A Data-Driven Operational Model for Traffic at Dallas Fort Worth International Airport

March 23, 2021

1 Abstract

Airports are on the front line of significant innovations, allowing the movement of more people and goods faster, cheaper, and with greater convenience. As air travel continues to grow, airports will face challenges in responding to increasing passenger vehicle traffic, which leads to lower operational efficiency, poor air quality, and security concerns. This paper evaluates methods for traffic demand forecasting, which will allow airport operations staff to accurately predict traffic and congestion. Using two years of detailed data describing individual vehicle arrivals and departures, aircraft movements, and weather at Dallas-Fort Worth (DFW) International Airport, we evaluate multiple prediction methods including the Auto Regressive Integrated Moving Average (ARIMA) family of models, traditional machine learning models, and DeepAR, a modern recurrent neural network (RNN). We find that these algorithms are able to capture the diurnal trends in the surface traffic, and all do very well when predicting the next 30 minutes of demand. Longer forecast horizons are moderately effective, demonstrating the challenge of this problem and highlighting promising techniques as well as potential areas for improvement.

Traffic demand is not the only factor that contributes to terminal congestion, because temporary changes to the road network, such as a lane closure, can make benign traffic demand highly congested. Combining a demand forecast with a traffic microsimulation framework provides a complete picture of traffic and its consequences. The result is an operational intelligence platform for exploring policy changes, as well as infrastructure expansion and disruption scenarios. To demonstrate the value of this approach, we present results from a case study at DFW Airport assessing the impact of a policy change for vehicle routing in high demand scenarios. This framework can assist airports like DFW as they tackle daily operational challenges, as well as explore the integration of emerging technology and expansion of their services into long term plans.

2 Introduction

Mobility technologies are transforming the way we live and do business. Nowhere is this more evident than in the multi-modal transportation hubs–primarily airports–that connect people and move goods around the world. The increased use of smart mobility technologies at transportation hubs holds the promise of providing consumers and businesses with many benefits including increased convenience, efficiency, and resilience. However, the challenges of adapting complex transportation networks to rapidly evolving technology trends are significant; Non-optimal planning and/or execution may result in increased energy consumption, costs, and system inefficiencies.

To support this research, we have developed a partnership with Dallas-Fort Worth International Airport (DFW), the nation's first carbon-neutral airport which is simultaneously positioned in the urban region of greatest population growth [41]. DFW is the fourth busiest airport in the world by aircraft movements (takeoffs and landings) and has service to 249 destinations, including 62 international and 187 domestic destinations. The airport served a record 69,194,406 passengers in 2018. There are approximately 60,000 people that work at DFW, including airport employees, concessionaires, and others. Meanwhile, air traffic is forecast to double in the next 20 years, and DFW is committed to responding to this increase in demand by growing its capacity.

In this work, we developed an operational model for airport passenger traffic by combining traffic prediction and demand forecasting with traffic microsimulation as depicted in Figure 1. To accomplish this, (1) we predict traffic volume into the airport, and (2) we use traffic microsimulation to distribute the predicted traffic volume obtained from the first task to the airport road network. The microsimulation model represents traffic by simulating the behavior and interaction of individual vehicles. It is designed both for real-time forecasting of traffic conditions and long-term infrastructure planning. A predictive demand forecast model can assist in responding to peaks and valleys in traffic that may impact operations and security, as well as provide a platform for experimentation with the airport infrastructure and exploration of impacts in terms of fuel use, emissions, revenue, and delays. We have designed and evaluated this model with the specific task of capturing traffic and congestion at the terminal curbside in order to better understand why and where traffic events occur and how they may best be mitigated. The developed model is generalized so that it may be applied to any airport central terminal area (CTA) having basic observability of historic traffic ingress and egress, flight schedules, and weather.

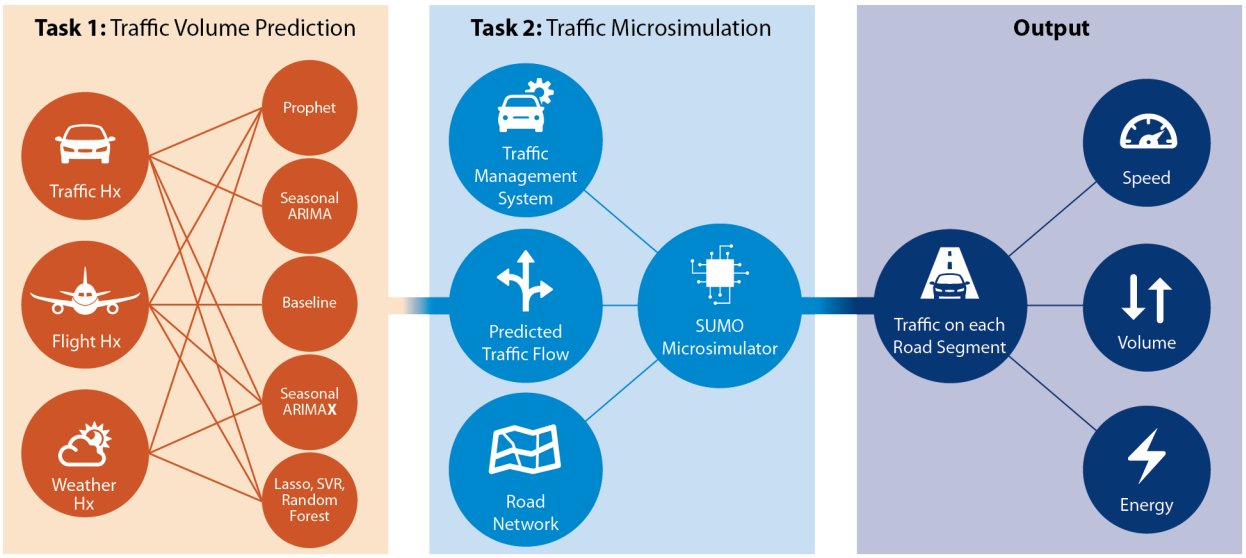


Figure 1: The operational model developed in this work operates with two major tasks: (1) traffic volume prediction and (2) traffic microsimulation. We evaluate several major volume predictive models which utilize different combinations of input features. Microsimulation utilizes predicted flow, traffic management, and road network information. The resulting data and visualizations show traffic on road segments as well as diagnostic metrics such as speed, volume, emissions, and energy usage.

Traffic flow prediction specific to airport region has rarely been studied in the literature. Davidson *et al.* [8] predicted the traffic flow in and out of the airport by using unit effect graphs to generate the distribution of the number of passengers arriving at and leaving the airport according to flight schedules. Later researchers used similar approaches to generate traffic demand at the airport, then applied discrete event simulations to further analyze curbside traffic congestion [4, 40]. Traffic prediction in general urban settings has been widely explored in literature, and abundant prediction algorithms have been discussed, from heuristics based naïve approaches such as historical averaging [25, 29] and clustering [45, 46] to various parametric and non-parametric approaches [28]. Due to the limited work on airport-specific traffic modeling, we have also investigated a broad class of methods suitable for modeling and predicting traffic and timeseries dynamics in general.

The autoregressive integrated moving average (ARIMA) model has been broadly applied in analyzing time-series data. After its first application in traffic prediction by Hamed and Al-Masaeid [14], various versions of ARIMA have been explored: seasonal ARIMA (SARIMA) that incorporates the seasonality of the time series data has been proved to generate more accurate predictions but demands much higher computational cost [48]; ARIMA with regressors (ARIMAX) can incorporate covariates to improve prediction accuracy [47]; as a less-costly alternative of SARIMA, the combination of Kohonen maps and ARIMA

(KARIMA) uses clustering to measure seasonality then applies ARIMA for each cluster [42]; and Tran *et al.* [39] combined SARIMA with generalized autoregressive conditional heteroskedasticity (GARCH) algorithm to capture the volatility of traffic flow. Some other traffic flow modeling algorithms are also essentially derived from or applications of ARIMA, such as seasonal Holt Winter's model [30], exponential smoothing [27], and ATHENA [7, 22].

Non-parametric models have also been used for traffic flow prediction, such as K-nearest neighbor (KNN), decision tree, support vector regression (SVR), and artificial neural networks (ANN) [43]. Smith [33] found that a simple implementation of KNN sometimes generates reasonably good traffic volume prediction, but it usually requires a lot of data. It has been shown that the prediction accuracy of SVR usually is not as good as SARIMA for traffic prediction as it does not address seasonality properly [44], but seasonal SVR, which introduces a seasonal kernel for SVR, is a competitive alternative when performing forecasts during the most congested periods with significantly less computational cost than SARIMA [24]. The combination of SVR and denoising algorithms is proved to help improve prediction accuracy of SVR for some practices [18, 35]. Different versions of ANN have been explored for traffic flow prediction [9], among which recurrent neural networks (RNN), especially long short term memory recurrent neural network (LSTM), has proven to be able to capture both the long-term trend and also the seasonality of traffic flow and in many cases to offer more desirable prediction accuracy than ARIMA and SVR models [1, 20, 38, 50], but it usually requires a lot of data for model training [9]. The DeepAR [32] implementation we evaluate in this paper represents this state-of-the-art LSTM-based autoregressive RNN.

The above methods focus on a one dimensional, or univariate, forecast. As of late, additional efficiencies have been discovered and utilized in the traffic prediction domain when working with multiple streams of data [49, 23]. These graph-based methods make use of the topology of the network associated with the road sensors and are able to produce even more effective models than when utilizing multiple independent univariate forecasts. The traffic data provided by DFW for this research is observed in two locations, but we combine this into a single stream of inflow demand, because we are actually interested in the traffic volume within the CTA. Future work may consider these multivariate methods if more detailed data is provided at a larger number of locations.

Traffic flow prediction based on historical data using the aforementioned approaches is usually suitable for short-term prediction, as in forecasting of the next time period, usually no more than 30 minutes. However, for long-term prediction, as in forecasting of the following day, week, or even longer, some models can suffer from error propagation [34]. Transportation agencies demand traffic prediction models that are robust for both short-term and long-term predictions and sensitive to weather conditions and special holidays [44]. Su *et al.* [34] described a method called Functional Non-parametric Regression (FNR) and showed that FNR provides superior prediction accuracy than SARIMA, neural networks (NN), and SVR for long-term predictions. The forecasting framework Prophet, which appeared recently in the literature, generates decomposition of time series data and is shown to be able to reflect trend better than SARIMA and BPNN [37]. In this work, we evaluate the most promising of these approaches in their efficacy at forecasting CTA traffic in comparison to a baseline.

As mentioned previously, traffic volume is only one of many factors that impact traffic congestion. In order to understand how traffic demand may practically impact airport operations, we utilize traffic microsimulation. Prior microsimulations for airport traffic mostly use proprietary microscopic simulation software [17] including Leigh Fisher Associates Curbside Traffic Simulation (LFACTS) model [10], Advanced Land Transportation Performance Simulation (ALPS) [21], Terminal, Roadway, and Curbside Simulation (TRACS) [15], and VISSIM [12]. In this work, we chose to use the Simulation of Urban Mobility (SUMO) microsimulator [2], an open source simulator, to enable our modeling to be replicated for other airports without requiring access to proprietary and generally expensive simulators. SUMO was previously used to model airports' ground public transportation, including buses and trains, for a multi-airport region [26]. In contrast, our SUMO model focuses on simulating traffic created by personal vehicles in and out of the DFW airport.

3 Data

In order to develop a model for airport traffic flow and its impact on curb congestion, we model the traffic demand at the curb from a given flight schedule and other exogenous data, and then use this estimate of

traffic to simulate different scenarios at the airport. Data collected from key mobility touch points throughout an airport are critical to understanding the challenges and opportunities associated with shifts in mobility technologies and provide a solid foundation for executing solutions with a high level of confidence. This section discusses the data available for modeling, what it reveals about the system under study, and how we utilize that data to design and drive our models.

Surface Traffic: The Dallas-Fort Worth International Airport (DFW) is accessed by vehicles via a control plaza on either the north or south side of the airport. This control plaza captures the traffic egress and ingress both for passenger traffic as well as some fraction of through-traffic that uses the airport tollway to bypass nearby highway traffic. Since we are interested only in the subset of vehicles that are dropping-off and picking-up passengers at the curb, we exclude all vehicles where the total time between entering and exiting the control plaza is less than eight minutes (bypass traffic) or greater than two hours (parking). Shuttle (Bus) and public transit data are also available and provide a complete picture of the passenger arrival process, however we do not use those data sources in this work, because at DFW, shuttle bus and passenger vehicles use different levels for drop-off and pick-up. We are primarily investigating the impact of passenger vehicle congestion.

A timeseries plot for the daily number of vehicles is shown at the bottom of Figure 2. Comparing this representation to the number of flights (top line of the graph), we can see that the surface traffic graph has the same general trend as the flights, but is overall smoother. Results show that the number of flights during the day at DFW has very predictable cycles.

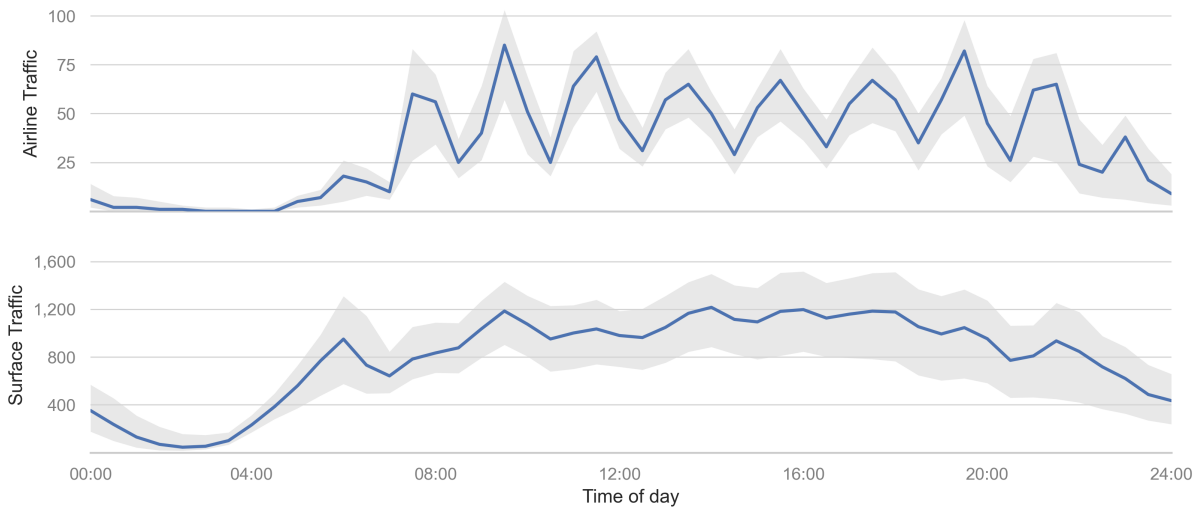


Figure 2: Timeseries plots for daily number of flights (top) and daily count of vehicles (bottom) every 30 minutes for the period between 2018-10-01 to 2019-09-30. We generate this plot by overlaying each day as a function of time. The colored region around the mean line in each graph represents the inner 90% of the data, which shows that the cycles are very consistent over time and less dramatic in the traffic data.

Airline Traffic: We obtain arriving and departing flight data from the Harris Symphony system [16]. The data contains scheduled fight times, actual flight times, type of flight (e.g., cargo or passenger), and the unique aircraft identifying N-number. Similar to prior studies utilizing the flight schedule as a proxy for estimating surface traffic [8], we assume that this data is proportional to the actual load at the airport and is an important feature in our predictive models. For modeling the vehicle traffic, we only consider passenger flights, and we use the scheduled flight times. We associate the number of seats on each unique airplane, joining on the N-number, to the flight schedule in order to estimate the maximum number of passengers expected at the DFW airport [11]. Again, the top graph in Figure 2 shows the daily distribution of flights at DFW and the very consistent cyclic pattern it has.

The flight data allows us to know the number of flights arriving and departing as well as the number of seats. We do not know how many people were on the plane, the load, or how what percent of those

people will actually use the curb as opposed to taking a connecting flight, or the origin-destination number. While we could estimate these values using the average passenger load at DFW of 82% from the Bureau of Transportation Statistics [3] and an origin-destination of around 38%, we acknowledge that our goal is not to estimate the number of curb passengers, but rather estimate the number of vehicles that will be using the passenger drop-off and pick-up curb, and for this, the total number of flights and total number of seats are sufficient exogenous features. We also observe that some of the passengers will take a shuttle bus to the rental car center, one of the parking lots, or to another terminal. The data indicates that this represents about 19% of the passengers at the curb, adding additional support to the difficulty of trying to actually estimate curb vehicle passengers.

Exogenous Features: Weather, flight delays, traffic accidents, and other exogenous events can impact the traffic arrival pattern. In order to account for this, we include detailed, high-resolution weather information including temperature, wind speed, precipitation, and pressure in our modeling. The historic weather data for the DFW airport was also downloaded from the Harris Symphony system [16]. Weather events significant enough to cause flight delays or difficulty driving to the airport could cause changes in congestion as well as changes in the number of vehicles and passengers expected to arrive at the airport. The weather data incorporated into the modeling is historic data, which could present a challenge due to the difference between forecast weather and actual weather data. This could be a limitation of using the historic data in our modeling process and could provide us with more information for prediction than would ultimately be available in real time forecasting of congestion.

Combining surface traffic, airline information, and exogenous weather data, we are able to assemble two contiguous years' data set from October 2017 to September 2019. On average, over the last year of data, we estimate that around $38K$ vehicles enter the the control plaza *per day* to pick-up or drop-off passengers, but can be as high as $52K$ vehicles on high demand days. Over the same period, there are approximately $1,650$ average flights per day providing roughly $278K$ available seats. On the highest production day, there were $1,978$ flights and a total of $334K$ seats.

To support the modeling task, we aggregate these data into fixed-length bins. The size of the aggregation period changes the magnitude of the observed cycles throughout the day, and we believe that the resolution of the underlying model is important for accurately predicting and modeling the congestion at the curb. Specifically when we re-sample the data to a 60 minute frequency, harmful smoothing occurs which removes features that are present when we aggregate the data at the 30 minute level. Considering this trade-off, we choose to aggregate the data into 30 minute bins, which for two years gives us $35,000$ observations $(365 * 24 * 2 * 2)$.

3.1 Periodic Patterns

A key consideration in modeling airport traffic is the intrinsic periodicities in the data. To better understand these dynamics, including what frequencies are present and how they change over time, we performed a *wavelet analysis* on the traffic data. Wavelet transforms are similar to Fourier transforms because both determine frequencies in a signal, however, the wavelet analysis can also determine how these frequencies change over time [13, 5]. The graphs that we examined display which frequencies are present on the y-axis and what time they appear on the x-axis, while the coloring at a specific point shows how intensely the traffic data showed a match with that frequency at that time [36].

Figure 3 shows the month of August, with the upper graph displaying the count of vehicles every 30 minutes—the signal—and the lower graph displaying the wavelet transform of this signal using the real component of a Morlet wavelet as the analyzing wavelet. The wavelet transform also has a white parabolic line marking the cone of influence; below this curve, the analyzing wavelet is nearing the edges of the signal and may only be showing edge effects, not true periodicity.

Due to the nature of wavelet analysis, the y-axis is given as an inverse log scale in base 2, meaning that the smaller periods are shown at the top of the wavelet transform image, and the larger periods are shown at the bottom and all values are given in hours. The darkest reds and darkest blues are of the most interest, because those points display the greatest overlap with the Morlet wavelet of that frequency compared to other points. Furthermore, the dark red indicates a peak of given frequency was found in the signal, while the dark blue indicates that a trough was found.

Referring to Figure 3, the strongest periodicity signals in the data appear to occur at lower time frequen-

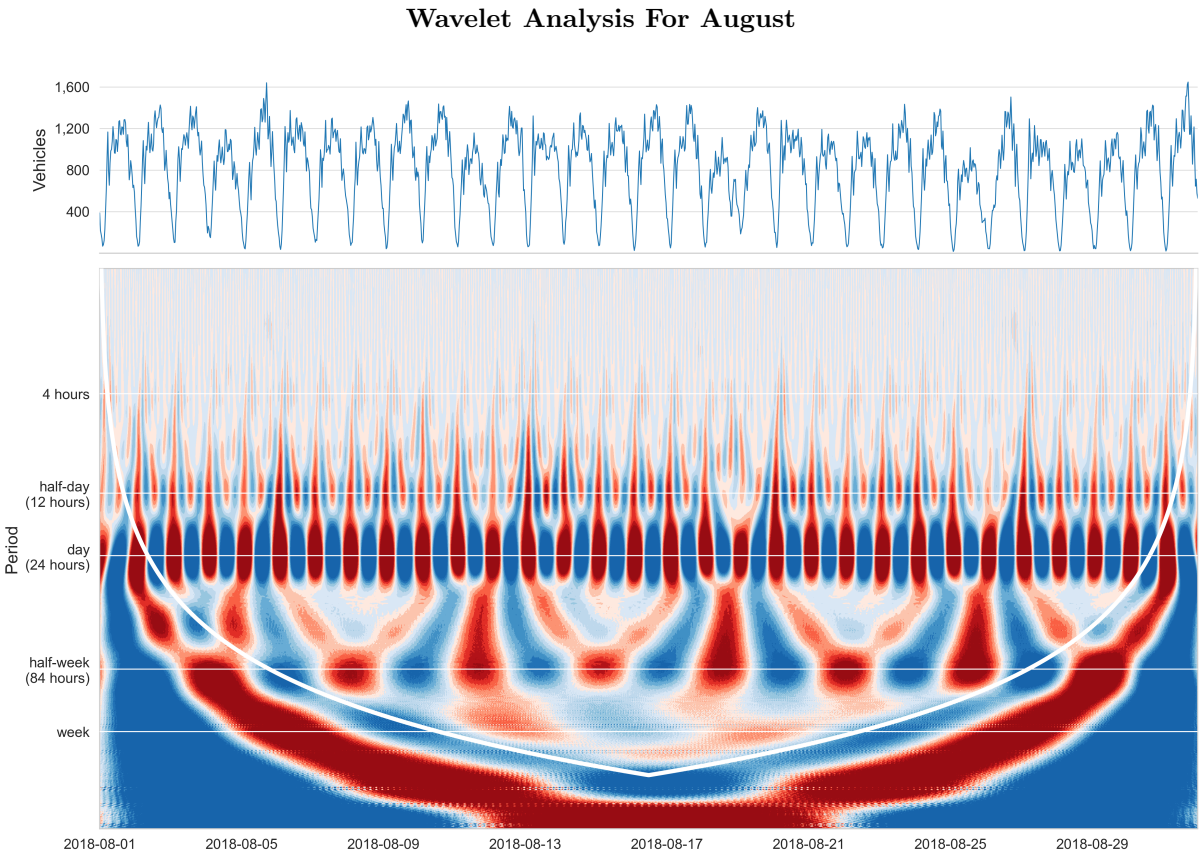


Figure 3: Wavelet Analysis on the traffic data for August 2018. The markings on the right bottom graph represent periods of interest and include strong frequencies at the day (24 hours), half-day(12 hours), and half-week (84 hours). Weaker features includes the 4 hour and 1 week areas.

cies that could be identified by the day of week or hour of day. The most pronounced signals occur daily between hours 16 and 32, or on average every 24 hours. We also observe a strong pattern near the half-day, or 12 hour mark, and some less pronounced features around the 4 hour level.

The circular-looking features, that are centered around a half-week (84 hours), move through time at this frequency, each peak and each trough is usually about 1.75 days which together form the 84 hour period. The blue troughs center around mid-week and mid-weekend while the red peaks center around Fridays and Mondays. Light colors occur above the dark blue circular features on Wednesdays, indicating that there is little periodicity between 32 and 64 hours (1.3 and 2.7 days) near midweek. But, Friday through Monday shows a stronger 1-3 day periodicity as seen by the dark red and dark blue features that extend upward in this section between 32 and 64 hours.

Based on these results, and in agreement with prior work (e.g., [44] and [24]), we choose to emphasize periodic features in our models both by using seasonal variants of moving average models (e.g., SARIMA) and by including periodic labels directly in training of non-parametric models. This wavelet analysis increases our understanding of the periodicity of the traffic data and provides a good understanding of which time features will be important. This leads us to add weekday and day-of-month variables to our machine learning features. A potential future direction could be to use the wavelet coefficients directly in the machine learning feature space to see if they better predict future traffic.

The strong autocorrelation on the daily level shows up in other statistical tests too. When testing for stationarity, using the standard Augmented Dickey-Fuller (ADF), the lags used for autocorrelation returned by the test are 48, which is a day at the 30 minute frequency we are using. The ADF test also returns a small enough p-value, much less than 0.05, that we are able to reject the null hypothesis that the data is not stationary, at least over the time frame we are considering. A seasonal decomposition using the daily

frequency also suggests a strong trend here.

4 Methods

In this section we discuss the features of our data set, the predictive demand models evaluated, and the integration between demand predictions and the SUMO microsimulation.

4.1 Data Features

Our target variable described above and displayed in Figures 1 (bottom) and 3 (top), the surface traffic, is the count of all drop-off and pick-up vehicles entering the CTA at DFW every 30 minutes. We denote this as *surface_traffic*. We have also described the total number of flights Figures 1 (top) and the total number of seats. We denote these variables as *airline_traffic* and *airline_seats*. The detailed weather data we have provides us four additional variables: temperature, humidity, pressure, and wind speed.

In this section we describe some additional features that include [1] convolution of the arrival and departure flight times to estimate terminal traversal delay, [2] date and time features to allow for modeling of monthly, weekly, and diurnal periodicities, and [3] lagged parameter values for algorithms that are not autoregressive.

Terminal Traversal Delay: One of the first observations we noticed was that the *airline_traffic* data is much more periodic than the *surface_traffic*, which is to say that they are not highly correlated. A similar pattern emerges for the *airline_seats* when we add in the number of seats on each flight in the *airline_traffic*. We reasoned that it would take a certain amount of time for passengers to get to the curb once they arrived, and in the same way, departing passengers will arrive to the airport early for a departing flight. To address this, we performed a grid search over possible distributions for both arrivals and departures and then analyzed the correlation of the resulting flight and traffic data. We refer to this optimal change as the *adjusted_airline_seats*, because we use the number of seats on each flight as a basis for our calculated delay. We found the optimal delay for arrivals and departures to be 40 minutes and 110 minutes respectively. We reason that this makes sense because it takes around 40 minutes to get to the curb for arrival flights and passengers tend to arrive about 110 minutes before their departing flights. While this is reasonable in terms of what might physically happen at the airport, it is also acceptable to apply this transformation simply because it improves the correlation with the *surface_traffic* data. The highly cyclic *airline_traffic* may actually reduce curb congestion given these estimates of terminal traversal.

Periodic Markers: The time of day provides several periodic features that may be useful in predicting *surface_traffic*. There are at least two ways to encode this information. The first method, *one-hot encoding*, creates a categorical feature for each value. These methods can increase the dimensionality of the feature space and also may not capture the cyclic nature of the data. The second method, *sine-cosine encoding*, maps each value to a sine and cosine value so that the beginning and end of the period line up. For this work, we considered the following features: year, month, day, week, day of week, hour, and half-hour and encode with the one-hot method. These features are most relevant for algorithms that do not learn a periodic encoding.

Lagged features: The group of classic machine learning algorithms—linear regression, SVR, and XGBoost—are not autoregressive and will benefit from lagged features. For the purpose of this evaluation, we lag the *surface_traffic*, *airline_seats*, and *adjusted_airline_seats*. When forecasting the next interval, we assume these lagged parameters will be available, but because these algorithms also model each observation independently, we do not include the *surface_traffic* when forecasting longer time horizons. In other words, unlike the data coming from flight schedules, the *airline_traffic*, *airline_seats*, and *adjusted_airline_seats*, which are features that are known in advance, we can only use the lagged *surface_traffic* for the next forecast (e.g. 30 minutes).

In summary, the data features available for predicting *surface_traffic* are: *airline_traffic*, *airline_seats*, *adjusted_airline_seats*, *year*, *month*, *day*, *week*, *day_of_week*, *hour*, *half_hour*, *temperature*, *pressure*, *humidity*, and *wind_speed*, as well as our lagged values for *surface_traffic*, *airline_seats*, and *adjusted_airline_seats*.

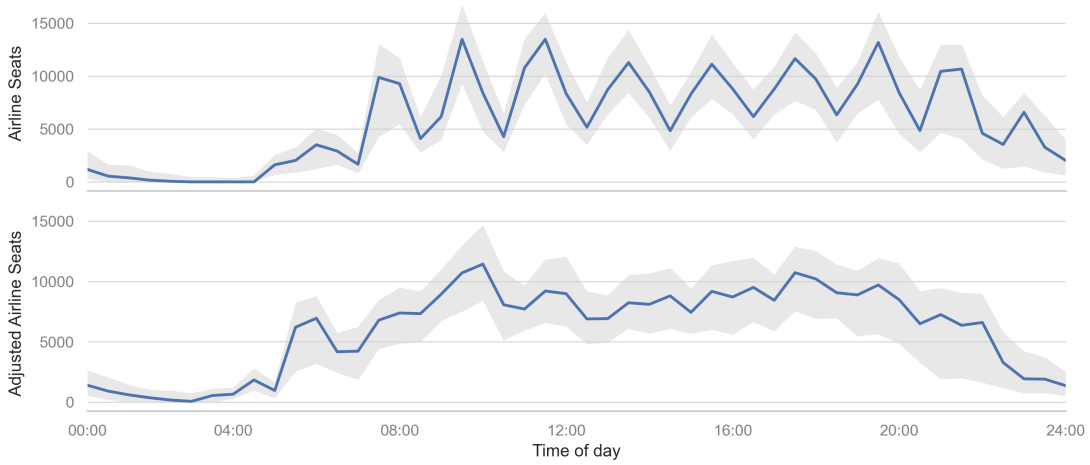


Figure 4: A grid search over different possible distribution shifts applied to the arriving and departing *maximum passenger* data shows that we can increase this correlation by shifting the arrival flights forward by 40 minutes and shifting the departing flights back by 110 minutes. Comparing the bottom figure with the traffic counts in Figure 2 it is more clear that this shift smooths out some of the cyclic patterns that exists in the flight data. This transformation can be applied to future flight schedules as well.

4.2 Feature Importance

There are several ways to identify which features will likely be the most effective in predicting the *surface_traffic* and we focus on three of them: correlation, mutual information statistics [31], and feature importance from boosted decision trees. The first method looks at the correlation between the target and a feature with the assumption that a higher positive correlation will be a good predictor of the target value. This is the same reasoning we used to justify the creation of our *adjusted_airline_seats* parameter. Mutual information statistics calculate the information gain, or reduction of uncertainty, that each variable has on the target. Finally, using a boosted decision tree, like XGBoost, will report on which features were most important in learning the target parameter values.

We computed these statistics across our validation sets (described below) and normalized the values so that they would be more comparable. The non-lagged parameters with the highest values across all three tests were: *adjusted_airline_seats*, *hour*, *half_hour* , *airline_traffic*, *airline_seats*, *humidity*, and *temperature*. The lagged parameters with the highest values were *surface_traffic* and *adjusted_airline_seats*.

4.3 Model Fitting and Validation

With a potential set of features identified we are now able to determine which combination of model and features provides the best modeling for predicting *surface_traffic*. This, of course, also depends on the algorithms we deploy, the hyper-parameters used, and the periods for which we forecast. In this section, we describe the methodology used for identifying an optimal subset of features, the metrics we use for measuring performance, and the algorithms used.

Baseline: The baseline model we use is a simple univariate linear model that predicts *surface_traffic* from the *airline_traffic* and *airline_seats* data. This method is in the spirit of the model described by Davidson in 1969 [8] and is not expected to be effective, but rather provide perspective for the methods described below. This baseline linear model does not use any lagged values or weather data.

SARIMAX: The SARIMAX model is based on the ARIMA model with the added capability to include seasonality and exogenous parameters. The ARIMA model is a combination of an auto regressive model that produces a forecast based on the history of the target values and a moving average of the historical values. The parameters of the ARMIMA model specify the lags used in the auto regressive component (p), the degree of differencing required (d), and the number of moving average terms (q). The seasonal version

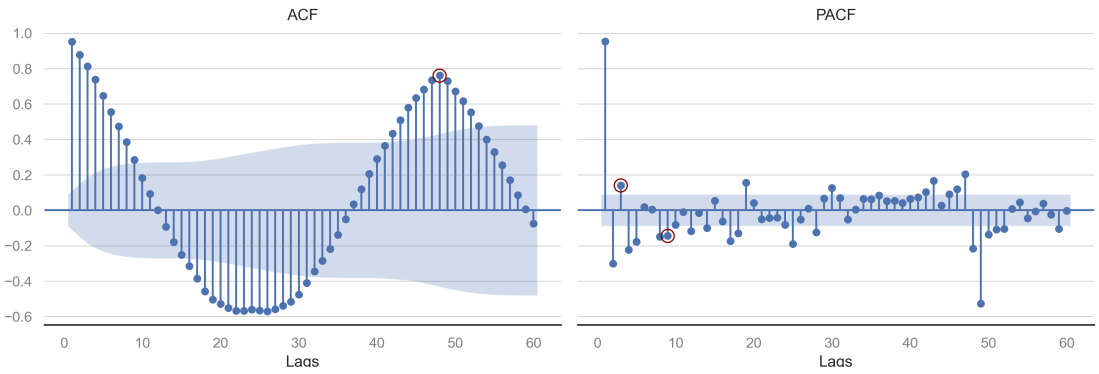


Figure 5: The ACF and PACF plots for the surface traffic. The ACF plot shows shows a peak significant seasonality of 48 lags, which is a single day with our re-sampling rate. The PACF plot suggests values $p = 2$ and $q = 8$ as a starting point for evaluation.

repeats these parameters and also has a term that specifies the length of the seasonal component (s). Decent starting values for p and q can be obtained by the partial autocorrelation function (PACF) that determines the influence of the lag on the target value. The graphs in figure 5 suggests a (2,0,8) model as a starting point for ARIMA and the autocorrelation function (ACF) confirms the seasonality parameter of 48 that we have presented evidence for previously.

Machine Learning: The problem of predicting traffic from a well-structured feature set is a supervised regression problem. We evaluated three classic regression algorithms on our data: multiple linear regression, Support Vector Regression (SVR), and XGBoost [6]. Lasso is a linear model that constrains the coefficients by adding a penalty to the cost function. This helps prevent over-fitting and can help in feature selection [19]. The SVR algorithm is an extension to the Support Vector Machine (SVM) that allows for a qualitative prediction by modifying the definition of the margin used in the cost function [19]. Like the SVM, it uses a kernel to account for non-linear feature interaction. XGBoost is a parallel gradient boosting decision tree implementation that aggregates the performance of several decision trees by limiting the number of predictors each tree can utilize [6].

Each of these methods have hyper parameters that can impact their efficiency. In each instance, we tune the parameters by varying them over a set of reasonable possibilities and evaluating how well each set performed across our test validation sets.

DeepAR: The DeepAR [32] algorithm produces a probabilistic forecast based on training an auto regressive LSTM-based recurrent neural network. As with other methods, exogenous features can be used to potentially improve the forecast as long as they are also available during the prediction horizon. While multiple streams of target variables may be used, we are only utilizing DeepAR for a single target value, the *surface_traffic*. One of the advantages of DeepAR is that it learns encodings for times series at the granularity that is most effective so the periodicities we discussed earlier do not need to be discovered or included. As with the other models, DeepAR has a set of hyper parameters that can impact its behavior, such as learning rate, number of epochs, and the mini-batch size, and we vary these over a reasonable set of values to tune the parameters to this problem and evaluate them on the test validation sets.

We are primarily interested in evaluating how well or poorly each model predicts the traffic across three forecast periods: 30 minutes, 1 day, and 1 week. We refer to the 30 minute forecast as the **observation** forecast, because it is the frequency of our input data set and is, therefore, a measure of how well our models predict each observation. We split our data into training and test sets for cross-validation by selecting three full days that represented a high, medium, and low demand. We did this by selecting the six month period between April and September of 2019 and calculating the cumulative sum of surface traffic each day, and then used the 85% percentile for the high demand day, 50% percentile for the medium demand day, and the 15% percentile as the low demand day. Figure 6 shows the three days highlighted against a backdrop of all the days during this period. The grey background shows the spread of all the data.

These three days do not represent the average behavior that we would expect, but rather indicate how well the algorithms will perform across different demands. We also evaluate the single day forecast on a

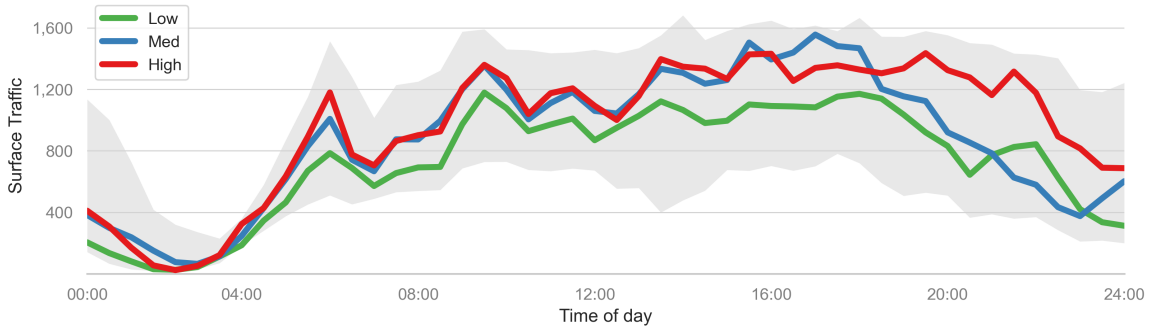


Figure 6: The three days chosen for cross-validation. The low, medium, and high refer to the level of traffic that occurred during the day for a six month period during 2019.

random sample of 30 days using the best hyper-parameters learned on these three cross validation days. This larger test gives us a better indication of the expected behavior for our algorithms.

The metric we used to identify effectiveness is the Root Mean Squared Error (RMSE) of the predicted traffic from the *hold-out test* set, where a lower value is best. For the 30 minute prediction, we walk forward for each day, resulting in a total of 144 (3*48) training runs for each algorithm and 144 comparisons for the RMSE calculation. The one day prediction also results in the same 144 comparison values for the RMSE, but only requires 3 training runs for the ARIMA family of algorithms and the DeepAR algorithm. The simple machine learning models are still evaluated on each observation, but as mentioned earlier are only allowed lag variables that would be available (e.g. they can not use the lagged *surface_traffic*). Finally, the 1 week prediction uses the three cross-validation days as a starting point and forecasts 336 observations ($3 * 48$) for a total of $1,008$ comparisons in the RMSE. For discussion, we also compute the Mean Absolute Percent Error (MAPE) for each of the tests we run.

4.4 Integration with Microsimulation

The second goal we have is to integrate the model prediction into a microsimulation of the DFW airport to understand how events, such as an increase in scheduled flights or a lane closure, will impact the congestion at the curb. The traffic forecast is the expected number of passenger vehicles at *all* the terminals for a given 30 minute period of time. For the microsimulation to be useful, we need to split this forecast into groups whose destination is one of five terminals: A, B, C, D, or E. We estimate this by using the distribution of flights that utilize each terminal and then make the assumption that the traffic obeys a similar distribution. This distribution is based on a sample of data that is aggregated and not available at the detail that would allow us to use as part of the prediction process.

We built the SUMO microsimulation model from a road network and a demand model comprised of various route trips. For the network, we extracted the DFW airport from OpenStreetMap. The network captures various attributes of the airport geometries, and validation of this network is necessary for accurate results. For the demand side of the model, we utilized predicted traffic demand forecasts from the fitted predictive model.

We made several assumptions during the construction of the vehicle trips. Since the predictive model does not account for explicit control plaza entrance and exits of each vehicle, we used entrance and exit distributions derived from the control plaza data and sampled from the observed daily distribution to create a trip. For example, 55% of trips entered through the north control plaza with 80% of the trips exiting the same control plaza from which they entered. The only exception was for pass-through traffic which exited the opposite control plaza 80% of the time. For future work, we plan on breaking up these distributions temporally (e.g. hourly) and generalizing our assumptions based on daily, monthly, and yearly sampling for improved likelihoods. We also assumed curbside pickup and drop-off dwell times of 120 seconds on average sampling from a discrete distribution. DFW airport records the actual dwell time distributions, and our future work will integrate these observations.

SUMO is capable of various outputs, which include but are not limited to: emissions, fuel consumption, trip duration, speed, relative speed, detailed trajectories, and waiting times due to congestion. we can usee these outputs when simulating future demand to determine if there will be increase congestion in the CTA. For example, in addition to waiting times, if the trip duration increases and the speeds decrease, there is reasonable evidence to assume we have some increased road congestion. Furthermore, we can visualize where in the CTA this is occurring and replay the events with different policies to see how best to mitigate a congested scenario.

In summary, the SUMO microsimulation takes as input a demand schedule for every 30 minute period during the day and creates individual vehicles in SUMO that have a terminal destination that follows the wait time distribution from the surface traffic data. This model does not need to be linked to a demand forecast to be useful. Operators could replay a particular high demand day to explore policies that would have help congestion, or they might want to simulate a rough estimate of what they believe traffic will be like in a year on a high demand day.

5 Results

We executed several hundred runs for each algorithm on our three cross-validation days and across the three prediction horizons, which were 1, 48, and 336 steps, in order to tune the hyper parameters and find the most effective solutions. Table 1 shows the overall results for the best configuration for each algorithm across the three prediction horizons for both the RMSE (1a) and the MAPE (1b). At a high level, both the *Baseline* and *ARIMAX* algorithms were less effective than the other four algorithms, but the SARIMAX algorithm performed better than with a seasonal lag of 48, which is one day at our sampling rate. The best prediction was DeepAR for the 1 and 48 steps level. SVR performed best at the weekly forecast (336 steps) when measured by RMSE, but XGBoost had the lowest MAPE.

prediction_length algorithm	1 (30 min)	48 (1 day)	336 (1 week)
Baseline	270.0	270.0	272.0
ARIMA	112.0	254.0	307.0
DeepAR	**67.0**	**103.0**	151.0
Linear	72.0	133.0	162.0
SARIMAX	71.0	137.0	185.0
SVR	75.0	137.0	**147.0**
XGBoost	74.0	135.0	156.0

(a) RMSE

prediction_length algorithm	1 (30 min)	48 (1 day)	336 (1 week)
Baseline	0.90	0.90	0.72
ARIMA	0.22	0.78	1.01
DeepAR	**0.08**	**0.16**	0.20
Linear	0.12	0.26	0.24
SARIMAX	0.11	0.24	0.24
SVR	0.14	0.32	0.30
XGBoost	0.11	**0.16**	**0.18**

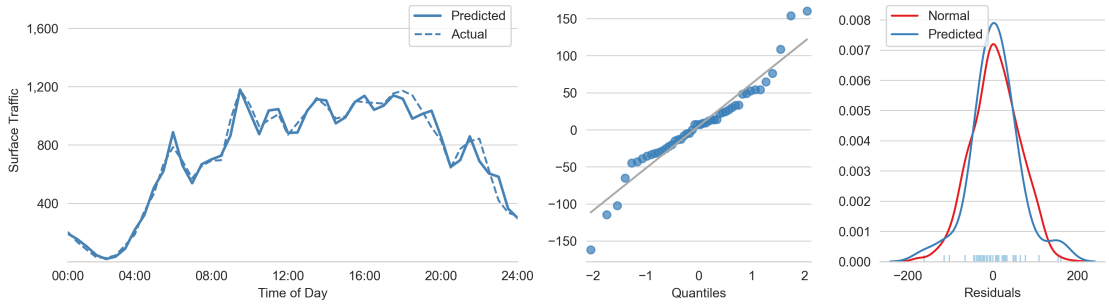
(b) MAPE

Table 1: RMSE and MAPE

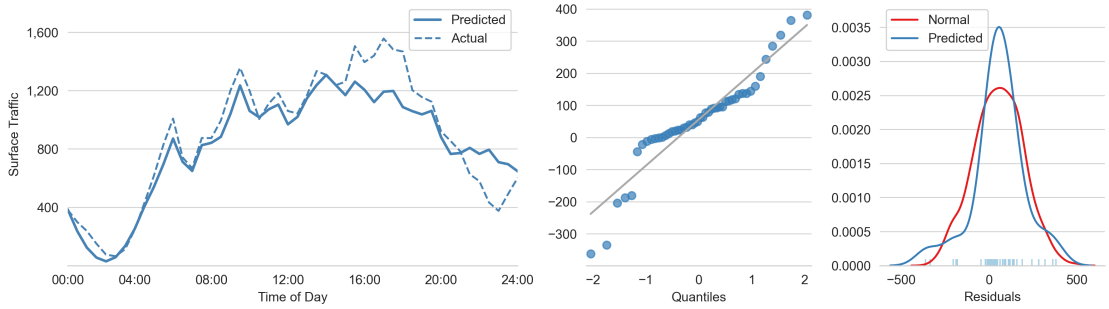
When predicting the next 30 minutes of surface traffic, the top five algorithms: DeepAR, SARIMAX, linear, SVR, and XGBoost all preformed well. Figure 7a shows the results of DeepAR on the low demand cross-validation day using a walk forward prediction of 30 minutes. The right-most graphs in Figure 7a show the residual quantiles as a function of normal quantiles (middle graph) and the distribution of the residuals in the far right graph. The residuals are fairly normal with a slight skew and some evidence of heavier tails. The left-most graph in Figure 8 shows the performance of DeepAR, SARIMAX, SVR, and XGBoost broken down by cross-validation day. We observe that the low demand day is easiest for all algorithms and that they all perform well on this set of days for this time horizon of 30 minutes, with DeepAR performing the best.

The day and week predictions are not as effective or as consistent. Figure 8 shows the RMSE for the top algorithms based on the cross-validation day. Here we observe that the the classic machine learning algorithms SVR and XGBoost appear to be more consistent; that is, they exhibit less variation in their RMSE values but do not find the most effective solutions. Figure 7b shows the medium demand day for DeepAR which is its least effective cross-validation day. The left-most graph shows how DeepAR misses a

critical peak between 16 and 20 hours and then under-predicts demand between hours 20 and 24. This is reflected in the longer tails of the residual distribution plots.



(a) A walk forward prediction (1 step ahead) for DeepAR on the Low day.



(b) The full day prediction (48 steps ahead) for DeepAR on the medium day, it's worst performance.

Figure 7: **DeepAR results**: The top graph 7a shows an effective 30 minute forecast. As the forecast period gets longer, DeepAR is less effective, as shown in the bottom graph.

In order to explore how the degree of consistency among normal days may lead to forecasting errors on abnormal days, we selected 30 days at random from the last year of our data, re-trained, and tested our top algorithms on these new cross validation days with the hypothesis that the further away a day is from the average day, the more difficult it would be to predict. The day that most closely resembles the average demand for the last year of data we have is August 13, 2019. It is the day the differs the least from the computed mean demand during the FY19 year. Each of the 30 days were assigned a demand value based on their total demand; if the day's demand was 5,000 more than the average day we categorized it as a high demand day. Similarly, if the demand for the day was 5,000 lower than the average demand day we categorized it as a low demand day. All other days are considered and denoted *medium* demand. The 10,000 vehicle range we used is about 20% of the average total vehicle demand.

The left-most graph in Figure 9 shows the RMSE distributions for all solutions based on total daily demand. As can be seen in the figure, low and high distributions include much less effective forecasts and their standard deviations are about twice that of the medium distribution. Statistically they are not the same distributions. The right-most graph in Figure 9 shows the RMSE values as a function of the distance from each cross validation days to the average day measured as RMSE. The dashed 45 degree line represents how well each cross validation day would have performed if we used the average day as a prediction. This means that when algorithms fall below this line, they out performed this naive forecast. This graph also suggests that days with demand curves that are increasingly different from the average day are more difficult to predict. These results suggests that long-term forecasts pose a greater challenge compared to short term forecasts, particularly for high demand days. DeepAR has the lowest RMSE for the 30 samples, and although the standard deviation of the algorithms tested is similar, its results are not statistically significant.

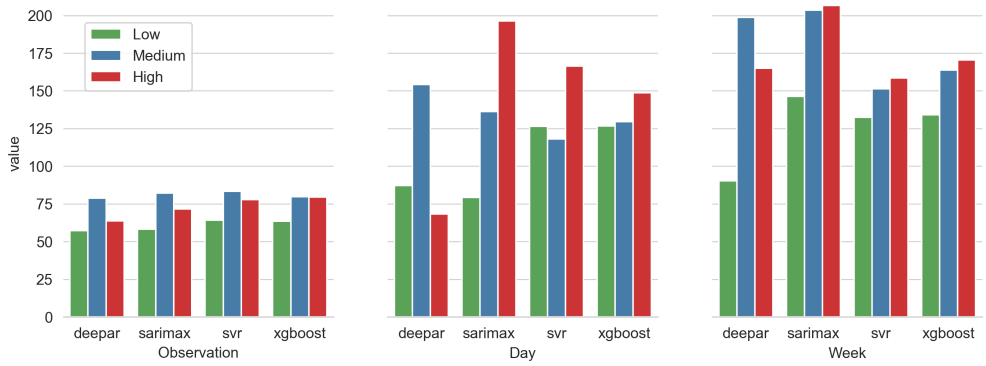


Figure 8: A graphical version of the RMSE results from Table 1a broken down by cross-validation day.

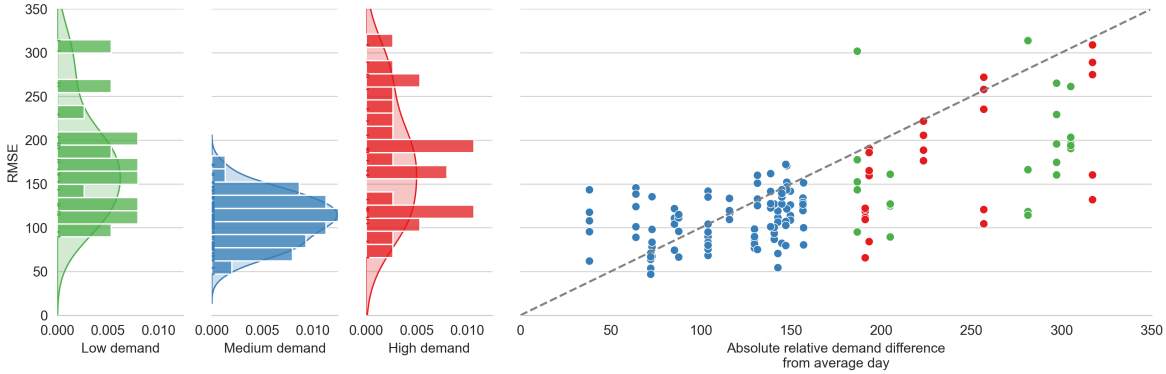
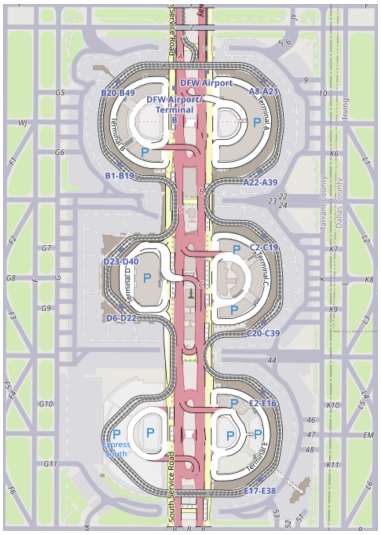
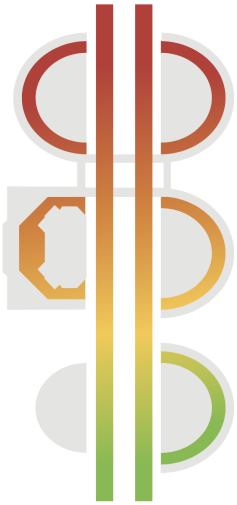


Figure 9: The left graph shows the significantly different RMSE distributions for each of category of demand: low, medium, and high. The right graph shows the same RMSE values but as a function of distance, which we measure as the RMSE, from the mean day of the last year of data. High and low demand days are further from the mean day in terms of shape and are more difficult to forecast.

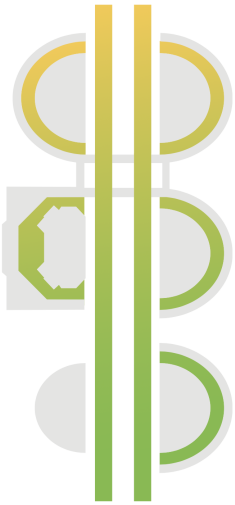
6 Case Study



(a) The Open Street Map representation of DFW.



(b) A high demand day with no intervention.



(c) The same demand day with a simple intervention policy.

Figure 10: **Case Study**: The SUMO microsimulation is based on the Open Street Map of DFW 10a. Ideally we would use the demand forecast to create a prediction of the number of vehicles entering the CTA at 30 minute intervals. The high demand period we simulated creates a network that at one point will experience congestion as seen in 10b. Applying a simple policy that diverts some of the traffic to other terminals eliminates the heavy congestion seen in 10c.

To present how microsimulations may be used in practice with a demand prediction model, we developed a case study to demonstrate how different traffic control strategies influence congestion and energy consumption for high-volume days. We simulated a high-volume day (Monday, June 11th, 2018) with 20% additional demand resulting in a total of around 72,000 vehicles traveling to or from the airport during the 48 half-hour periods during the day. We considered two different control policies for trips in/out of the airport: *intervention with policy* and *no intervention*, with vehicles taking the shortest path to their assigned destination. The *no intervention* policy assumes all vehicles take the shortest path calculated at their departure time for their trip, without accounting for predicted changing traffic conditions. The *intervention with policy* strategy assumes departure vehicles are informed of anticipated congestion at their destination terminal using, for example, our predictive model. Accordingly, they are advised to use another terminal or parking lot to be dropped off to mitigate congestion. The intervention policy is implemented with the assumption that 50% of people will obey the recommendation. For an airport like DFW where passengers can easily travel between terminals, this is a realistic policy.

The results of the two scenarios are shown in Figure 10. Looking at the results from SUMO, we notice that the *intervention with policy* performs the best during peak demand on the network. With further investigation, we found that during the peak hour from 6:00am to 7:00am, the policy saved 34.4% of fuel consumption compared to no intervention. Additionally, the *intervention with policy* saved each trip 4.47 minutes per trip on average during the peak hour. That equates to a 38.9% savings on trip duration at the airport during that high demand period. Other notable findings in the results showed how *mean relative speed* was affected positively by the policy. Mean relative speed is the ratio of observed average speed over posted speed. Thus, lower values can be used as proxies for congestion, and values near 1.0 can be considered free-flowing traffic. Lastly, we observed from the *mean travel time* divergence that occurs in the morning peak demand. This divergence is also reflected in the *vehicles currently running* by showing that the *no intervention* policy dissipates the volume demand slower than the *intervention with policy*.

These scenarios are introductory inquiries into future work on curbside congestion management. The traffic demand and curbside dynamics still need to be calibrated with ground truth observations to get more

realistic results. In future work, we plan to use the SUMO microscopic model and demand framework to study the impact of airport infrastructure modifications on curbside congestion.

7 Conclusion

Here we have presented a first of its kind, data-driven operational model for airport traffic combining a demand forecast with a microsimulation of the CTA. Our hope is that the airport operations team at DFW could use this framework in the future to both forecast near term and medium term demand in the CTA in real-time and understand the implications on the current state of the road network. This framework could assist airports like DFW as they tackle daily operational challenges, explore the integration of emerging technology, and plan the expansion of their services in the long term. Operators of our model could simulate novel scenarios to explore potential policy and infrastructure changes by replaying high demand periods, or even increasing demand as we did in the case study.

We have shown that several models are capable of capturing the strong daily trend observed in the data at DFW airport and are effective in predicting traffic during the next 30 minutes. Forecasts for the one day and single week ahead pose more challenge for the models, and there may be room for improvement to achieve high fidelity results on these time scales. The highest and lowest demand days are the most difficult to forecast for the models we tested and more work is needed in this area to produce highly accurate longer-term demand forecasts around extremes. A key motivation for this work is to develop a universal framework for operational modeling of traffic demand at the airport complex that could be used by any airport. In future work we intend to build out this capability as part of a production system available to DFW and other airports that would like to participate. Finally, the code we used for the experiments in this paper, as well as the two full years of surface traffic and weather data, are publicly available at *removed_for_anonymity*.

References

[1] U. Ali and T. Mahmood. Using Deep Learning to Predict Short Term Traffic Flow: A Systematic Literature Review. In *Intelligent Transport Systems From Research and Development to the Market Uptake*, pages 90–101. Springer International Publishing, 2018.

[2] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz. Sumo–simulation of urban mobility: an overview. In *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind, 2011.

[3] Bureau of Transportation Statistics. Airlines and airports. load factor, 2019. `https://www.transtats.bts.gov/`.

[4] M. C. v. Burgsteden, P. E. Joustra, M. R. Bouwman, and M. Hullegie. Modeling road traffic on airport premises. In *2000 Winter Simulation Conference Proceedings (Cat. No.00CH37165)*, volume 2, pages 1154–1163 vol.2, Dec. 2000.

[5] B. Cazelles, M. Chaves, and D. Berteaux. Wavelet analysis of ecological time series. *Population Ecology*, pages 287–304, 2008.

[6] T. Chen, T. He, M. Benesty, V. Khotilovich, and Y. Tang. Xgboost: extreme gradient boosting. *R package version 0.4-2*, pages 1–4, 2015.

[7] M. Danech-Pajouh and M. Aron. ATHENA: A method for short-term inter-urban motorway traffic forecasting. 6:11–16, Jan. 1991.

[8] K. B. Davidson, G. C. Martin, and A. J. Morton. A traffic prediction model for Brisbane airport. *Australian Road Research*, 3(10), June 1969.

[9] L. N. N. Do, N. Taherifar, and H. L. Vu. Survey of neural network-based models for short-term traffic state prediction. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(1):e1285, 2019.

[10] G. Duncan and H. Johnson. Development and application of a dynamic simulation model for airport curbsides. In *The 2020 Vision of Air Transportation: Emerging Issues and Innovative Solutions*, pages 153–164. 2000.

[11] Federal Aviation Administration Aircraft Inquiry". N-number data, 2019. `https://registry.faa.gov/aircraftinquiry/NNum_Inquiry.aspx`.

[12] M. Fellendorf and P. Vortisch. Microscopic traffic flow simulator vissim. In *Fundamentals of traffic simulation*, pages 63–93. Springer, 2010.

[13] A. Graps. An introduction to wavelets. *IEEE Computational Science and Engineering*, 1995.

[14] Hamed Mohammad M., Al-Masaeid Hashem R., and Said Zahi M. Bani. Short-Term Prediction of Traffic Volume in Urban Arterials. *Journal of Transportation Engineering*, 121(3):249–254, May 1995.

[15] B. Hargrove and E. Miller. Tracs: Terminal, roadway, and curbside simulation: a total airport landside operations analysis tool. *Transportation Research Circular*, 2002.

[16] Harris. Harris symphony dataset, 2019. `https://www.harris.com`.

[17] T. M. Harris, M. Nourinejad, and M. J. Roorda. A mesoscopic simulation model for airport curbside management. *Journal of Advanced Transportation*, 2017, 2017.

[18] W.-C. Hong. Application of seasonal SVR with chaotic immune algorithm in traffic flow forecasting. *Neural Computing and Applications*, 21(3):583–593, Apr. 2012.

[19] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[20] D. Kang, Y. Lv, and Y. Chen. Short-term traffic flow prediction with LSTM recurrent neural network. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, Oct. 2017.

[21] Kimley-Horn and A. Inc. Transit consulting services. https://www.kimley-horn.com/service/transit-consulting-services/, 2019. Accessed: 2019-07-09.

[22] H. R. Kirby, S. M. Watson, and M. S. Dougherty. Should we use neural networks or statistical models for short-term motorway traffic forecasting? *International Journal of Forecasting*, 13(1):43–50, Mar. 1997.

[23] Y. Li, R. Yu, C. Shahabi, and Y. Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.

[24] M. Lippi, M. Bertini, and P. Frasconi. Short-Term Traffic Flow Forecasting: An Experimental Comparison of Time-Series Analysis and Supervised Learning. *IEEE Transactions on Intelligent Transportation Systems*, 14(2):871–882, June 2013.

[25] D. Nikovski, N. Nishiuma, Y. Goto, and H. Kumazawa. Univariate short-term prediction of road travel times. In *Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.*, pages 1074–1079, Sept. 2005.

[26] U. Noyer, F. Rudolph, and M. Jung. Simulating a multi-airport region on different abstraction levels by coupling several simulations. *EPiC Series in Engineering*, 2:14–24, 2018.

[27] M. O'Mahony, B. Ghosh, and B. Basu. Time-series modelling for forecasting vehicular traffic flow in Dublin. 2005.

[28] C. P Ij Van Hinsbergen, J. Lint, and F. M Sanders. Short Term Traffic Prediction Models. *14th World Congress on Intelligent Transport Systems, ITS 2007*, 7, Nov. 2007.

[29] D. Park and L. R. Rilett. FORECASTING MULTIPLE-PERIOD FREEWAY LINK TRAVEL TIMES USING MODULAR NEURAL NETWORKS. *Transportation Research Record*, (1617), 1998.

[30] A. R. Raikwar, R. R. Sadawarte, R. G. More, R. S. Gunjal, P. N. Mahalle, and P. N. Railkar. Long-Term and Short-Term Traffic Forecasting Using Holt-Winters Method: A Comparability Approach with Comparable Data in Multiple Seasons. *Int. J. Synth. Emot.*, 8(2):38–50, July 2017.

[31] B. C. Ross. Mutual information between discrete and continuous data sets. *PloS one*, 9(2):e87357, 2014.

[32] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

[33] B. Smith. Forecasting freeway traffic flow for intelligent transportation systems application. Jan. 1995.

[34] F. Su, H. Dong, L. Jia, Y. Qin, and Z. Tian. Long-term forecasting oriented to urban expressway traffic situation. *Advances in Mechanical Engineering*, 8(1):1687814016628397, Jan. 2016.

[35] J. Tang, X. Chen, Z. Hu, F. Zong, C. Han, and L. Li. Traffic flow prediction based on combination of support vector machine and data denoising schemes. *Physica A: Statistical Mechanics and its Applications*, Mar. 2019.

[36] A. Taspinar. A guide for using the wavelet transform in machine learning, dec 2018.

[37] S. J. Taylor and B. Letham. Forecasting at scale. Technical Report e3190v2, PeerJ Inc., Sept. 2017.

[38] Y. Tian and L. 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, Dec. 2015.

[39] Q. T. Tran, Z. Ma, H. Li, L. Hao, and Q. K. Trinh. A Multiplicative Seasonal ARIMA/GARCH Model in EVN Traffic Prediction. *International Journal of Communications, Network and System Sciences*, 8(4):43–49, Apr. 2015.

[40] C. Tunasar, G. Bender, and H. Young. Modeling curbside vehicular traffic at airports. In *1998 Winter Simulation Conference. Proceedings (Cat. No.98CH36274)*, volume 2, pages 1113–1117 vol.2, Dec. 1998.

[41] U.S. Census Bureau. New census bureau population estimates show dallas-fort worth-arlington has largest growth in the united states, March 2018. `https://www.census.gov/newsroom/press-releases/2018/popest-metro-county.html`.

[42] M. Van Der Voort, M. Dougherty, and S. Watson. Combining kohonen maps with arima time series models to forecast traffic flow. *Transportation Research Part C: Emerging Technologies*, 4(5):307–318, Oct. 1996.

[43] E. I. Vlahogianni, J. C. Golias, and M. G. Karlaftis. Short-term traffic forecasting: Overview of objectives and methods. *Transport Reviews*, 24(5):533–557, Sept. 2004.

[44] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias. Short-term traffic forecasting: Where we are and where we're going. *Transportation Research Part C: Emerging Technologies*, 43:3–19, June 2014.

[45] W. Weijermars and E. v. Berkum. Analyzing highway flow patterns using cluster analysis. In *Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.*, pages 308–313, Sept. 2005.

[46] D. Wild. Short-term forecasting based on a transformation and classification of traffic volume time series. *International Journal of Forecasting*, 13(1):63–72, Mar. 1997.

[47] B. M. Williams. Multivariate Vehicular Traffic Flow Prediction: Evaluation of ARIMAX Modeling. *Transportation Research Record*, page 7, 2001.

[48] B. M. Williams and L. A. Hoel. Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. *Journal of transportation engineering*, 129(6):664–672, 2003.

[49] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang. Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*, 2019.

[50] B. Yang, S. Sun, J. Li, X. Lin, and Y. Tian. Traffic flow prediction using LSTM with feature enhancement. *Neurocomputing*, 332:320–327, Mar. 2019.