A Neural-Queuing Approach to Modeling Airport Surface Traffic at Charlotte-Douglas International Airport

Liam McKenna*, Kenny Chour†, Milad Memarzadeh‡, and Krishna Kalyanam§
NASA Ames Research Center, Moffett Field, CA 94035, USA

Rajnikant Sharma¶
Air Force Institute of Technology, Wright-Patterson AFB, OH 45433, USA

In the pursuit of more powerful analysis tools for Air Traffic Management, training and evaluation of tool effectiveness is typically performed within the context of a simulated system that can adequately replicate the complexity of the National Airspace System. The ground operations of airports is of particular concern for approaches to improving congestion and traffic flow, but due to complex aircraft taxi procedures and variation of air traffic playbooks, forecasting is difficult. This work outlines a framework to aggregate and predict complex airport surface traffic in the context of optimizing air traffic flow and ground operations. In our approach, an Artificial Neural Network is trained on historical data from Charlotte-Douglas International Airport. The Neural Network then informs a Queueing Network for controlling aircraft traffic flow through various congestion points on the tarmac. The resulting system is capable of forecasting arrival and departure traffic over a long time horizon (e.g. 365 days). The simulation environment then provides a basis for deriving useful performance metrics which could be used as feedback to optimization techniques. The output of the surface model we develop shows high prediction accuracy compared to the historical service rates, queue lengths, and taxi times.

I. Introduction

Airports are among busiest locations in the world, serving millions of flights annually. The airport's surface is often congested due to long queues of aircraft at runways or terminal locations, leading to increased fuel burn, delays, and emissions [1, 2]. Therefore, it is imperative to accurately predict traffic flow behavior to analyze congestion mitigation strategies. In runway configuration management (RCM), the goal is to optimally select the flow configuration of an airport to maximize traffic flow. Various methods for performing RCM include optimization [3], heuristics [4], and Reinforcement Learning (RL) approaches [5], but validating their efficacy is challenging. Thus, there is a need to fill this validation gap with an efficient traffic flow modeling and simulation framework.

Traditional methods of capturing airport traffic flow include stochastic queue modeling [6] or microsimulations (such as commercially available tools like SIMMOD[7]). Stochastic queue modeling is very stable and tractable but struggles to account for all of the conditions affecting a given airport (weather, layout, traffic patterns, and human factors). Microsimulations are highly accurate but are very cumbersome to set up and computationally demanding, making their use impractical for fast-time evaluations.

More recently, data-driven approaches have been adopted to overcome these challenges [8–11]. Notably, traffic prediction approaches using Machine Learning (ML) have proven their versatility and effectiveness predicting traffic dynamics in the short term, but ML models typically perform poorly for extended prediction horizons (e.g. longer than a week).

This work proposes the use of a hybrid framework to capture the complex, long-term traffic flow behavior of an airport's surface. We devise a method to simulate airport surface traffic flow using a combination of numerical discrete time models and encoding of historical state information. Our approach addresses some shortcomings faced by other simplified airport performance models and model-free approaches by encoding airport states with an Artificial

*Graduate Student, University of Cincinnati and NASA OSTEM intern, AIAA Student Member
†Senior Software Engineer, Metis Technology Solutions Inc., AIAA Member
‡Aviation Systems Division, NASA Ames Research Center
§Aviation Systems Division, NASA Ames Research Center, AIAA Associate Fellow
¶Autonomy & Naviation Technology Center, Airforce Institude of Technology, AIAA Associate Fellow

Neural Network (ANN). We validate the accuracy and stability of the proposed Neural-Queue framework by comparing simulated traffic and congestion metrics produced in an extended simulation with real (historical) traffic data.

A. Related Works

1. Airport Surface Queueing Model

Queueing Theory (QT) is a branch of operations research that provides tools to model the behavior of waiting lines or queues. Airports experiencing congested surface traffic behave very similarly to fundamental queuing problems (i.e., customers waiting in a queue for service), and aircraft waiting for takeoff has even become a common method of framing simple QT examples [12]. As a result, the movement and operation of aircraft on airport taxiways can be formulated quite easily and effectively using QT methods, and such methods have been applied to airport operations for decades. In particular, Badrinath et al. [13] have produced a queueing formulation that provides good performance predicting traffic flow on the airport surface. The authors' fundamental, single-server Poisson queue system is derived approximately using a fluid-flow model. Instead of the discrete, probabilistic model most commonly used in QT work, Badrinath et al employed a fluid-flow approximation described by Tipper et. al [14]. This approximation derives the aggregate flow in and out of the queue using an Ordinary Differential Equation. To find the performance at any point, a utilization factor is approximated and depends on the length of the queue. The utilization factor becomes a coefficient to modulate the true flow out of the queue system in a tractable manner, and combination with the flow into the queue produces a complete and self-regulating dynamic system which is capable of closely matching performance of a simulated benchmark solution, using some performance constants obtained empirically. Badrinath et al. chose Charlotte-Douglas International Airport (CLT) for study, and queue length forecasting was obtained by integrating the approximated performance of several queues which are chosen to represent areas of congestion on the airport surface. Queues were connected as a network, with inter-queue traffic modeled as a connected graph to match observed trends. Fitting service rates and utilization functions to the historical data, simulation of the flow with scheduled arrival and departure rates showed a good resemblance to the original behavior with regards to predicted queue length, when trained and analyzed over 5 months of summer operations (May-July 2015, May-June 2016, containing approx. 55,000 aircraft operations). Taxi times are also approximated using the forecasted queue behavior, and these also match with taxi-time data observed in the real airport system.

2. Machine Learning Applications in Queueing Systems

Machine learning, and in particular Deep Learning, have become very prominent in intelligent system design and control due to their remarkable versatility. Notably, the capabilities of neural networks with regard to dynamics systems is a burgeoning area of research. Physics-Informed Machine Learning (PIML) is one such example where a dynamic system is integrated with learning models [15] to introduce more capability where conventional approaches are impractical. The physics informed approach additionally solves the stability and compounding error present in ML dynamics simulation alone [16]. Combining Machine Learning (ML) with QT has been explored as a highly effective method of creating data-driven and adaptable queueing models [17, 18], where traditional models are limited with respect to state-dependent or time-dependent behavior, which is usually very difficult or sometimes completely intractable to formal QT analysis methods.

3. Contribution

The approach by Badrinath et al. is shown to effectively predict the flow of traffic and forecast surface congestion on the airport taxiway, but the service rate predictions are limited to single linear regression determined empirically using operational data. Traffic conditions on the airport surface are affected by a slew of factors, including time of day, weather conditions, runway configuration, landing rates, and the length of other queues, etc. which may not be appropriately captured via linear regression. Additionally, the approach outlined in [13] used 5 months of summer operational data for training and evaluation. In our neural-queuing approach, we extend the queue network model outlined in Badrinath et al. by informing service rates using an Artificial Neural Network (ANN), which will dramatically increase the capability of the queue network model by encoding nonlinear relationships between service rates with queues, weather, and other factors. Furthermore, we train and evaluate our model using a dataset containing all 2022 operations at CLT airport. We then evaluate our model by performing a 365-day uninterrupted simulation of traffic conditions using the neural queuing model to verify that our approach is stable and accurate even for very long time horizons.

II. Problem Formulation

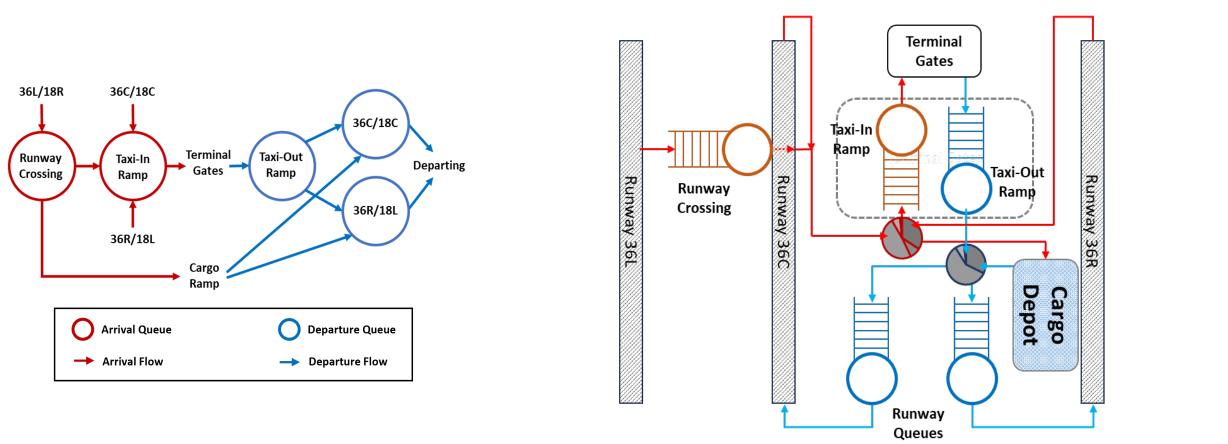
Beginning with the most formalized structure of QT, consider a Poisson process modeling the arrival of customers into a system (such as a runway, terminal ramp or taxiway crossing), which serves one customer at a time, on a first-come-first-served (FCFS) basis. Let λ denote the arrival rate of customers entering the system. Customers are being served and leaving the queue at an average rate of μ (service rate), which is also a Poisson process. λ and μ both vary with time, but we will assume for now that the system is quasi-stationary. The equation for a FCFS queue with a single server and unlimited capacity can be described as:

$$\frac{dx}{dt} = \lambda(t) - \mu(t), \ t \geq 0 \tag{1}$$

Where x is the state of the queue, containing the number of customers in the queue, $x \in \{0, 1, 2, 3, \ldots\}$. In our numerical model, we approximate the dynamical system as that of a discrete-time system formulated as a time-stepper Neural ODE [16], where the next state $\hat{x}_{k+1}, k \in \{0, 1, 2, \ldots\}$ is a function of the current state $\hat{x}_k$ and the derivatives of the system:

$$\hat{x}_{k+1} = \hat{x}_k + \hat{\lambda}^{ext}(t_k) - \mathcal{N}(t_k, \hat{x}_k) + \sum(\rho\hat{\mu}_{k-1}) \tag{2}$$

Where $\hat{x}$ is a queue vector consisting of m queues, $\hat{\lambda}^{ext}(t)$ is a time-varying arrival vector of length m obtained from flight schedules, $\hat{\mu}_{k-1}$ is the service rate vector of the previous timestep, ρ is a transition matrix (Square matrix with rows summing to 1) determined empirically, and $\mathcal{N}$ is an ANN mapping of the current state to the estimated service rate vector $\mathcal{N} : \hat{x}_k \rightarrow \hat{\mu}_k$. The transition matrix ρ is included to represent the flow received from another queue, which represents internal flow. Adapting the queue network system to surface traffic flow at CLT, a system is proposed consisting of $m = 5$ queues, including runways*36C/18C, 36R/18L, the 36C/18C runway crossing from 36L/18R, and the ramp from the terminal, which is split into departure and arrival ramps (See Figure 1b). We can assume knowledge of a pre-existing flight schedule, and therefore arrival rates into the system are assumed to be known. This leaves the unknown service rate parameter $\mu(t)$ for each queue. Queue-specific service rate for the airport system is highly dependent on aircraft state information such as queue lengths, weather, runway configuration, and time as well as human factors. Due to the highly uncertain and variable nature of predicting service rates, we use an ANN for regression of the queue network service rates ($\mathcal{N}$ in Equation 2)using the airport state information.



(a) Neural-Queue, Simplified Network Topology

(b) Map style layout with flow ratio intersections shown (North flow configuration, further explained in Figure 2)

Fig. 1 Neural-Queue network layout

*Airport runways are named based on their flow orientation, which are determined by their magnetic heading (in degrees) rounded to the nearest 10 degrees. A letter is also added to differentiate parallel runways (L for Left, C for Center, and R for Right, relative to runway orientation). Therefore, the center runway with a magnetic heading of 360 degrees is named 36C, or "Three-Six-C".

III. Methodology

A. Data

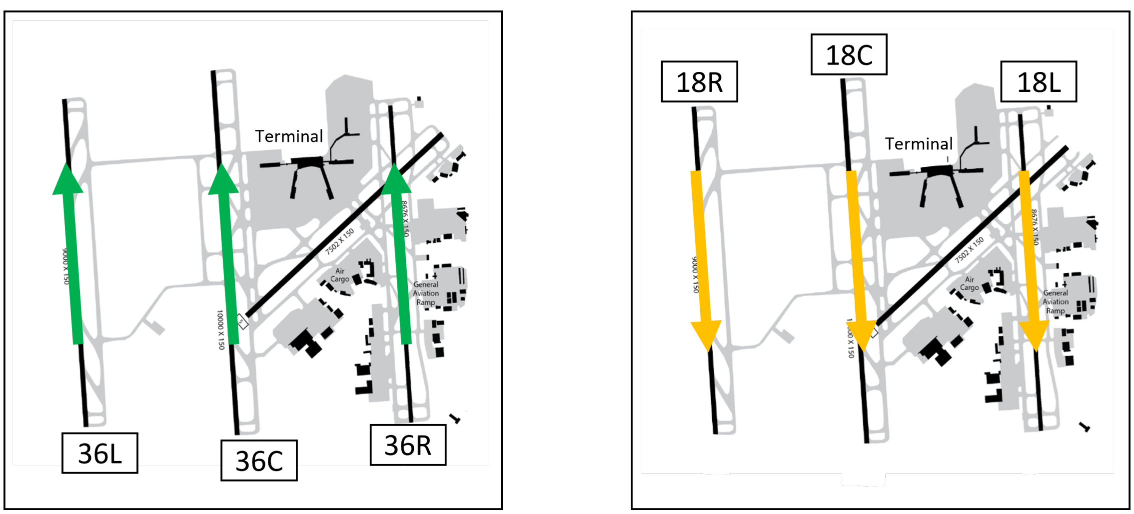
Historical airport traffic, weather and other relevant data is sourced from FAA Aviation System Performance Metrics (ASPM) [†] as well as from NASA Sherlock Data Warehouse [‡].

1. IFF Track Data

Airport Surface Detection Equipment, Model X (ASDE-X) is employed by airports and control centers to be able to track the location and status of flights occurring in the vicinity. Aircraft tracks include a long history of the ASDE-X samples and show a temporal trajectory traveled for each aircraft. NASA's Sherlock Data Warehouse stores the Identification- Friend or Foe (IFF) track data history for flights across the US received by local Air Traffic Control (ATC) centers. The IFF data used in this work includes aircraft tracks near CLT airport. CLT is a good benchmark airport for this studyl, being ranked as the 9th highest-traffic (US) domestic airport for passenger flights [19], while also maintaining relatively simple runway configuration protocols (shown in Fig. 2). Aircraft tracks begin once pushback occurs (for departures) or as aircraft move into the local airspace of the relevant airport (for arrivals). The tracks contain data sampled at 1hz such as latitude, longitude, speed, heading, current epoch, etc. Header entries in the track data contain flight mission data such as operation type (departure or arrival), beacon codes, destination etc.

B. Pre-processing

IFF track data obtained from NASA Sherlock is ill-suited to directly inform a queueing model, so initial processing must be performed to convert this information into a trainable set of data. First, the IFF files for each day of operations are queried for all departure and arrival flights during the day. Flight track information (coordinate position, speed, heading, etc.) is stored as time series per flight. Once collection of flight data is successful, a stepper model temporally advances through the available history duration to extract important airport statistics history. Each step, the active flights for that time are collected out of the data history and processed by local queueing zones to determine ownership, waiting status, and taxi time.

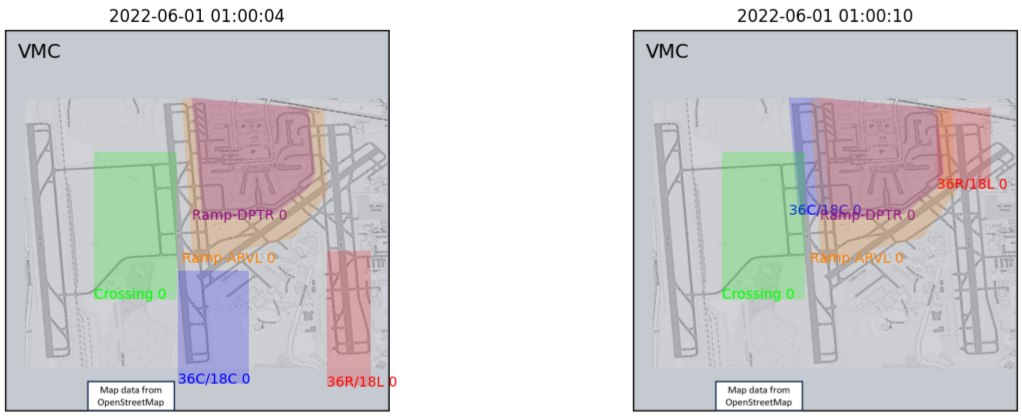


(a) CLT North Flow Configuration **(b) CLT South Flow Configuration**

Fig. 2 Common configurations for CLT airport with runways. Map image obtained from [20]

[†] https://aspm.faa.gov/

[‡] https://sherlock.opendata.arc.nasa.gov/sherlock_open/



(a) North Flow Bounding Boxes **(b) South Flow Bounding Boxes**

Fig. 3 CLT airport queue zone bounding boxes. Map background image sourced from [21]

Queue bounding boxes have been arranged to count the number of aircraft "waiting" at areas where congestion typically occurs (3). These areas include the runways 36C/18C(blue) and 36R/18L(red), The crossing(green) of runway 36C/18C from 36L/18R, the taxi-in ramp(orange) areas and the taxi-out ramp(purple) areas. In these zones, the arrival rate (# of aircraft entering the queue), service rate (# of aircraft leaving the queue) and queue length at every timestep is measured. Queues are modified based on the current runway configuration of the airport to measure aircraft at the expected queuing location. North flow and south flow are considered exclusively for this work, and variations of either configuration were observed to be used during the majority of total airport operation in 2015 [20]. See Figure 2 for an example of the airport configurations as well as the queue arrangement. ASPM data is queried for weather conditions: 1) Instrument Meteorological Conditions (IMC) or 2) Visual Meteorological Conditions (VMC), wind components, and temperature. Each flight's taxi time is measured between initial track/landing and departure/final track.

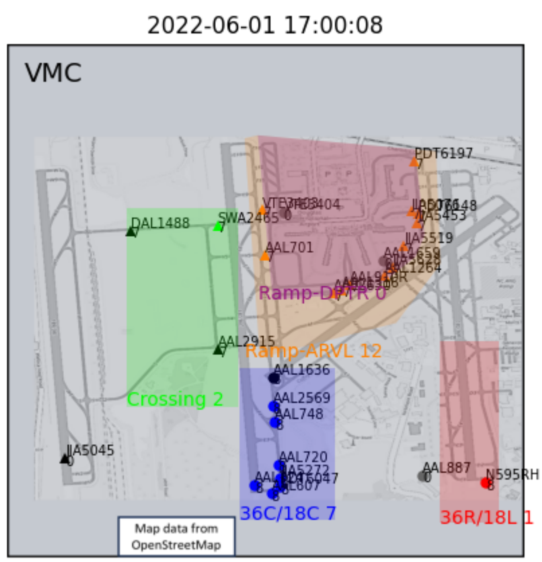


Fig. 4 CLT airport processing, Mid-day. Map background image sourced from [21]

At the end of the simulation processing, the 1 Hz samples are reduced to 5 minute chunks representing the aggregated queue dynamics extracted from the track data from 1/1/2022 to 12/31/2022.

C. Training

Following the pre-processing phase, a dataset is constructed containing samples of the following macroscopic queue-network parameters:

Feature	Data Type
Time-of-Day	Discrete [0, 1, 2, ... 23]
Day-of-Week	Discrete [0, 1, 2, ... 6]
Month-of-Year	Discrete [0, 1, 2, ... 11]
Flow	Discrete [0, 1]
IMC	Discrete [0, 1]
Temperature	Continuous
Wind (East-West)	Continuous
Wind (North-South)	Continuous
36C/38C Queue Length	Continuous
36R/18L Queue Length	Continuous
Crossing Queue Length	Continuous
Taxi-In Queue Length	Continuous
Taxi-Out Queue Length	Continuous
36C/38C Landing Rate	Continuous
36R/38L Landing Rate	Continuous

Table 1 Features used for training ANN model, with data types.

Feature	Data Type
36C/38C Service Rate	Continuous
36R/38L Service Rate	Continuous
Crossing Service Rate	Continuous
Taxi-In Service Rate	Continuous
Taxi-Out Service Rate	Continuous

Table 2 Target variables used as label data for training, with data types.

The discrete features are one-hot encoded, and the service rates are separated for use as target variables. The dataset used to train the model encompasses the 45 features (after one-hot encoding) listed in table 1. Following training, the fully-connected ANN is capable of predicting the service rates for all queues given the airport state.

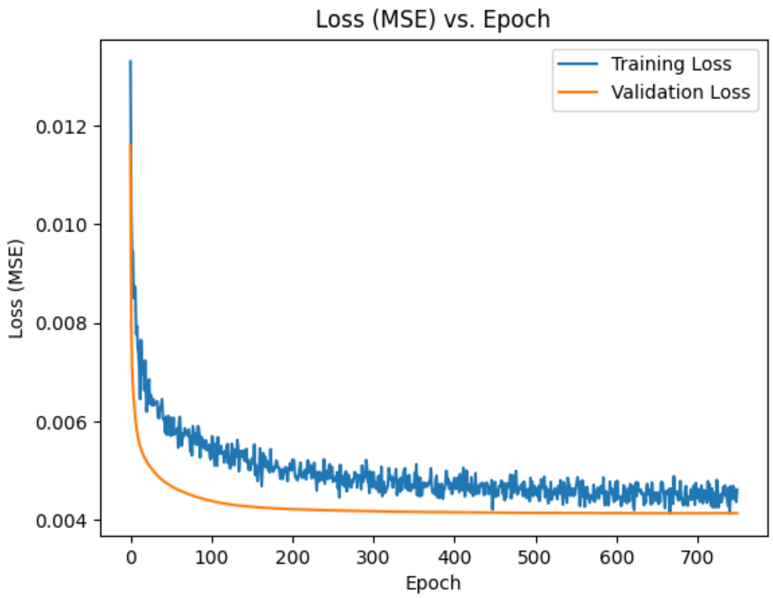


Fig. 5 Training Loss per Epoch

The ANN is constructed using 5 layers containing 45, 135, 135, 45, and 5 neurons, respectively. Each layer has a Leaky ReLU activation and 10% dropout is applied. The size of mini-batch is fixed to 128 and the model is trained for 750 epochs. We split the data into 70% for training and 30% for testing. Adam optimizer is used for training with learning rate 2.5×10^{-5} and Mean Squared Error (MSE), defined below, is used as the loss function:

$$MSE = \frac{1}{n}\sum_{i=0}^{n}(y_i - \hat{y}_i)^2 \tag{3}$$

D. Neural-Queue Model

With a trained model and arrival data processed from the IFF File, queuing dynamics can now be simulated with weather and arrival states. Arrival and weather states are currently being taken from the IFF processed data for testing. With the IFF history data, the trained model will parameterize an ODE solver to integrate a queue state according to the required states. Pseudocode of the neural-queue solver step (Equation 2) is shown here, for a single queue:

Algorithm 1 Neural-Queue Solver

```
while t < t_end do
    s ← data[t, :]              ▷ Query history data for airport state
    μ = ANNPredict(s)           ▷ See Table 1
    λ ← s[λ]                    ▷ Get Arrival Rate from historical data
    x = x + (λ − μ)             ▷ Update queue length
    t = t + Δt
end while
```

The queue model is used to replicate each of the 5 queue objects at a timestep of $\Delta t = 30$ seconds, with their “output” (served aircraft) feeding into the next network queue. Aircraft customer flow utilized by more than 1 queue is split using a transition matrix when necessary (split ratios determined empirically), such that only the arrival rates from outside of the system (landing rates, gate pushback rates) are needed to inform traffic loading during the simulation. Running the simulation with the arrival / pushback rates and weather states of the IFF history data can then be compared to the actual queue lengths measured during pre-processing.

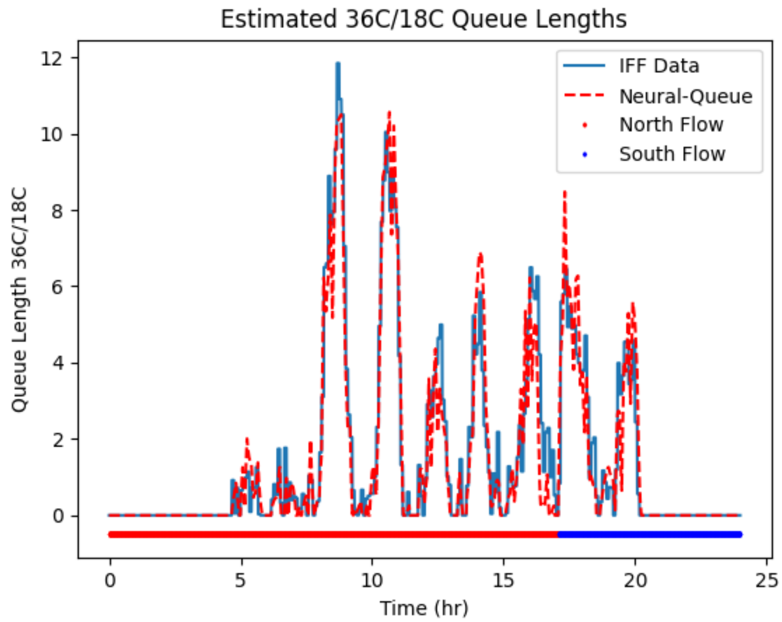


Fig. 6 Neural Queue estimation of traffic flow and queue lengths for 36C

E. Evaluation of Taxi-Time

Taxi-times of aircraft departing or arriving from the airport terminal can help assess the ability of the Neural-Queue model to describe traffic and congestion effects on an aircraft's trip in and out of the airport. Taxi-times at any time can be determined using the simulated queue lengths discussed, by querying wait times of queues along an aircraft's planned path. Calculation of the expected taxi time is performed with the following process:

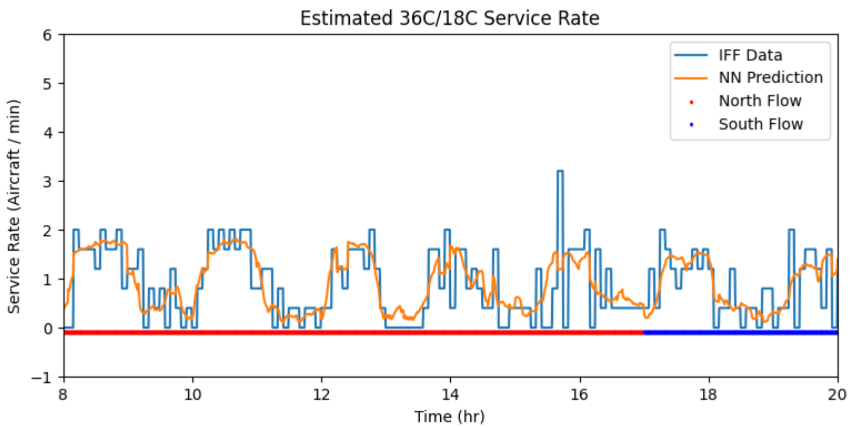
Algorithm 2 Wait-Time Estimation

```
t = t_0                        ▷ Start time
W = 0                          ▷ Wait time running total
q ← q[t_0]                     ▷ Query queue length at start
while q > 0 do
    t = t + Δt
    μ = μ_hist[t]              ▷ Query simulation for service rate at time t
    q = q − μ                  ▷ Update queue position based on service rate
    W = W + Δt
end while
W = W − q/μ[t]                 ▷ Total weight time in queue, correction for overshooting
```

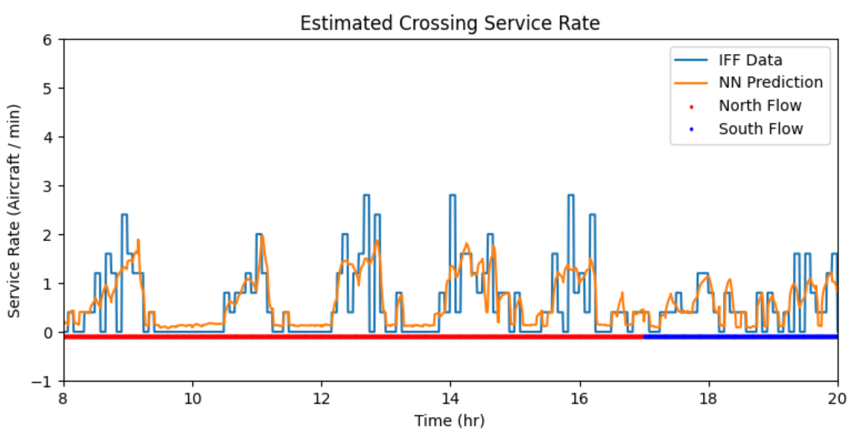
In this way, the average aggregate taxi times for departure flights can be deduced by similar combination of 36C/18C and 36R/18L routes. Arrival taxi time prediction is calculated in the same way using travel through the crossing. After wait time is calculated, additional time is added to each estimate to account for the aggregate time spent between queues on the airport taxiways. To compare with the true performance, true taxi time averages are measured during pre-processing and calculated per 15-minute chunk.

IV. Results

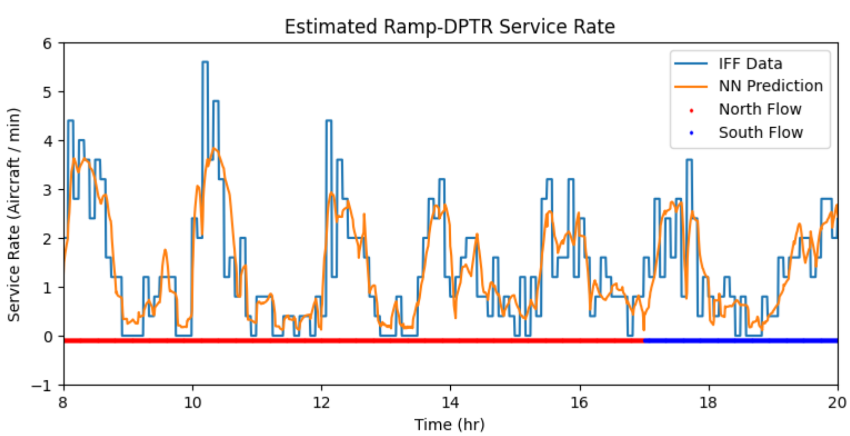
A dataset containing all ASDE-X activity and weather data for all 365 days of the calendar year 2022 was pre-processed into 5-minute chunks for training the ANN. After training, the same dataset is queried for scheduled terminal departure rates and landing rates per runway. The simulation is run over the entire year using the Neural-Queueing model, with the scheduled rates from the historical data. Weather and runway configuration are also used from the historical data. The following plots are used as example performance from a standard day (July 14, 2022) using the predicted behavior of the queue network, and following with taxi time prediction for the same day with the derived queue behavior.



(a) 36C/18C runway queue service rate

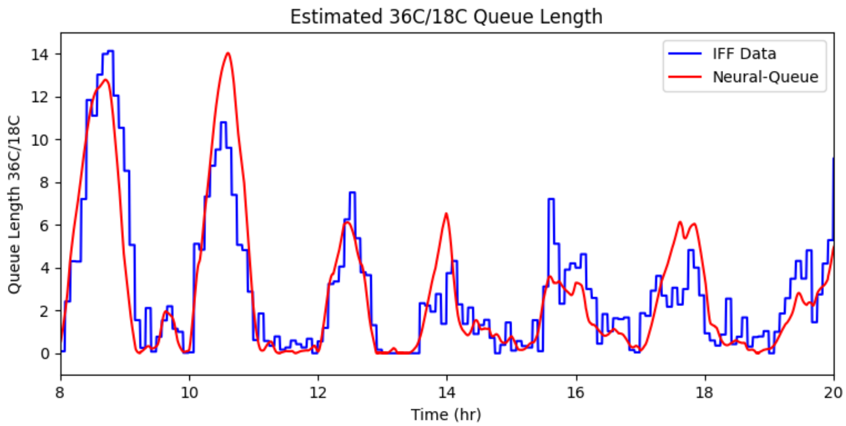


(b) Crossing queue service rate

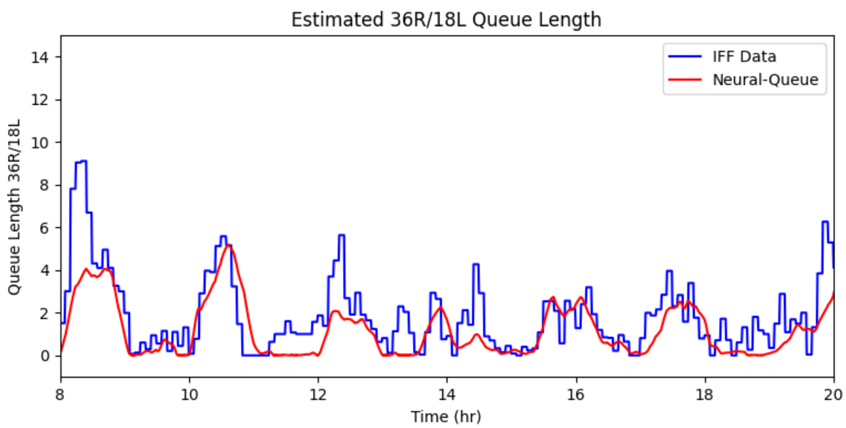


(c) Departure ramp queue service rate

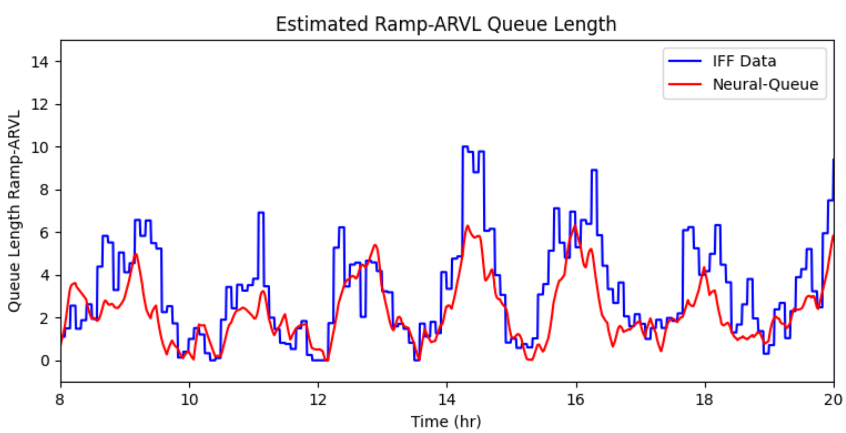
Fig. 7 Prediction of queue service rate μ



(a) 36C/18C runway queue length



(b) 36R/18L runway queue length



(c) Arrival ramp queue length

Fig. 8 Simulated queue length

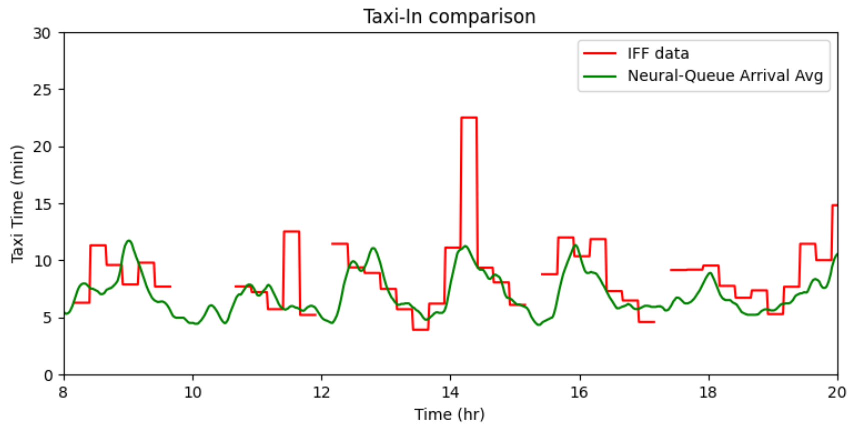


Fig. 9 Prediction of average taxi times (Arrivals)

The error metrics for the 365 days simulation is calculated using eqs. 3-6:

$$MAE = \frac{1}{n}\sum_{i=0}^{n}|y_i - \hat{y}_i| \tag{4}$$

$$STD(MSE) = \sqrt{\frac{1}{n}\sum_{i=0}^{n}((y_i - x_i)^2 - MSE)^2} \tag{5}$$

$$STD(MAE) = \sqrt{\frac{1}{n}\sum_{i=0}^{n}(|y_i - x_i| - MAE)^2} \tag{6}$$

Service Rate				
Queue Location	MSE	MSE-STD	MAE	MAE-STD
36C/18C	0.036	0.081	0.138	0.129
36R/18L	0.032	0.076	0.132	0.120
Crossing	0.031	0.094	0.115	0.135
Ramp-ARVL	0.087	0.200	0.207	0.209
Ramp-DPTR	0.066	0.160	0.176	0.189

Table 3 Mean squared error (MSE) and mean absolute error (MAE) of predicted service rate (in aircraft/min) for 365-days simulation, with standard deviation (STD)

Queue Length				
Queue Location	MSE	MSE-STD	MAE	MAE-STD
36C/18C	3.087	10.497	0.940	1.484
36R/18L	2.015	6.711	0.793	1.177
Crossing	0.410	3.897	0.229	0.598
Ramp-ARVL	3.864	14.816	1.199	1.557
Ramp-DPTR	1.989	12.914	0.679	1.236

Table 4 Mean squared error and mean absolute error of predicted queue length for 365-days simulation, with standard deviation of error

Taxi Time Prediction		
Operation	MSE	MAE
Taxi-in	23.534	2.711
Taxi-out	79.165	4.586

Table 5 Mean squared error and mean absolute error of predicted taxi times (in minutes) for 365-days simulation

V. Discussion

Reviewing performance from the model over the year-long dataset, it is clear that the Neural-Queue model is capable of matching the aggregate airport traffic behavior and remains stable for long time horizons (error statistics for 365 days is shown). The predictions for service rates maintain a close relationship (Figure 7) with the trends, on average, with the ground-truth data obtained during pre-processing. The same can be said for the queue lengths, which normally fall within <1.5 waiting aircraft of queue length error. It is notable that there are queue length spikes that occur in the real system that are missing in the simulated performance. Such disturbances can be quite problematic as there are a wide variety of transient traffic effects and stoppages such as aircraft maintenance issues and runway/ taxiway closures which can locally slow or halt the flow of traffic. These instances are very difficult to predict and the current model is not able to model such sparse occurrences. Additionally, service rates of the runways may vary due to differing aircraft weight classes or departure fixes which can affect the maximum rate of departures a runway can support. This behavior is not captured in this work.

Considering taxi-time predictions, the model shows a generally close match with the ground truth, but there are clear errors visible in the Figure 9 shown. Taxi time collection from the historical data proves to be quite difficult, as there can be a large variety of potential paths through the airport and can be changed significantly by the runway and gate used, and whether the aircraft is an air cargo / general aviation flight. In this way, taxi time averages are subject to considerable variation especially considering there may be less than 5 aircraft taxiing for a period of 15 minutes. These factors make the ground truth data quite difficult to use as the aggregated taxi times may differ when particularly long taxi routes are taken, or if there is an issue causing flights to taxi longer than normal such as a section of the taxiway closed for maintenance. These incidents can cause the average time predictions to become skewed which leads to large errors. In this work, we have chosen to only compare the prediction for chunks where the number of taxi times considered is greater than 5, but this can still be subject to variability. Overall, the model shows great promise when predicting general queue lengths and taxi times.

VI. Conclusions and Future Work

The Neural-Queue model presented in this work shows promising results for forecasting airport traffic behavior and help improve efficiency of ground operations. We have been able to capture the aggregate service rates of congestion points at CLT airport using a trained ANN, and have succeeded in using the trained policy to simulate the queue network formulated on principles of QT. By simulating the network using historical or forecast data, it is then possible to derive valuable metrics of airport performance such as taxi times.

To comprehensively validate our Neural-Queuing model, we aim to expand the structure of the queuing model to effectively capture the transient flow of configuration changes, as well as to more accurately capture the time-varying usage of runways. In our improved model, it is desirable to use different flow paths to describe a more accurate aggregate of taxi-times. By re-evaluating our neural-network model approach, we can more effectively capture the relationship between airport state and queue services rates with a specific emphasis on transient, short-term effects. In the future, we also aim to apply this neural-queuing model to create an online simulation of an airport. For validation of our approach, expansion to other airports is also planned, as well as utilizing airport throughput as an additional metric to compare with the real system.

Acknowledgements

The authors would like to thank retired Air Traffic controller and subject matter expert Kari Gonter for his valuable insight on Air Traffic Control practices.

References

[1] Kistler, M., and Gupta, G., "Relationship Between Airport Efficiency and Surface Traffic," *9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*, Aviation Technology, Integration, and Operations (ATIO) Conferences, American Institute of Aeronautics and Astronautics, 2009. https://doi.org/10.2514/6.2009-7078, URL https://arc.aiaa.org/doi/10.2514/6.2009-7078.

[2] Simaiakis, I., and Balakrishnan, H., "Impact of Congestion on Taxi Times, Fuel Burn, and Emissions at Major Airports," *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 2184, No. 1, 2010, pp. 22–30. https://doi.org/10.3141/2184-03, URL https://journals.sagepub.com/doi/10.3141/2184-03.

[3] Li, L., Clarke, J.-P., Chien, H.-H. C., and Melconian, T., "A probabilistic decision-making model for runway configuration planning under stochastic wind conditions," *2009 IEEE/AIAA 28th Digital Avionics Systems Conference*, 2009, pp. 3.A.5–1–3.A.5–10. https://doi.org/10.1109/DASC.2009.5347528, URL https://ieeexplore.ieee.org/document/5347528, iSSN: 2155-7209.

[4] Ng, K. K. H., Lee, C. K. M., and Chan, F. T. S., "A robust optimisation approach to the aircraft sequencing and scheduling problem with runway configuration planning," *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2017, pp. 40–44. https://doi.org/10.1109/IEEM.2017.8289847, URL https://ieeexplore.ieee.org/abstract/document/8289847, iSSN: 2157-362X.

[5] Memarzadeh, M., Puranik, T. G., Kalyanam, K. M., and Ryan, W., "Airport Runway Configuration Management with Offline Model-Free Reinforcement Learning," *AIAA SciTech*, National Harbor, MD, 2023. https://doi.org/10.2514/6.2023-0504, URL https://arc.aiaa.org/doi/abs/10.2514/6.2023-0504.

[6] Simaiakis, I., and Balakrishnan, H., "A Queuing Model of the Airport Departure Process," *Transportation Science*, Vol. 50, No. 1, 2016, pp. 94–109. https://doi.org/10.1287/trsc.2015.0603, URL https://pubsonline.informs.org/doi/10.1287/trsc.2015.0603, publisher: INFORMS.

[7] "Simmod PRO!" ATAC Corporation, 2024. URL https://www.atac.com/simmod-pro/, accessed: 2024-10-21.

[8] Lunacek, M., Williams, L., Severino, J., Ficenec, K., Ugirumurera, J., Eash, M., Ge, Y., and Phillips, C., "A data-driven operational model for traffic at the Dallas Fort Worth International Airport," *Journal of Air Transport Management*, Vol. 94, 2021, p. 102061. https://doi.org/https://doi.org/10.1016/j.jairtraman.2021.102061, URL https://www.sciencedirect.com/science/article/pii/S0969699721000442.

[9] Feng, H., and Zhang, Y., "Prediction of Hourly Airport Operational Throughput with a Multi-Branch Convolutional Neural Network," *Aerospace*, Vol. 11, No. 1, 2024, p. 78. https://doi.org/10.3390/aerospace11010078, URL https://www.mdpi.com/2226-4310/11/1/78, number: 1 Publisher: Multidisciplinary Digital Publishing Institute.

[10] Zhu, X., Lin, Y., He, Y., Tsui, K.-L., Chan, P. W., and Li, L., "Short-Term Nationwide Airport Throughput Prediction With Graph Attention Recurrent Neural Network," *Frontiers in Artificial Intelligence*, Vol. 5, 2022. https://doi.org/10.3389/frai.2022.884485, URL https://www.frontiersin.org/articles/10.3389/frai.2022.884485, publisher: Frontiers.

[11] Yan, Z., Yang, H., Li, F., and Lin, Y., "A Deep Learning Approach for Short-Term Airport Traffic Flow Prediction," *Aerospace*, Vol. 9, No. 1, 2022, p. 11. https://doi.org/10.3390/aerospace9010011, URL https://www.mdpi.com/2226-4310/9/1/11, number: 1 Publisher: Multidisciplinary Digital Publishing Institute.

[12] Gross, D., Shortle, J. F., Thompson, J. M., and Harris, C. M., *Fundamentals of Queueing Theory, 4th Edition*, John Wiley & Sons, Inc., Hoboken, New Jersey., 2008. https://doi.org/10.1002/9781118625651.

[13] Badrinath, S., Li, M. Z., and Balakrishnan, H., "Integrated Surface–Airspace Model of Airport Departures," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 5, 2019, pp. 1049–1063. https://doi.org/10.2514/1.G003964, URL https://arc.aiaa.org/doi/10.2514/1.G003964.

[14] Tipper, D., and Sundareshan, M., "Numerical Methods for Modeling Computer Networks Under Nonstationary Conditions," *Selected Areas in Communications, IEEE Journal on*, Vol. 8, 1991, pp. 1682 – 1695. https://doi.org/10.1109/49.62855.

[15] Nghiem, T. X., Drgoňa, J., Jones, C., Nagy, Z., Schwan, R., Dey, B., Chakrabarty, A., Di Cairano, S., Paulson, J. A., Carron, A., Zeilinger, M. N., Shaw Cortez, W., and Vrabie, D. L., "Physics-Informed Machine Learning for Modeling and Control of Dynamical Systems," *2023 American Control Conference (ACC)*, 2023, pp. 3735–3750. https://doi.org/10.23919/ACC55779.2023.10155901, URL https://ieeexplore.ieee.org/document/10155901, iSSN: 2378-5861.

[16] Legaard, C. M., Schranz, T., Schweiger, G., Drgoňa, J., Falay, B., Gomes, C., Iosifidis, A., Abkar, M., and Larsen, P. G., "Constructing Neural Network-Based Models for Simulating Dynamical Systems," , Jul. 2022. https://doi.org/10.48550/arXiv.2111.01495, URL http://arxiv.org/abs/2111.01495, arXiv:2111.01495 [cs].

[17] Garbi, G., Incerto, E., and Tribastone, M., "Learning Queuing Networks by Recurrent Neural Networks," *Proceedings of the ACM/SPEC International Conference on Performance Engineering*, ACM, Edmonton AB Canada, 2020, pp. 56–66. https://doi.org/10.1145/3358960.3379134, URL https://dl.acm.org/doi/10.1145/3358960.3379134.

[18] Sherzer, E., Baron, O., Krass, D., and Resheff, Y., "Approximating G(t)/GI/1 queues with deep learning," , 2024. URL https://arxiv.org/abs/2407.08765.

[19] "2023 Airport Traffic Report," Port Authority of New York and New Jersey, 2023. URL https://www.panynj.gov/content/dam/airports/statistics/statistics-general-info/annual-atr/ATR_2023.pdf, accessed: 2024-10-7.

[20] "CLT Airport Capacity Profile," Federal Aviation Administration, 2015. URL https://www.faa.gov/sites/faa.gov/files/airports/planning_capacity/profiles/CLT-Airport-Capacity-Profile-2015.pdf, accessed: 2024-10-8.

[21] OpenStreetMap, 2024. URL https://www.openstreetmap.org/copyright, accessed: 2024-10-7.