA (ARMA) series. However, their major limitation is that a linear correlation structure is assumed among the time series values, which means no nonlinear patterns can be captured by the models [8]. We believe this partially explained the moderate performance we observed with the ARIMA models.

Deep Neural Network When fitting the deep neural network (DNN), each network was trained 20 times using different initial weights each time, and the final forecast was the median forecast from all of the models. In this way, the neural network was almost like an ensemble of networks though each having the same architecture. Additionally, 5-fold cross-validation was used to select the optimal number of layers and number of hidden nodes. Since our preliminary transformations greatly simplified the time series, the neural network was able to output optimal results using just one hidden layer.

Differencing was used in some of the weekday and region subsets of the series to stationarize and deseasonalize the data to enable a better fit. The differencing that was utilized was either first-order or annual differencing; about half of the data sets had optimal results with just first-order differencing and the other half used annual differencing. When we model data to compare results to the base time series that did not separate the days of the week, 7-day differencing was used. It should be noted that the fitting of the DNN models uses a special package in R called *nnfor*.

The DNN models overall performed about the same as the ARIMA models in terms of our WASE metric, though the performance varied somewhat by day of the week and by regions pairs. It should be mentioned that the DNN does perform well for some days of the week and regions where the ARIMA model suffers, and vice versa. This would actually suggest that an ensemble method might perform best overall.

State Space The Bayesian state space model performed the best out of all other models before our novel transformations, and performed comparably to the other models after those transformations. This model was allowed to have annual seasonality, which it implemented using a method analogous to having seasonal dummy variables [23]. Although these models are also able to utilize an AR (auto-regressive) component, we did not use an AR component in the model because we wanted to make it more distinct from the ARIMA model.

After fitting to the data, the model was able to identify a clear annual seasonality (based on a 12 month cycle), but the long-term trend was roughly ten times greater in scale. Much of the accuracy of the state space model came more from its ability to predict the long-term trend and distinguish this component from the seasonal cycle. Since this model defines a separate latent variable for the trend, and then constantly updates this trend as it sees new data, we believe this special formulation gave it the needed boost to perform well relative to the other models.

LSTM LSTMs are very versatile in that there are a seemingly endless number of ways one can structure such a model to solve a given forecasting problem. The particular LSTM architecture we used is to model an entire year of data based off of the entire previous year of data (and using a rolling window at the most granular level of the time series to get the maximum number of observations). The LSTM was therefore designed to essentially learn how a rolling 12-month period influences the following 12-month period, across several years of data. We implemented this model as a single LSTM layer that then outputs its shared hidden units into a deep feed forward layer. The deep feedforward layer then outputs an entire vector of predictions for the target year. Although this is a much more ambitious design than a one-step-ahead architecture, we want to test how well the LSTM would perform given such a difficult task.

The LSTM model we used results in the worst performance of the models. We believe that the primary reason for this is because the test data (the last year of the data, which is 2019) is essentially all trending downward, while the model learns on training data comprised primarily of an upward trend and a reversal trend (from upward to downward). This suggests that our particular implementation of an LSTM model is best used when the data is stationary, or the trend is fairly consistent throughout the training and test set, or all possible trends occur with enough observations in the training set.

Another factor to note is that LSTM is typically trained using a mean squared error (MSE) cost function, while our performance metric is a special metric that gives a greater weight to errors in a particular direction. In other words, the LSTM is optimizing parameters that minimize a cost function that is not quite the same as the performance metric. Since there is a risk for deep learning models to overfit, it is possible that a better model could be achieved with a custom cost function. It should also be noted that the normal ASE of the LSTM is about half that of the WASE.

7 Ethics

The ethics involved in time series modeling are very unique, especially when dealing with univariate time series, as our paper does. Time series data is often aggregated and comprised of comparatively few observed data points, at least relative to typical regression or classification applications. Due to the high-level aggregation, there is also usually no personally-identifiable information being used. Rather, time series modeling typically takes a birds eye perspective and looks at the big picture; it points to where trends are going and what seasonality looks like. Because of this high-level perspective, time series models can carry a lot of weight in broad strategic decision making and managerial justifications, and so it is imperative to understand the internal workings of time series models and how particular models extrapolate historical data into the future.

In a time where it is becoming increasingly easier and easier to implement complex statistical models using software packages without understanding how those models use historical data and forecast it into the future, there is a great deal of risk for practitioners to simply put faith into a model's predictions without understanding why the models are making the predictions they are, and what could cause the model to make different prediction. In the realm of time series especially, certain choices are made when constructing a model that can deterministically guarantee how the model will extrapolate a trend into the future before even seeing the data. Therefore, we believe it is especially important in the time series realm for data practitioners to understand the internal workings and nuances of the models they implement.

Specifically for BART ridership, some ticket data such as Clipper Card data, is considered personally identifiable information (PII) [24]. Similar to a Social Security Number, some Clipper Card data can be personally identifiable to an individual. The nine digit Clipper Card serial number and linked telephone number, for instance, is considered PII and cannot be publicly available for all to view [25]. Therefore, not all data can be utilized for analysis regardless of how valuable such analyses can be. Metropolitan Transportation Commission (MTC) is the transportation planning, financing and coordinating agency for the nine-county San Francisco Bay Area. BART, for a variety of reasons, is attempting to be a Clipper only agency [26]. As a result, BART and MTC will have more and more PII about its riders. It is imperative that BART, MTC, and other agencies recognize that this type of granular level data is not to be used lightly and they need to comply with the ACM Code of Ethics and Professional Conduct [27] if they were to use this type of data for analysis.

8 Conclusions and Future Work

Based on the modeling and analysis, we have seen that our time series transformations (removing outliers, averaging values, and decoupling the weekdays in the series) can play a tremendous role in helping models to better uncover the patterns in the series and increase forecasting accuracy overall. Of particular note was the decoupling of the seven days of the week into seven different time series, which especially boosted the accuracy of the ARIMA and DNN models. Although such a process may seem laborious in some contexts, clever software engineering can abstract away a lot of the labor.

Additionally, we saw that each model had varying success for different regions pairs and day types, suggesting that when we slice the same series into different underlying component series, these series can have their own unique characteristics that play to the strengths of some models over others. Since it would be difficult to implement a single model that can dominate all other models on all component series, the decoupling transformation opens the door to the possibility of using all models by fitting specific models to specific component series.

In regards to our BART dataset in particular, we found that the Bayesian state space model generally outperformed all other models if no transformations are applied. We believe the performance of this model comes from its ability to accurately decompose a time series into seasonal and trend pieces in an optimal way (since it is a Bayesian approach, it is theoretically the most optimal decomposition given the observed data). Of course, such a model would not necessarily outperform other models unless the data itself was consistent to such an architecture. Therefore, we believe that BART ridership is well-described using trend and seasonality as components that sum up to the observed series.

Since the results from this paper suggest that one model performs better than the other under certain circumstances, for future work an ensemble approach could be applied from the models developed in this paper to further improve the forecasting accuracy. In addition, this paper presents a univariate time series analysis which serves as a time and cost saving tool to predict monthly average ridership. If resources are allowed, multivariate analysis with the addition of potentially useful predictors (such as the population density of an area, the characteristics of surrounding neighborhoods and traffic) should be examined to provide a more insightful tool.

References

1. Srujana Bhoopanam. *Alternative modes of transportation and BART A proposal to integrate competing transportation systems in the San Francisco Bay Area, California*. PhD thesis, Columbia University, 2019.
2. Tim Hill, Marcus O'Connor, and William Remus. Neural network models for time series forecasts. *Management science*, 42(7):1082–1092, 1996.
3. Zaiyong Tang, Chrys De Almeida, and Paul A Fishwick. Time series forecasting using neural networks vs. box-jenkins methodology. *Simulation*, 57(5):303–310, 1991.
4. G Peter Zhang and Min Qi. Neural network forecasting for seasonal and trend time series. *European journal of operational research*, 160(2):501–514, 2005.
5. Mehdi Khashei and Mehdi Bijari. A novel hybridization of artificial neural networks and arima models for time series forecasting. *Applied Soft Computing*, 11(2):2664–2675, 2011.
6. Mehdi Khashei and Mehdi Bijari. An artificial neural network (p, d, q) model for timeseries forecasting. *Expert Systems with applications*, 37(1):479–489, 2010.
7. Ashu Jain and Avadhnam Madhav Kumar. Hybrid neural network models for hydrologic time series forecasting. *Applied Soft Computing*, 7(2):585–592, 2007.
8. G Peter Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.
9. Hongxin Shao and Boon-Hee Soong. Traffic flow prediction with long short-term memory networks (lstms). In *2016 IEEE Region 10 Conference (TENCON)*, pages 2986–2989. IEEE, 2016.
10. Yongxue Tian and Li Pan. Predicting short-term traffic flow by long short-term memory recurrent neural network. In *2015 IEEE international conference on smart city/SocialCom/SustainCom (SmartCity)*, pages 153–158. IEEE, 2015.
11. Xiaolei Ma, Zhimin Tao, Yinhai Wang, Haiyang Yu, and Yunpeng Wang. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, 54:187–197, 2015.
12. Wen-Chyuan Chiang, Robert A Russell, and Timothy L Urban. Forecasting ridership for a metropolitan transit authority. *Transportation research part A: policy and practice*, 45(7):696–705, 2011.

13. Javier Gutiérrez, Osvaldo Daniel Cardozo, and Juan Carlos García-Palomares. Transit ridership forecasting at station level: an approach based on distance-decay weighted regression. *Journal of Transport Geography*, 19(6):1081–1092, 2011.
14. Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2 edition, 2018.
15. Wayne A Woodward, Henry L Gray, and Alan C Elliott. *Applied time series analysis with R*. CRC press, 2 edition, 2017.
16. Rob J Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, 2 edition, 2018.
17. Patrizia Campagnoli, Giovanni Petris, and Sonia Petrone. *Dynamic Linear Models with R*. Springer, 2009.
18. Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016.
19. Matt Dancho and Sigrid Keydana. Predicting sunspot frequency with keras, 2018. https://blogs.rstudio.com/tensorflow/posts/2018-06-25-sunspots-lstm/.
20. Francois Chollet and J.J. Allaire. Time series forecasting with recurrent neural networks, 2017. https://blogs.rstudio.com/tensorflow/posts/2017-12-20-time-series-forecasting-with-recurrent-neural-networks/.
21. Christopher Olah. Understanding lstm networks, 2015. https://colah.github.io/posts/2015-08-Understanding-LSTMs/.
22. Jason Brownlee. Time series forecasting with long short-term memory network in python, 2017. https://machinelearningmastery.com/time-series-forecasting-long-short-term-memory-network-python/.
23. Steven L. Scott. Fitting bayesian structural time series with the bsts r package, 2017. http://www.unofficialgoogledatascience.com/2017/07/fitting-bayesian-structural-time-series.html.
24. Clipper faq, 2020. https://www.clippercard.com/ClipperWeb/faq.do.
25. Metropolitan transportation commission clipper program privacy policy, 2020. https://www.clippercard.com/ClipperWeb/privacy.do.
26. Bay area rapid transit: Tickets and clipper, 2020. https://www.bart.gov/tickets.
27. Acm code of ethics and professional conduct, 2018. https://ethics.acm.org/.
28. Arun Srinivasan Matt Dowle. data.table: Extension of ‘data.frame‘, 2019. https://cran.r-project.org/web/packages/data.table/data.table.pdf.
29. Dan Vanderkam. dygraphs: Interface to ’dygraphs’ interactive time series charting library, 2018. https://cran.r-project.org/web/packages/dygraphs/dygraphs.pdf.
30. Garrett Grolemund Vitalie Spinu. lubridate: Make dealing with dates a little easier, 2020. https://cran.r-project.org/web/packages/lubridate/lubridate.pdf.
31. Wayne Woodward. tswge: Applied time series analysis, 2016. https://cran.r-project.org/web/packages/tswge/tswge.pdf.
32. Nikolaos Kourentzes. nnfor: Time series forecasting with neural networks, 2019. https://cran.r-project.org/web/packages/tswge/tswge.pdf.

R packages used for analysis [28,29,30,31,32]

A Appendix

This section includes figures and tables that provide additional information to supplement the analysis. We present the methodology we used to handle outliers, the BART station list, and the service maps.

A.1 Dealing with Outliers

There was much debate and discussion on how to handle outlier days; days with either high or low ridership and not just limited to holidays. Below is an example as to why outliers are a problem:

Table A.1: List of Selective Outliers with Abnormal Ridership

Date	Day Type	Total Ridership
2013-06-02	Sunday	149,825
2013-06-09	Sunday	142,541
2013-06-16	Sunday	157,191
2013-06-23	Sunday	148,431
2013-06-30	Sunday	300,741

The average of these five Sundays listed in Table A.1 is 179,745 which is not a good approximation to any of these Sundays. It over approximates the first four Sundays by about 30,000 while simultaneously underestimating the last Sunday by 120,000. By removing the outlier (which witnessed the annual Gay Pride Parade in San Francisco), we get an average of 149,497 better approximating the first four Sundays with one number.

There are many ways to label days with abnormal ridership patterns but for simplicity we decided on a basic approach: days with ridership well above or below the norm for each day type. Table A.2 lists the criteria for days to be labeled as either abnormally high or low ridership.

Table A.2: Ridership Criteria Used to Determine Abnormal Ridership

Ridership Indicator	Day of Week	Ridership Criteria
Low	All Days	50,000 or lower
High	Weekends (Saturdays to Sundays)	250,000 or higher
High	Weekdays (Mondays to Fridays)	500,000 or higher

Table A.3 shows a list of non-holiday dates, or surrounding dates, between 2011 and 2019 that were removed due to the ridership levels higher or lower than expected, either due to holidays, strikes, or local events greatly affecting the ridership.

Table A.3: List of Days with Abnormal Ridership

Date	Ridership Indicator	Day of Week	Total Ridership
2011-06-26	High	Sun	253,174
2011-10-16	Low	Sun	3,729
2012-06-24	High	Sun	266,868
2012-10-06	High	Sat	318,812
2012-10-31	High	Wed	570,234
2013-06-29	High	Sat	255,733
2013-06-30	High	Sun	300,741
2013-07-01	Low	Mon	3,265
2013-07-02	Low	Tue	8
2013-07-03	Low	Wed	6
2013-08-31	High	Sat	272,751
2013-10-18	Low	Fri	2,832
2013-10-21	Low	Mon	2
2014-06-29	High	Sun	282,111
2014-10-11	High	Sat	271,349
2014-10-31	High	Fri	508,252
2015-06-19	High	Fri	546,247
2015-06-27	High	Sat	276,956
2015-06-28	High	Sun	333,073
2016-01-30	High	Sat	279,415
2016-02-05	High	Fri	526,415
2016-02-06	High	Sat	413,097
2017-01-21	High	Sat	348,168
2017-06-15	High	Thu	521,066
2017-06-25	High	Sun	250,259

A.2 BART Station List

A list of all active BART stations as of December 31, 2019, is shown in Table A.4 below. It indicates the respective service line, region, county and city each station belongs to.

Table A.4: BART Station List

Full Name	Op Name	Region	County	City
Ashby	R10	1	Alameda	Berkeley
Berkeley	R20	1	Alameda	Berkeley
North Berkeley	R30	1	Alameda	Berkeley
El Cerrito Plaza	R40	1	Contra Costa	El Cerrito
El Cerrito Del Norte	R50	1	Contra Costa	El Cerrito
Richmond	R60	1	Contra Costa	Richmond
12th St Oakland	K10	2	Alameda	Oakland
19th St Oakland	K20	2	Alameda	Oakland
MacArthur	K30	2	Alameda	Oakland
Rockridge	C10	3	Alameda	Oakland
Orinda	C20	3	Contra Costa	Orinda
Lafayette	C30	3	Contra Costa	Lafayette
Walnut Creek	C40	3	Contra Costa	Walnut Creek
Pleasant Hill	C50	3	Contra Costa	Pleasant Hill
Concord	C60	3	Contra Costa	Concord
North Concord	C70	3	Contra Costa	Concord
Pittsburg-Bay Point	C80	3	Contra Costa	Pittsburg
Pittsburg Center	E20	3	Contra Costa	Pittsburg
Antioch	E30	3	Contra Costa	Antioch
West Oakland	M10	4	Alameda	Oakland
Lake Merritt	A10	4	Alameda	Oakland
Fruitvale	A20	4	Alameda	Oakland
Coliseum	A30	4	Alameda	Oakland
San Leandro	A40	4	Alameda	San Leandro
Bayfair	A50	4	Alameda	San Leandro
Oakland Int'l Airport	H10	4	Alameda	Oakland
Castro Valley	L10	5	Alameda	Castro Valley
W. Dublin/Pleasanton	L20	5	Alameda	Dublin
Dublin/Pleasanton	L30	5	Alameda	Pleasanton
Hayward	A60	6	Alameda	Hayward
South Hayward	A70	6	Alameda	Hayward
Union City	A80	6	Alameda	Union City
Fremont	A90	6	Alameda	Fremont
Warm Springs	S20	6	Alameda	Fremont
Montgomery St	M20	7	San Francisco	San Francisco
Powell St	M30	7	San Francisco	San Francisco
Civic Center	M40	7	San Francisco	San Francisco
Embarcadero	M16	7	San Francisco	San Francisco
16th St Mission	M50	8	San Francisco	San Francisco
24th St Mission	M60	8	San Francisco	San Francisco
Glen Park	M70	8	San Francisco	San Francisco
Balboa Park	M80	8	San Francisco	San Francisco
Daly City	M90	8	San Mateo	Daly City
Colma	W10	9	San Mateo	Colma
South San Francisco	W20	9	San Mateo	South San Francisco
San Bruno	W30	9	San Mateo	San Bruno
Millbrae	W40	9	San Mateo	Millbrae
San Francisco Airport	Y10	9	San Mateo	San Francisco