MULTI-SCALE AND COLLABORATIVE DISASTER EVACUATION PLANNING FRAMEWORK

FINAL REPORT

AUGUST 2021

Authors

Kai Zhang, Yongxin Liu, Yupeng Yang, Dahai Liu, Houbing Song

Embry-Riddle Aeronautical University

US DEPARTMENT OF TRANSPORTATION GRANT 69A3551747125

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation, University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Multi-scale and Collaborative Disaster Evacuation Planning Framework		5. Report Date August, 2021 6. Source Organization Code :ERAU Cost Center 61603
7. Author(s) Kai Zhang, Yongxin Liu, Yupeng Yang, Dahai Liu, Houbing Song		8. Source Organization Report No. CATM-2021-R2.ERAU
9. Performing Organization Name and Address Center for Advanced Transportation Mobility Transportation Institute 1601 E. Market Street Greensboro, NC 27411		10. Work Unit No. (TRAIS) 11. Contract or Grant No. USDOT: 69A3551747125
12. Sponsoring Agency Name and Address University Transportation Centers Program (RDT-30) Office of the Secretary of Transportation–Research U.S. Department of Transportation 1200 New Jersey Avenue, SE Washington, DC 20590-0001		13. Type of Report and Period Covered Final Report: 01/20-08/21 14. Sponsoring Agency Code USDOT/OST-R/CATM
15. Supplementary Notes:		
16. Abstract 17. When emergency occurs, no tools are available to assist the decision-making of airline planning and Coordination. In this project, we use big data and multi-agent modeling to integrate ADS-B data and Weather information, to optimize and visualize the airspace strategic planning during disaster, and Develop a forecast and recommendation system to aid the authorities and public for optimal airline Evacuation process, by using the deep reinforcement learning technique.		
18. Key Words Emergency; Evacuation; Simulation; Social force model	19. Distribution Statement Unrestricted; Document is available to the public through the National Technical Information Service; Springfield, VT.	

20. Security Classif. (of this report) Unclassified	21. Security Classif. (of this page) Unclassified	22. No. of Pages 100	23. Price …

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

CONTENTS

Executive Summary

Extreme weather conditions, such as floods, hurricanes, and wildfires, cause large-scale human population movements and evacuations in the world. Taking flights to evacuate the area before a disaster approaches is mostly preferred because it is much more efficient than driving or taking trains. However, making effective flight plans serving evacuation before these natural disasters occur is a significant challenge because it relies heavily on many dynamic factors including human resources, environments, and traffic loads.

In this study, we propose an adaptive flight dispatching framework that aims to coordinate available resources such as aircraft and airports to transport people from evacuation areas to safe places. We formulate the evacuation flight dispatch as a weighted graph matching problem where weights between edges (aircraft-airway) indicating the dynamics of the air route network are inferred by machine learning approaches. The problem is solved by a learning and planning manner: (a) in the offline learning phase, we first summarize demand (evacuation airway) and supply (aircraft) patterns into a spatiotemporal quantization, each of which refers to expected value of an aircraft/pilot being in a particular state. Furthermore, in order to tackle uncertainty in the complex air route network for real-time air traffic management, we apply elaborated machine learning models to estimate air mobility indicators accurately in a local view; (b) in the online planning phase, the dispatch is solved using the combinatorial optimization approach, where each aircraft-airway pair is valued considering both immediate rewards and future gains that are attained based on the trained predictive machine learning model and the learned unified evaluation metric on spatiotemporal states.

Our framework is evaluated using data collected during Hurricane Irma, and we also conducted extensive experiments in terms of limited resources, incident response, and uncertainty of external factors, which demonstrate the effectiveness and robustness of the proposed schema in the real-world application. It is believed that our model will be beneficial for real-time evacuation planning and decision-making for a wide range of situations that involve air traveling.

Background

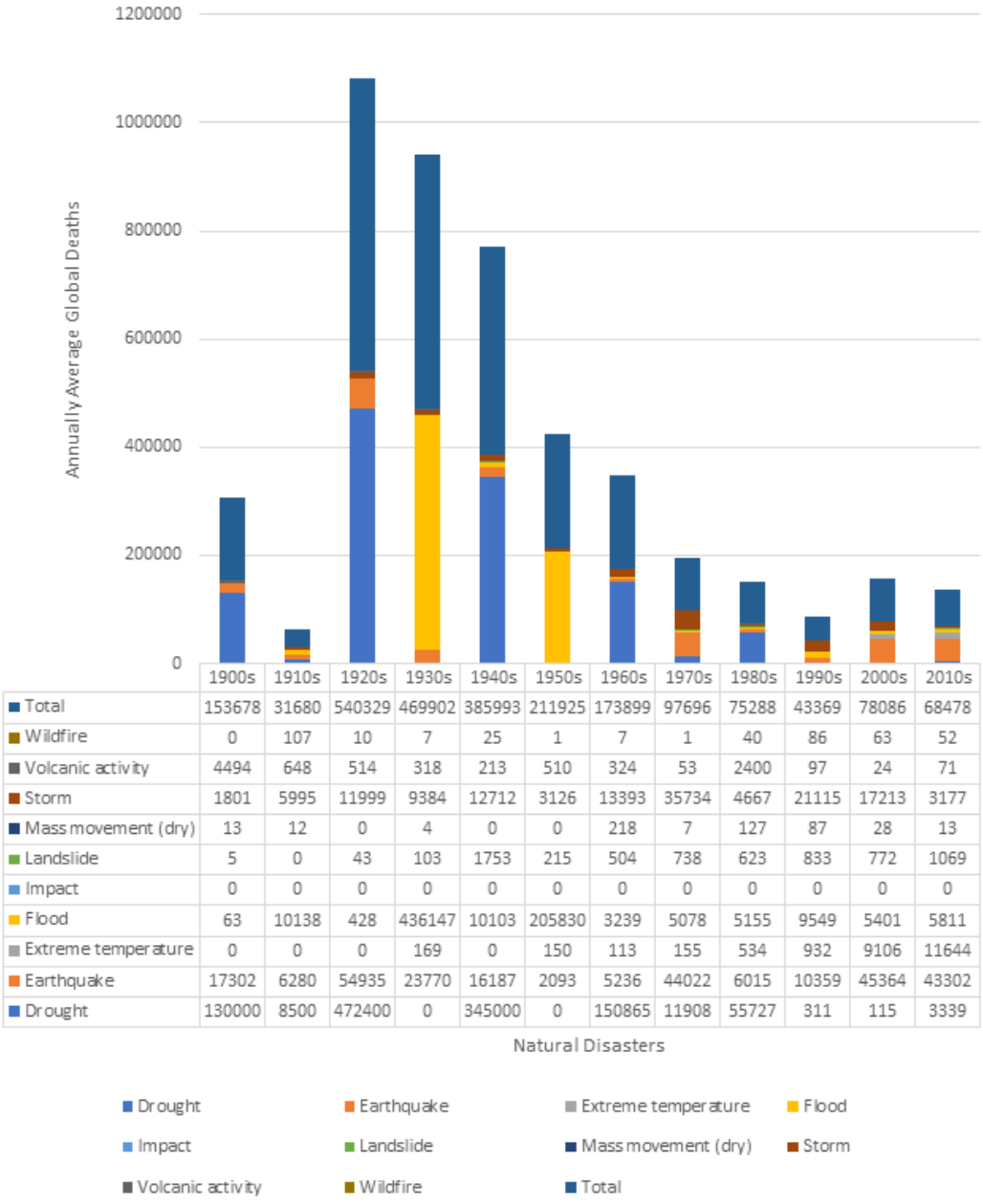
In this section, we first review the characteristics of disasters particularly natural catastrophe, then we introduce the fundamental techniques for weather forecasting and the demand modeling of evacuation transportation. The last and the most important part in the literature review is on the algorithms on traffic assignment since it is directly related to the core component (flight dispatching schema) in our work.

An Overview of Nature Disasters

Regina et al. (2009) established the hierarchy of disaster categories based on the existing disaster categories using several global data sets. They classified the generic disasters into natural and technological groups. Natural disasters are then classified into six categories (i.e., geophysical, meteorological, hydrological, climatological, biological, extra-terrestrial). Under each category, disasters are subclassified into self-explanatory main-types, sub-types and sub-sub types if available. For example, under the category geophysical, there are three main-types: earthquake, volcano and mass movement (dry). Under the main-type mass movement (dry), there are sub-types: rockfall, avalanche, landslide, and subsidence. Under the sub-type of avalanche, there are sub-sub types of snow avalanches and debris avalanches. Technological disasters (e.g., radiation leaks, building collapses) are unpredictable disasters that are related to human errors in technology (Lindsey, 2021). As the other group of generic disaster, technological disasters have an impact on individuals, families and communities involved in the disasters. They cause

economic losses and psychological diseases (e.g., post-traumatic stress disorder, stress, depression, anxiety) as well.

According to Ritchie and Roser (2019), over the past decade, natural disasters cause approximately 60,000 deaths annually all around the world, which is 0.1% of total deaths. Besides, disasters also cause severe global economic losses, which account for 0.12%-0.50% of GDP during 1990 to 2017. In the United States, direct economic loss attributed to disasters relative to GDP ranged from 0.10% to 0.32% during 2015 to 2019 (United Nations, 2021). For unpreventable disasters (e.g., earthquakes), they happen at a low frequency but have strong impacts on human society. However, people can suffer less with earlier prediction (Ritchie & Roser, 2019), which is supported by a significant decline in deaths over the past century (Figure 1). Figure 1 and Figure 2 show the severity and frequency of each type of natural catastrophe.



	1900s	1910s	1920s	1930s	1940s	1950s	1960s	1970s	1980s	1990s	2000s	2010s
Total	153678	31680	540329	469902	385993	211925	173899	97696	75288	43369	78086	68478
Wildfire	0	107	10	7	25	1	7	1	40	86	63	52
Volcanic activity	4494	648	514	318	213	510	324	53	2400	97	24	71
Storm	1801	5995	11999	9384	12712	3126	13393	35734	4667	21115	17213	3177
Mass movement (dry)	13	12	0	4	0	0	218	7	127	87	28	13
Landslide	5	0	43	103	1753	215	504	738	623	833	772	1069
Impact	0	0	0	0	0	0	0	0	0	0	0	0
Flood	63	10138	428	436147	10103	205830	3239	5078	5155	9549	5401	5811
Extreme temperature	0	0	0	169	0	150	113	155	534	932	9106	11644
Earthquake	17302	6280	54935	23770	16187	2093	5236	44022	6015	10359	45364	43302
Drought	130000	8500	472400	0	345000	0	150865	11908	55727	311	115	3339

Figure 1. Annually Average Global Deaths from Natural Disasters per Decade, 1900-2015.(Ritchie, n.d.)

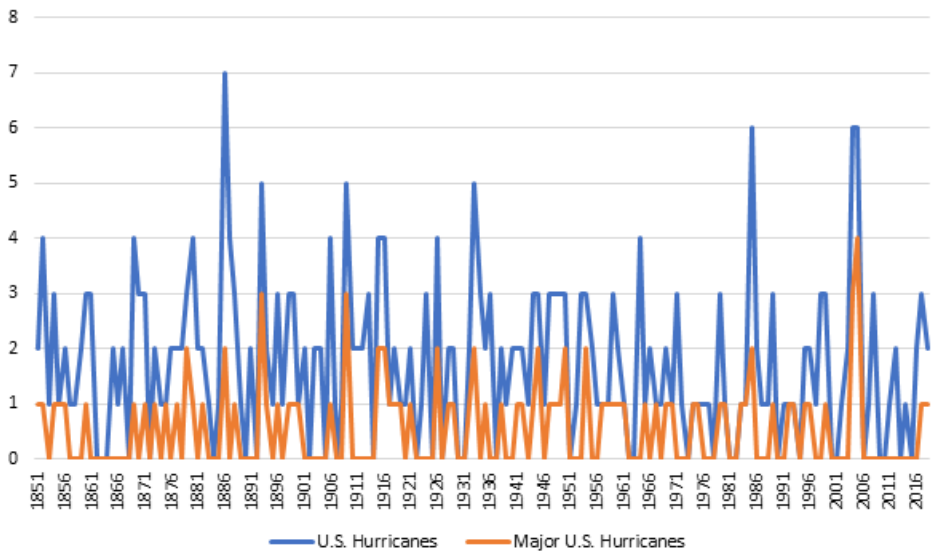


Figure 2. Frequency of U.S. Hurricanes and major U.S. Hurricanes, 1851-2018. (National Oceanic & Atmospheric Administration, 2019)

Forecasting Models for Disaster Management

According to the definition by International Federation of Red Cross and Red Crescent Societies (IFRC) (n.d.), disaster management is mainly the responsibility of relevant organizations and involves the mobilization of resources from all sectors of society. The essence of disaster management is to protect human welfare and reduce the impact of disasters on human society. Participants involved in disaster management may include 1) governmental disaster management agencies, 2) non-governmental organizations (NGOs), including the private sector and academia and 3) multilateral organizations and international financial institutions (Coppola, 2015).

The stages of disaster management include pre-disaster preparation, response to disasters, and post-disaster restoration. Residents near the disaster site are often the first to respond to the disaster (IFRC, n.d.). Therefore, in the pre-disaster preparation stage, residents of the same community should be mobilized to teach them how to deal with disasters together, thereby strengthening the ability of the community to resist disasters. The main role of the response stage is to minimize the number of injured, loss of life, economic loss, and environmental damage. The response stage is characterized by high pressure, limited time, and limited information (Coppola, 2015). As soon as people realize that a disaster has occurred, they will immediately begin to respond. The tasks include saving lives, search and rescue, evacuation, disaster assessment, life support, and donation management. In the post-disaster restoration phase, the country, communities, families and individuals repair or rebuild the living facilities damaged by the disaster and strengthen the buildings' ability to withstand disasters. The objective of this stage is to restore social order and stabilize the lives of the victims. Restoration is the most diverse and expensive stage in disaster management.

Considering the predictability of hurricanes, effective and efficient evacuation planning (in the preparedness phase) before hurricane approach plays a significant role in reducing causalities and losses. In June 2018, Centers for Disease Control and Prevention (CDC) published a reference document entitled "Preparedness and Safety Messaging for Hurricanes, Flooding, and Similar Disasters" that contained key messages on hurricanes and flood related health threats. It instructed people how to evacuate safely after hearing an evacuation order in severe hurricane weather.

(1) Weather Modeling

National Centers for Environmental Information (n.d.) introduced the global numerical weather prediction system, mathematical modeling, to predict weather: The Global Forecast System (GFS) is a National Centers for Environmental Prediction (NCEP) weather forecast model that generates data for dozens of atmospheric and land-soil variables, including temperatures, winds, precipitation, soil moisture, and atmospheric ozone concentration. The system couples four separate models (atmosphere, ocean model, land/soil model, and sea ice) that work together to accurately depict weather conditions. The model is constantly evolving, and regularly adjusted to improve performance and forecast accuracy. GFS is a global model with a base horizontal resolution of 18 miles (28 kilometers) between grid points. Temporal resolution covers analysis and forecasts out to 16 days. Horizontal resolution drops to 44 miles (70 kilometers) between grid points for forecasts between one week and two weeks.

Another existing method is the convolutional long short-term memory (ConvLSTM), a machine learning approach. ConvLSTM, a type of recurrent neural network, is used for spatio-temporal prediction. ConvLSTM has convolutional structures in internal matrix multiplication, both the input-to-state and state-to-state transitions. Shi et al. (2015) used ConvLSTM to build an end-to-end trainable model for the crucial and challenging precipitation nowcasting problem. They successfully predicted the future rainfall intensity in a local region over a relatively short period of time. Besides, they found out that ConvLSTM consistently outperformed fully connected long short-term memory and the state-of-the-art operational ROVER algorithm for precipitation nowcasting.

(2) Factors Affecting Demand Modeling

Evacuation research involving human factors and psychology is on the rise. The basic purpose of this type of modeling is to predict how people behave in danger (Schatz et al., 2014). Improved computer performance helps researchers study more complex behavior patterns (Hofinger et al., 2014). Compared with field experiments, the analysis of real events may enable researchers to learn more about human behavior, because participants usually know that there is no real danger in field experiments (Zinke et al., 2014). The three aspects that are usually considered when modeling are the type of building or infrastructure and its environment, the incident with the rapidness of its impact, and the human physical body (Kirchberger, 2006). Tubbs and Meacham (2007) summarized three typical assumptions about human behavior used to model behavior.

In recent evacuation research, active topics include physical characteristics, information processing (i.e., perception, recognition, appraisal of evacuation alarms), cognitive needs (i.e., certainty, feeling of control, curiosity), and group phenomena (i.e., affiliation and leadership in small groups in crowds). A human factors approach considers psychological variables and the interrelations between human factors at different levels (i.e., the individual level, group level, organizational level, technological level, the system's environment) (Hofinger et al., 2014).

The influence of social networks is also one of the factors in deciding whether to evacuate. The first evacuation advice people receive may come from friends and relatives (Lindell & Perry, 1987). People tend to trust the information of those around them. If friends and relatives say that the situation is dangerous, people will choose to believe it. Maybe at first people decided to stay there, but the evacuation appeals of people around them would encourage

people to change their minds and evacuate together (Baker, 1991). Mining useful information from the four dimensions of social media data (i.e., time, space, network, and content) can help people increase situational awareness and improve disaster response (Wang & Ye, 2018).

The study found that when the authorities recommended to evacuate, companies closed and evacuated. This correlation makes it difficult for researchers to determine whether official warnings or social cues are the cause of the evacuation (Baker, 1991). Hasan and Ukkusuri (2011) hypothesized that social cues are the factors leading to evacuation. They created a social contagion model to study the conditions that determine the cascading reaction of the network through evacuation. They also considered the impact of the mixed model of social communities, the first person to respond to the evacuation, and the decisions made by neighbors. They applied a mathematical and simulation model to investigate these factors in order to achieve the expected behavior of regional evacuation based on social connections. Rogers and Sorensen (1991) developed an early combined model of diffusion and contagion for the spread of emergency alerts. Their model used logistic functions to represent the spread of emergency warnings. The parameters on the model can be adjusted according to the maximum penetration rate of each system to represent different early warning systems, which are related to the characteristics of the system and the possibility of the public being in different positions at a specific time. The contagion aspect of the model represented the first-time people hear the event and spread the information to others (Rogers & Sorensen, 1991).

Traffic Assignment and Route Selection

After obtaining predicted factors such as weather conditions and evacuee demands, the traffic assignment sought to link land use and behavior to inform transportation planning. In this part, we review how to associate "driver" with routes once transit services are anticipated to follow pre-defined routes. Broadly, such a problem can be solved by three major methodologies, namely *Mathematical Optimization, Simulation-based Modeling, and Reinforcement Learning.* Note that most literature focuses on road traffic research.

(1) Mathematical Optimization

The large-scale emergency evacuation planning and management is typically addressed using network-based mathematical programming, in which User Equilibrium (UE), the System Optimal (SO) approaches and Nearest Allocation (NA) are the most popular and common models to date for traffic assignment, while the Constrained System Optimal (CSO) model provides a compromise between SO and UE/NA approaches (Bayran, 2016).

(a) User Equilibrium.

The first principal of UE is defined as "the journey times in all routes actually used are equal and less than those which would be experienced by a single vehicle on any unused route." (Wardrop, 1952). As an extensive version of UE, the Stochastic User Equilibrium (SUE) supposes that the evacuees select shortest travel time path based on their estimation of travel time. UE is attained once the variance of travel time estimated by the evacuees is zero. In such a case, travelers do not consider the whole evacuation system but just act selfishly for minimizing

their own travel times. Therefore, once no one can improve their travel time by unilaterally changing routes, the system converges to a stable condition (Daskin, 1985).

The UE modeling is built based on a directed network $G = (N, A)$, where N is the set of nodes and A is the set of links in the network. Besides, each arc $a \in A$ represents a convex travel time function. Now we let O denote the set of origins while D is the set of destinations. The following constraints should be satisfied in a UE,

$$f_k^{rs}(C_k^{rs} - C^{rs*}) = 0,$$

$$C_k^{rs} - C^{rs*} = 0, \quad \text{for all r} \in \text{O, s} \in \text{D and k} \in P_{rs}$$

where f_k^{rs} is the flow assigned to route k, P_{rs} is the set of all routes from origin r to destination s, $C_k^{rs} = \sum_{a \in k} t_a(x_a)$ is the travel time along route k where x_a denotes the traffic flow on link a, $t_a(x_a)$ is the travel time along link a when the traffic flow is x_a, and C^{rs*} represents the minimum travel time from origin to destination (Chow, 2007). Such constraints indicate that in a case where a path carries flow, the travel time on this path equals to the minimum origin-destination (O-D) travel time, while in a case where a path is idle, the travel time on this path is greater than or equal to the minimum travel time in the same O-D pair.

According to the Beckmann's transformation (Beckmann, 1956; Daskin, 1985), the UE is formulated as follows:

$$min \sum_{a \in A} \int_0^{x_a} t_a(w) dw$$

$$s.t. \sum_{k \in P_{rs}} f_k^{rs} = q_{rs}, r \in O, s \in D,$$

$$f_k^{rs} \geq 0,$$

$$x_a = \sum_{r \in O, s \in D} \sum_{k \in P_{rs}: a \in k} f_k^{rs} a \in A,$$

where q_{rs} is the traffic demand in an O-D pair.

The objective function is the sum of the integrals of the link performance functions, in which the first constraint ensures that the flow on all paths connecting each O-D pair is equal to the corresponding traffic demand, which indicates that all O-D pairs must be assigned to the network. The second constraint makes sure that the assigned flow is nonnegative. And the third constraint calculates the traffic flow on each link.

The UE is implemented with the assumption of the fully observed information on the traffic situations for each evacuee. Specifically, it was assumed that evacuees know the travel times of every possible route and they can judge the validation of the optimal routes correctly (Galindo & Batta, 2013; Faturechi & Miller-Hooks, 2014). However, since the disasters and evacuations are typically rare event with unusual traffic demand compared to normal traffic situations (Dixit & Wolshon, 2014), it is impossible for evacuees to figure out the best route for them to minimize their evacuation time (Yazici, 2010; Pel, Bliemer & Hoogendoorn, 2012). In a word, UE is not a realistic approach for evacuation planning.

(b) System Optimal.

The UE merely satisfies the individual driver's behaviors rather than system optimum. However, in an evacuation plan, the authority's goal is typically to "minimize the average journey time" (Wardrop, 1952). For this purpose, the SO model is employed to distribute traffic over the road network to maximize the total benefit instead of the individual one for drivers.

From the perspective of modeling, SO allows to obtain a linear programming formulation as follows (Daskin, 1985):

$$\begin{aligned} & min \sum_{a \in A} x_a t_a(x_a) \\ & s.t. \sum_{k \in P_{rs}} f_k^{rs} = q_{rs}, r \in O, s \in D, \\ & f_k^{rs} \geq 0, \\ & x_a = \sum_{r \in O, s \in D} \sum_{k \in P_{rs}: a \in k} f_k^{rs} a \in A, \end{aligned}$$

The definitions of variables and constraints are the same as those in the UE model mentioned in the above paragraphs. Although a system optimal solution is obtained, in a practical disaster evacuation, people may not accept the recommendation provided by the SO model but choose the routes to reach the closest shelter. It makes the SO less practical in real-world applications.

(c) Nearest Allocation (NA).

Southworth (1991) and Barrett et al. (2000) classified the destination choices of travelers into three categories: nearest safe destination, soonest safe destination, and easiest safe destination. As discussed above, travelers may tend to act selfishly to select routes that take them to nearest route or shelter which may be at nearest distance or reached by shortest free-flow travel time for evacuees. Such a strategy is investigated and implemented by a lot of researchers (Yamada, 1996; Cova, 2003; Coutinho-Rodrigues et al., 2012; Sheu & Pan, 2014). The NA model is reasonable to some extent since the information on path lengths or free-flow travel time is more accessible to evacuees compared to actual travel times during an evacuation (Bayram, 2016). However, system stability and efficiency may not be guaranteed using this traffic assignment method.

(d) Constrained System Optimal (CSO).

CSO refers to the models that consider the individual preferences by imposing additional constraints to ensure that drivers choose the "acceptable" paths only (Jahn et al., 200). Roughgarden (2002) introduced the concept of unfairness into the routing problem. Specifically, the negative impact for travelers is measured as the ratio of the travel time of the recommended route to that of the shortest possible path the driver could have taken. To find the fair and efficient solutions for a route guidance system in every day normal traffic conditions, Li & Zhao (2008) and Zhou & Li (2012) developed the corresponding models, bounds, and algorithms that consider the driver needs.

Bayram et al. (2015) initialized the usage of CSO in the evacuation planning study, where the model aims to evacuate the disaster zone as quickly as possible with a fair assignment of evacuees to shelters and to routes. In their case, they use the *price of anarchy* to measure the impact of selfishness, the lack of coordination among evacuees or the fairness warranted by the evacuation management authority (Bayram, 2016). Roughgarden (2002) proved that the price of anarchy is bounded from above by $\alpha(L)$, and Correa & Stier-Moses (2007) further showed that users in the real networks can still benefit from coordination even if the price of anarchy is not very large.

Bhaskar, Fleischer, and Anshelevich (2015) define the *evacuation price of anarchy* as the ratio of amount flow that reaches the destination by a specific time in the worst equilibrium flow to the amount of flow that reaches the sink in the earliest arrival flow. It depends on the static flows. On the contrary, considering the time-varying characteristics of the traffic flows in evacuation models, dynamic traffic assignment (DTA) models including Dynamic System

Optimal (DSO) and Dynamic User Equilibrium (DUE) are proposed as a modified version of static models originating from CSO (Beckmann et al., 1956). To be specific, Daganzo (1994, 1995) introduced the DTA model based on the cell transmission model, and Ziliaskopoulos (2000) further formulated System Optimum Dynamic Traffic Assignment (SO DTA) as a Linear Programming Problem according to Daganzo's work.

Mathematical optimization, such as mathematical programming formulations, optimal control theory formulations, and variational inequality approaches, may not represent realistic time-dependent dynamic traffic characteristics and driver behaviors, and they may be intractable for realistic size networks (Bayram, 2016). Simulation based models have advantages in replicating real-world situations. The simulation-based evacuation planning is discussed in the next section.

(2) Simulation-based Modeling

The significant challenge in disaster operations management is how to execute the predesignated evacuation plan well regardless of unpredictable and chaotic situations, e.g., there may be massive traffic congestion or increased frequency of traffic accidents during an evacuation process (Na & Banerjee, 2019). Such a problem could be solved by analytical models for finding the optimal route and guaranteeing the solution convergence (Na & Banerjee, 2015). However, the main drawback of these models is that they cannot reflect the real-world traffic dynamics during an evacuation, that is, the rapid movement of evacuees and their interactions on the road.

In the past few decades, a lot of literature utilized the evacuation simulation model as a planning tool in diverse evacuation scenarios. In general, simulation modeling can be classified

into three types: macroscopic, mesoscopic, and microscopic models (Pidd, De Silva, & Eglese, 1996). Macroscopic simulation models take the highest-level view of transportation networks without involving no exhaustive descriptions of individual agents like evacuees or vehicles. The global-view modeling from the macroscopic simulation results in a significant advantage in reducing computational complexity and burden. The simulation based DTA models are commonly used in large-scale evacuation planning problems (Edara, Sharma, & McGhee, 2010; Montz & Zhang, 2015; Songchitruksa et al., 2012; Ukkusuri et al., 2017). However, it is noteworthy that the simulation approaches may not always lead to optimality or convergence and capture behavioral patterns such as choice of destination or compliance rate to evacuation orders (Na & Banerjee, 2019). Different from the macroscopic simulation models, microscopic simulation models capture detailed information of entities and thus allow for exhaustive traffic representation of each evacuee or vehicle. Jha et al. (2006) developed a microscopic simulation model named MITSIMLab to evaluate impacts of emergency evaluation plans under five different scenarios. Chen et al. (2006) proposed the microscopic evacuation simulation model based on VISSIM to estimate the minimum clearance time and the damage scale carried out by a hurricane landfall.

Along with such microscopic simulation-based DTA models, Cellular Automata (CA), Agent-based Simulation (ABS), Discrete Event Simulation (DES), and Agent-based Discrete Event Simulation (ABDES) have been also utilized to model evacuation processes.

(a) Cellular Automata (CA) Simulation.

In a CA simulation model, an evacuee or a vehicle is regarded as an object on a grid that discretizes the space or the roads in a traffic network. However, most of typical CA models

cannot always capture the traffic dynamics of evacuation vehicles in a complex and irregular traffic network in real time (Pelechano & Malkawi, 2008) since the real traffic network cannot be precisely aligned with the lattice.

(b) Agent-based Simulation (ABS).

An ABS model is the other microscopic simulation modeling approach for disaster evacuation problem. It has a superior ability to build a complex system compared to CA models. ABS models involve the agents, an autonomous and intelligent object, behaves, and interacts with each other under the specific rules (Macal & North, 2010). Therefore, ABS models have been extensively applied to model systems composed of agents in evacuation planning (Scheidegger, 2018). For example, Madireddy et al. (2011) investigated several traffic control strategies including contraflow lane reversal based on the ABS model for evacuation traffic management. Because of the increase in computing power for computational tools, multi-agent simulation framework can model the human behavior more realistically (Pan et al., 2007; Yuan et al., 2017) or can be used to examine a large-scale pedestrian evacuation process (Lammel et al., 2010; Wagner & Agrawal, 2014). In other words, the ABS models enable the representation of evacuation behavior which affects the characteristics and dynamics of the traffic network such as traffic flow and traffic congestion (Yin et al., 2014; Ukkusuri et al., 2017). To accurately obtain the behavior pattern of agents, the ABS model checks and updates agent state frequently.

(c) Discrete Event Simulation (DES).

In comparison to ABS which models the agents and their interactions, DES follows a top-down approach to model evacuation traffic network which is similar to the real model (Siebers et al., 2010). ABS models and DES models can achieve similar performance under sufficient

memory and computational time (Chan et al., 2010). Nevertheless, the ABS models with time-driven scheduling approach require intensive and expensive computational resources to reach a similar performance as stemmed from DES models with the event-based scheduling method (Zhang et al., 2014). Therefore, DES models can achieve reasonable performance with relatively inexpensive computing costs compared to ABS models. Discrete Event System Specification (DEVS) is a modular and hierarchical framework arising from the mathematical dynamical system theory for building a DES model (Zeigler et al., 2000). However, DES is still an inaccurate simulation model because it neglects the movement and interaction of intelligent entities.

(d) Agent-based Discrete Event Simulation (ABDES).

ABDES considers advantages of both ABS and DES models and models an evacuation process with complex traffic dynamics derived from various interactions between agents and variations of traffic conditions during an evacuation. It involves a large number of agents to cover operational flexibility and computational efficiency. ABDES models are not exactly the same as the conventional ABS and DES models since ABDES models are developed based on the event-based scheduling approach instead of time-driven scheduling used by conventional ABS models. Moreover, they have autonomous and intelligent agents with specific rules for movement that are not comprised in traditional DES models. For an evacuation transportation network, an ABDES modeling based on the parallel DEVS with grid- and coordinate-based continuous simulation space is proposed by Wagner (2004). In addition, taking into account the potential benefits of GIS due to the spatial nature of most evacuation models, a few researchers

attempted to combine simulation models with GIS to investigate the feasibility for evacuation (Wu et al., 2008,; de Silva & Eglese, 2000; Na & Banerjee, 2014).

(3) Reinforcement Learning (RL)

As mentioned above, DTA typically consists of two components, namely a route choice and/or departure time choice model determining inflow to each link and a dynamic network loading model propagating traffic flows through the network (Bliemer et al., 2017; Yu et al., 2020). However, such a unified paradigm could be challenging to accommodate the substantive amount of fine-grained information in terms of time thanks to emerging technologies like GPS and mobile phones (Shou & Di, 2020). Therefore, it prompts a paradigm shift from model-driven to data-driven routing that leverages high-resolution trajectories, for the sake of better modeling one's en-route adaptive behavior in a dynamic and competitive environment.

Along this direction, Reinforcement Learning (RL), particularly Multi-Agent Reinforcement Learning (MARL) (Hu & Wellman, 1998), could be a promising approach for dynamic routing games for evacuation planning with a large number of inter-acting agents such as evacuees or vehicles on traffic networks. Furthermore, with the introduction of Deep Learning or Deep Neural Networks in the RL, deep reinforcement learning (DRL) such as deep Q networks (Mnih et al., 2015) can now tackle a large-sized problem with a very large or even continuous state and action space. In brief, MARL is advantageous over the classical model based DTA framework regarding computational efficiency with a large number of agents on large-sized traffic networks, and real-time adaptiveness of modeling agent's behaviors and dynamic environments. We will review the literature on traffic assignment of single-agent and then move to that of multi-agent in the next section.

(a) Single-Agent Traffic Assignment.

To evacuate efficiently, evacuees need to choose a sequence of traffic segments from an origin to a destination. Therefore, such an en-route choice is inherently a sequential decision-making process, in which at each junction one decides the next link to take (Shou & Di, 2020). Markov Decision Process (MDP) is typically used to model such a sequential process, where drivers are rational agents to optimize a prescribed objective function (Kim, 2005; Kim, 2016). In general, we always assume every driver is a self-interest agent who aims to only maximize individual accumulative rewards or minimize individual cost or maximize payoffs. A single-agent Q-learning approach for adaptive routing problem in stochastic time-dependent network is proposed in Mao and Shen (2018). The authors utilized some global information such as congestion status in the definition of state and derived an optimal policy for the agent based on bootstrapped traffic profiles from the PeMS dataset. One of the main drawbacks of single-agent RL is that it fails to capture the competition among adaptive agents.

(b) Multi-Agent Traffic Assignment.

Multi-agent selfish routing games have been studied using both prescriptive and descriptive approaches. In detail, prescriptive approach answers the question “how one should behave in route selection”, and it can be classified as the model-based model since its high reliance on models that prescribe route selection behaviors. Descriptive approach answers the question “how one actually selects routes”, and it can be categorized as data-driven being accountable for real-world routing data (Shou & Di, 2020).

Based on the definition of action of an agent, we can broadly categorize MARL-based route choice models into two groups: route-based and en-route. In the first group, the action is

the route (Zhou et al., 2020; Ramos et al., 2018; Stefnello et al., 2016). For instance, Ramos et al. (2018) and Stefnello et al. (2016) assumed that every agent does follow his pre-selected route until he/she reaches his/her destination and let the action space for him/her to be the *k* shortest path from the origin to destination. However, drivers could deviate from the original route plan when they realize that traffic conditions on some alternative routes might be better, especially when they get stuck in traffic on the current route. To model such an en-route adaptive behavior of real drivers, the second group of researchers paid attention to en-route traffic assignment, in which the action space of an agent currently at a node is the outbound links from the node (Grunitzki, 2014; Buzzan & Gruntizki, 2016), namely each agent needs to decide which outbound link to select whenever the agent arrives at a node, until the agent reaches some terminal node.

Although these studies model multi-agent interactions, they only used independent multi-agent tabular Q-learning in which each agent was treated as an independent learner who has no information of other agents. The major issue of unguaranteed theoretical convergence arises when we adopt Q-learning since the environment is no longer Markovian and stationary (Matignon, 2012; Nguyen, 2020). Moreover, when the agent number increases largely, the learning becomes intractable due to the curse of the dimensionality and the exponential growth of agent interactions (Yang et al., 2018). To fill these research gaps, Shou and Di (2020) developed a multi-agent deep Q-learning approach which captures the interaction among adaptive agents via a flow-dependent mean action.

Air Mobility Indicator

The dispatched evacuation flights usually have a negative impact on the air route network since it adds extra work burden for air traffic control tasks. We quantified the impact by a term called Air Mobility Indicator, in order to facilitate the evacuation planning via air traffic. However, based on the review of previous studies, it was found that there has been no literature thus far that tries to address the flight dispatch during a hurricane evacuation. In the following section, we will focus more on discussing the potential variables/factors that may influence the efficiency of evacuation by flight. The air mobility indicators can be categorized into two major groups. The first one is the airspace complexity and the second one is the airport ground capacity. (As a matter of fact, these factors are dependable, but we usually employ them solely in reality.)

NASA Ames Research Center (ARC) (Laudeman et al., 1998) developed a measure of both en route and terminal airspace complexity, namely the Dynamic Density (DD). The calculation and prediction of DD has been identified as a key need for assessing workload of future traffic (RTCA Force, 1995; Mogford et al., 1995). The DD was conducted based on interviews and survey techniques with input from 65 qualified air traffic controllers. The significant contributions to the workload which are identified by controllers are Traffic Density (i.e., the number of aircraft in a sector), the number of aircraft undergoing trajectory change (i.e., heading, speed or altitude changes), and the number of aircraft requiring close monitoring due to reduced separation (Sridhar et al., 1998). As more DD variables were developed or validated, Kopardekar (2000) provided a unified review of DD variables to date. Arnab & Washington (2002) illustrated that airspace capacity is determined by controller workload, which is in turn

affected primarily by the air traffic and ATC sector geometry features (a.k.a airspace capacity drivers). The authors further analyzed the results from simulations of five different areas using the European Airspace Model (EAM) and pointed out that the cruise profile affected controller workload more. The evolution toward an autonomous aircraft framework envisages new tasks where assessing complexity may be valuable and requires a whole new perspective in the definition of suitable complexity metrics. Therefore, Prandini et al. (2011) presented a critical analysis of the existing approaches for modeling and predicting air traffic complexity, examining their portability to autonomous air traffic management systems. Despite of quantification of controller's workload which has been done, Gianazza (2017) indicated that these indicators are either difficult to interpret or subject to biases. To tackle these drawbacks, they proposed using historical records of the past sector operations to quantify the workload. To be specific, they assumed the workload is low when a given sector is collapsed with other sectors into a larger sector, normal when it is operated as it, and high when it is split into smaller sectors assigned to several working positions. However, all mentioned work did not open-source the dataset that limits the further development of relevant applications by other researchers.

Efficient airport ground operations are another one of the key aspects towards sustainable air transportation. This complex system includes elements such as ground movement, runway scheduling and ground services (bus scheduling) (Weiszer et al., 2015). Some attributes are described in the following table. We do not consider these variables in our study because we do not have access to the ground operations dataset.

	Description		
g_1	The total time objective	$\tau(origin, destination)$	Travelling time of the bus between *origin* and *destination*
g_2	The fuel consumption objective	N	Set of active bus trips
g_3	The bus scheduling cost objective	n	The number of bus trips
M	Set of all aircraft	$G = (O, P)$	Vehicle scheduling network
m	The number of all aircraft	O	Set of nodes on the vehicle scheduling network
A	Set of arriving aircraft	P	Set of arcs on the vehicle scheduling network
D	Set of departing aircraft	c	Cost of arc
$V(v_i, v_j)$	Function to calculate the wake vortex separations for weight categories v_i and v_j of leading aircraft i and following aircraft j	b	Variable determining if arc is covered by a bus
v	Weight category	k, l	Bus trip index
i, j, e	Aircraft index	$pop_{\max}$	Maximum number of generations
r	Actual landing time for arriving aircraft/take-off time for departing aircraft	pop_{number}	The current generation index
d	Arrival time of the departing aircraft at the runway holding point	c^a	The total strategic cost
w	Waiting time at the runway	C^{fuel}	Fuel cost
t^{rwy}	The total runway delay	C^{total}	The total cost
f^{rwy}	The total runway fuel	R	Rated output of aircraft
$\phi_{v_i}, \phi_{v_i}^{idle}$	Calculated and idle fuel flow for weight category v_i, respectively	h	Weight
δ	Safety time distance between taxiing aircraft	a, a_{max}	acceleration and maximum acceleration of aircraft respectively
y	Integer representing the speed profile	η	Thrust level
q	The shortest taxi route	Thr	Thrust
s	Stand	$weight$	Weight of aircraft
$T(q_i, y_i)$	Travel time of aircraft i taxiing on route q_i for given speed profile y_i	FR	Rolling resistance
x	The pushback time	p_{arr}	Arrival time of buses after arriving aircraft came to a stand
z	The arrival time to the stand	p_{emb}	Embarking/disembarking time of passengers
$F(q_i, y_i, v_i)$	The amount of fuel burned for aircraft i of weight category v_i during taxiing on the route q_i following the speed profile y_i	p_{dep}	Departure time of buses before the pushback time of departing aircraft
t^{taxi}	The total taxi time	$p_{headway}$	Headway between buses
f^{taxi}	The total fuel burned during ground movement	sp	Average speed of buses

Table 1. Variables on Airport Ground Operations Optimization (Weiszer et al., 2015).

Research Gap Analysis

Here we will discuss the limitation of the aforementioned traffic assignment models for evacuation planning via air traffic and analyzing the possible research gaps in the evacuation planning via air traffic if we apply the existing methods for the evacuation planning via ground transportation directly. Different from the evacuation by car, taking a flight to evacuate is managed by air transportation authorities. In other words, the UE model is not fit for evacuation planning by air. However, the "selfish" behaviors still exist in the planning. For example, the supporting airports do not want too many evacuation flights to take off from or arrive at them since it will cause an excessive work burden. The second difference is the network structure. The en-route model does not work well sometimes because the spatial node is hard to be defined

as the fixed point in the airspace and the actual flight trajectories may greatly deviate from their pre-departure plan (Liu & Hansen, 2018). Hence, we choose route-based modeling approach in our study.

Based on the literature, mathematical optimization cannot capture the real dynamics of the environment, and simulation-based model is not time-efficient because we must define various specific rules and validate them by simulation. RL may be a better choice to overcome this issue. Previous work using RL assumes that agents interact with the online environment actively and learn by the collected experience (data). However, the application based on such a RL approach is difficult to be implemented in practice because of the horrible sample inefficiency and the unexpected agent's behaviors. The second constraint is about the lack of high-fidelity simulator of the air route network which leads to the fact that the interactions are not defined accurately and credibly. The other limitations are the unguaranteed convergence and high computational complexity of RL algorithms especially multi-agent reinforcement learning (MARL) algorithms. For instance, each agent (aircraft in our problem setting) has different action spaces in different location-time states. One aircraft locating at airport A (suppose A is the evacuation airport) cannot choose the flight transaction from airport B to airport C. And the en-route aircraft should not be assigned with the other transaction unless an exception occurs. We call this kind of agent setting the multi-class agent RL where the action spaces are changing over time. If we adopt global view setting which models the entire dispatching system to alleviate the multi-class action space problem, the action space would be large in large-scale dispatching since the system must assign each aircraft to each transaction including not only spatial information but also temporal information. Therefore, we still need to model independent aircraft as an agent.

Considering these gaps, we propose the corresponding solutions as follows:

a) To handle the changing action space in the RL and let the assignment be fair for all participating airports. We formulate the flight assignment as a matching problem, specifically, maximum weighted graph matching problem. The agents are regarded as the one such that they share an identical action space – serve an evacuation flight or keep standby status. From another perspective, it is not required to consider algorithmic convergence. On the contrary, if we adopt the conventional RL paradigm, the agent may select the same action or the same flight transaction at the same time. Such a fully decentralized structure would suffer from low computational efficiency and non-convergence in general.

b) The real-time interaction between the agent and the environment is hard to model because there is no existing simulator for air traffic management in the market. As an alternative, we introduce the concept of the offline-data-driven reinforcement learning that utilizes only previously collected offline data, without any additional online interaction (Kumar et al., 2019; Fu et al., 2020). The essential difference between our application and the classical usage of offline RL is that we employ a dataset collected by a default behavior policy to estimate the global impact of each dispatched evacuation flight instead of learning the policy itself.

c) To tackle the uncertainty problem in evacuation process, we develop machine learning based predictors for air mobility indicators, which are used to calculate the instant reward in the online planning phase.

In the following sections, we will define the problems and solutions for these problems.

Problem Statement

Our dispatching problem is an extensive equivalence class of combinatorial problems, the bipartite graph matchings. To find the best matching (action) in a global view, we firstly should evaluate the value of an individual matching, which is indicated by a weight between an aircraft-airway pair, and then we formulate the dispatch's objective function as follows:

$$\underset{a_{ij}}{argmax} \sum_{i=0}^{m} \sum_{j=0}^{n} W(i,j) a_{ij}$$

$$s.t. \sum_{i=0}^{m} a_{ij} = 1, j = 1,2,3 \dots, n$$

$$\sum_{j=0}^{n} a_{ij} = 1, i = 1,2,3 \dots, m$$

$$a_{ij} = \begin{cases} 1 & \text{if flight } j \text{ is assigned to aircraft } i \\ 0 & \text{if flight } j \text{ is not assigned to aircraft } i \end{cases}$$

where $i \in [1, \dots, m]$ corresponds to all available aircraft at a time slot, while $j \in [1, \dots, n]$ represents evacuation flight transactions to be served. $W(i,j)$ estimates the expected return of aircraft i performing an action of serving transaction j. Note that in the case with $i = 0$ and $j = 0$, the dispatch system "do nothing" in this timestamp.

It is worth noting that the agents have a finite action space in that they must choose one of available evacuation flight transactions to serve or standby. The constraints guarantee that each aircraft will select only one of the available actions, while each flight could be assigned at

most one aircraft of being unserved at this time slot. In practice, the weights reflect the influence of the air route network caused by the evacuation flights, which is composed of (1) an instant reward for the aircraft serving this transaction attained from real-time information, and (2) an additional term representing the impact of this decision into the future. In our problem setting, the short-term influence is measured by some air mobility indicators such as air flow, flight delay and controller's workload. We regard the dispatch as a Markov Decision Process (MDP), and the long-term influence could be estimated or learned based on sequential short-term influences using a RL approach. Now let us assume air mobility indicators are independent of each other for simplicity in modeling real-time dynamics, we can apply machine learning algorithms to forecast them at a time slot, which are used to calculate immediate influence or reward.

Approach and Methodology

In this section, we present the details of the modeling procedure based on machine learning techniques. To be specific, we firstly describe the datasets which are readily accessible at Embry-Riddle Aeronautical University for modeling. Then, the hybrid machine learning models are proposed to predict the air mobility that includes air flow, flight delay, taxi time, etc. The predicted results are further used to provide the relatively accurate feedback for a flight in the complex and dynamic air route network. Finally, the reinforcement learning based flight dispatch framework is derived to solve the flight planning problem. The overview of our framework is illustrated in Figure 3.

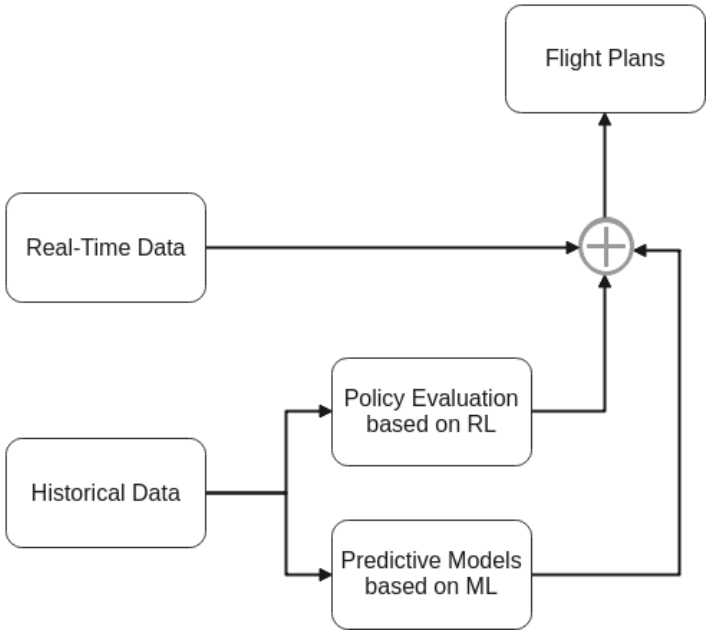


Figure 3. Flowchart of flight dispatching system.

Data Pipeline

We perform data gathering, data cleansing, data integration, and data visualization for the dataset as follows. The integrated data includes flight information (Airline On-Time Performance Data, AOTP), aircraft information (Automatic Dependent Surveillance-Broadcast, ADS-B), human factors (air traffic controllers, ATCO), and aviation weather information.

Data Sources and Gathering

AOTP (US DOT, 2021) collects on-time reports for flights of the reporting carriers operate. The on-time performance data contains on-time arrival and departure data for non-stop domestic flights by month and year, by carrier, and by origin and destination airport. AOTP

includes scheduled and actual departure and arrival times, canceled, and diverted flights, taxi-out and taxi-in times, causes of delay and cancellation, airtime, and non-stop distance. The AOTP is mainly used to train a departure performance or an arrival performance prediction model. To be specific, the performance could be flight delay, taxiing time and the elapsed time of flight. Therefore, some attributes in this dataset are not necessary depending on our goal such as cancellations and diversions, cause of delay, gate return information at origin airport, and diverted airport information, and they are removed prior to training machine learning models.

ADS-B is the latest technological leap in airspace surveillance, which provides higher accurate positional information of an aircraft than conventional radar to ground controller and other pilots directly. ADS-B is seen as a replacement for secondary surveillance radar because of significantly cheaper installation of ADS-B ground stations and its wide coverage. This gives air traffic controllers the potential to reduce the required separation distance between aircraft that are ADS-B equipped (Zhang, 2020). ADS–B data used in this project comes from ADS-B receivers of ERAU's NextGen Applied Research Laboratory (NEAR) located at the Daytona Beach campus and Prescott campus. The original data collected by NEAR are ASTERIX Category (CAT) 033 messages that contain 17 attributes such as latitude/longitude, velocity, aircraft ID and flight number. A target report construction example of CAT 033 is shown in Figure 4 below.

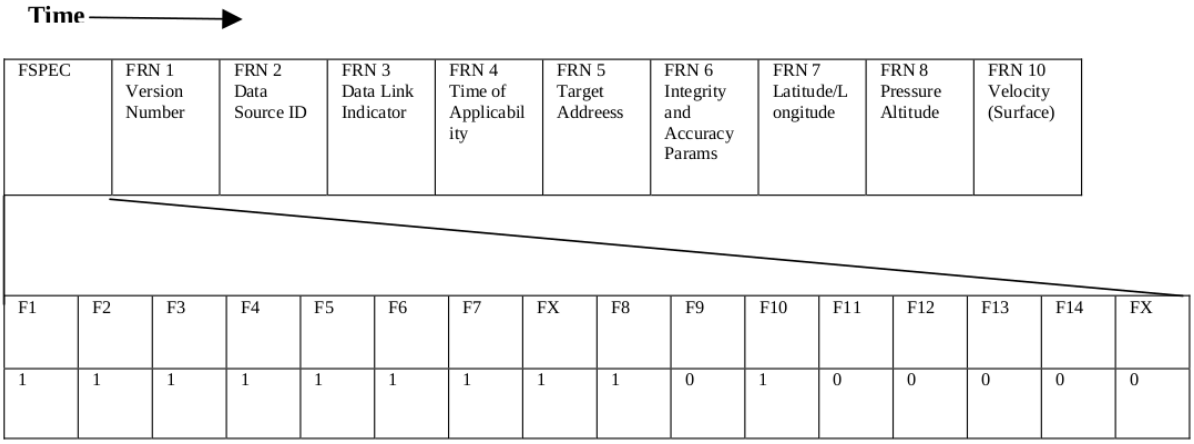


Figure 4. Get system specification ASTERIX CAT 033 (EUROCONTROL, 2019).

The Field Reference Number (FRN) establishes the order of the items in the FSPEC, and along with the Category code serves to uniquely identify each data item. We decoded and extracted 15 most relevant/useful information from CAT033: (1) version number; (2) data source identifier; (3) link technology indicator; (4) time of applicability; (5) target address; (6) integrity and accuracy parameters; (7) latitude/longitude; (8) pressure altitude; (9) velocity (airborne); (10) velocity (surface); (11) mode 3/A code; (12) target identification; (13) emitter category; (14) target status; and (15) geometric altitude.

ATCO data, which was retrieved from 123ATC.com, has 11 attributes: control towers ID, facility name, state, facility level, always open, certified professional controller (CPC) target, current staffing, projected staffing, average training time (years), training success rate, and maximum salary.

Two weather datasets, which were gathered and cleansed in this project, include (a) Local Climatological Data (LCD) from the National Oceanic and Atmospheric Administration (NOAA), and (b) the High-Resolution Rapid Refresh (HRRR) which is the latest version of the

Rapid Refresh (RAP) numerical weather model running by the National Centers for Environmental Prediction (NCEP). HRRR generates data down to a 3-km resolution grid every hour so users can analyze the weather conditions for quite small regions of interest. Contrary to LCD which records only ground weather situations, RAP is composed of various attributes at different height levels. Besides, the RAP data is in the form of GRIB2 file that is a WMO format for gridded data. To convert such a file to a more structured format, we employed *wgrib2*, a utility to read and write grib2 files, to export .csv file. The flight-related weather data involves temperature, precipitation, humidity, sky conditions, wind speed, wind direction, and so on.

Exploratory Data Analysis

Exploratory Data Analysis (EDA) refers to the critical process of performing initial investigations on data to discover patterns, to spot anomalies, to test hypothesis, and to check assumptions with the help of summary statistics and graphical representations. This step is necessary as it deepens the people's understanding of this data and helps with further potential exploration and prediction (Jiang, 2020). Some prominent features are described below in a detailed manner with the effect of variation on flight delay using R.

EDA for AOTP data: there are 4253524 flights in this dataset with the range of flight delay from –87 to 1660 minutes, and 4251121 flights have arrival delay less than 500 minutes. Therefore, the distribution of the flight delay is mostly distributed in the range of fewer than 500 minutes as shown in Figure 5. It is clear that the majority of the cases belong to non-delay class and the overall flight delay fits long-normal distribution. Airline is one feature that would be used to predict the flight delay as the operating mechanism of different airlines may cause issues like slower airplane fueling and maintaining, slower baggage loading which takes up more time.

The dataset contains 12 different airline companies. We compared the average delay with a standard error of the mean as shown in Figure 6. This figure clearly shows that Spirit Airlines, JetBlue Airways, Frontier Airlines and Virgin Atlantic Airways are more likely to have delay issues with the longer average delay, while corresponding deviations are also relatively larger at the same time which indicates a bit more unstable delay situation. Besides, Alaska Airlines, Delta Airlines and Hawaiian Airlines have obviously fewer average delays and are likely to have obviously fewer average delays and are likely to have earlier arrivals than

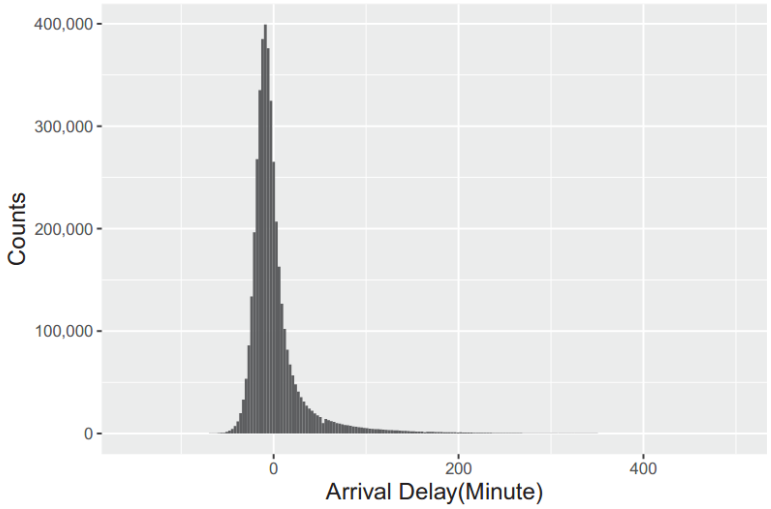


Figure 5. The distribution of flight delay.

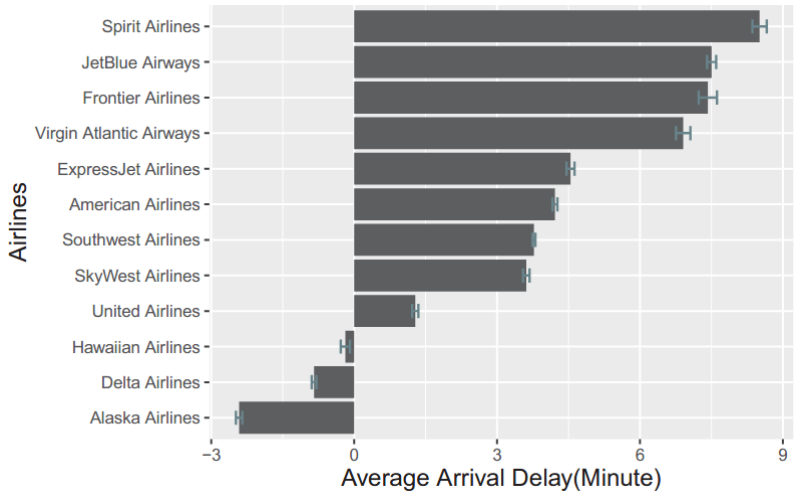


Figure 6. Flight delays in different airlines.

scheduled ones. Therefore, the percentage of delay is likely to vary with airline company. Moreover, the state of departure airport is another feature that needs to be studied for flight delay as shown in Figure 7. The tree-map is used where the area of each rectangle stands for the number of flights and the color indicates the average arrival delay in minutes. It is clear that the states with the top five largest number of flights are more likely to have longer delays such as Florida, Texas, California and Illinois, while the flights from Georgia still perform better in delays even with a large number of flights. Besides, states like Alaska, Hawaii and Montana obviously have less delay time with relatively fewer flights. These patterns prove that the flight delay does vary with the state of departure airport.

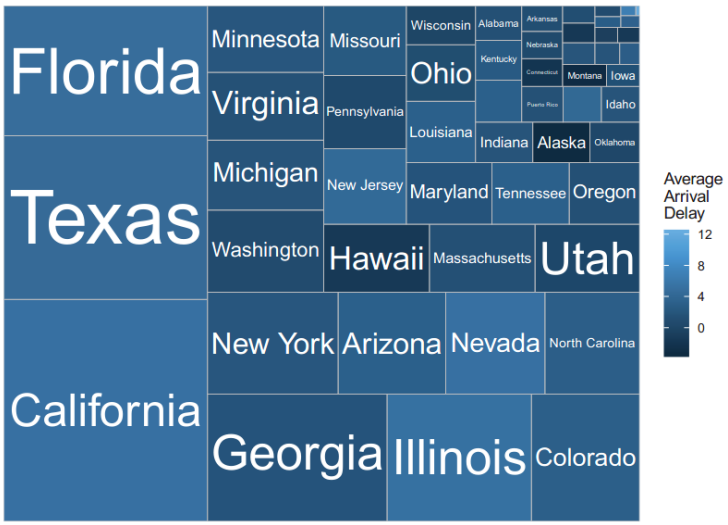


Figure 7. Flight delay and number of flights in different states.

Feature Engineering - Data Cleansing and Integration

As a preprocessing step, feature engineering refers to the process of using knowledge and statistical approaches to select and create representative features (Zhang, 2020). We aim to construct cleansed datasets with relatively high-dimensional feature space after feature engineering, so it enables the trained machine learning model to have better performance on unseen data.

There are some general rules of feature selection for all types of data. Specifically, (a) firstly, features with the percentage of missing values that exceed the threshold (80%) were removed; (b) secondly, we calculated the Pearson correlations of each pairwise variable; and (c) then removed redundant features beyond the threshold (80%); (d) among the remaining features, the presumed useful ones were selected according to our domain knowledge and task needs. For example, Table 2 presents part of attributes with a high number of missing data in the LCD dataset, which were then removed before model training. Besides, we removed attributes with

zero variance or relatively low variance, and pairwise correlation beyond threshold according to Figure 8 that depicts correlation matrix of some highly correlated attributes.

Feature Name	Missing Rate
SkyConditionFlag	99.80%
VisibilityFlag	99.90%
WeatherType	92.81%
WetBulbFarenheitFlag	1000%
ValueForWindCharacter	94.18%
PressureTendency	100%
PressureChange	100%
SeaLevelPressure	80.64%
HourlyPrecip	95.71%
ValueForWindCharacter	94.18%

Table 2. Part of attributes with high number of missing data.

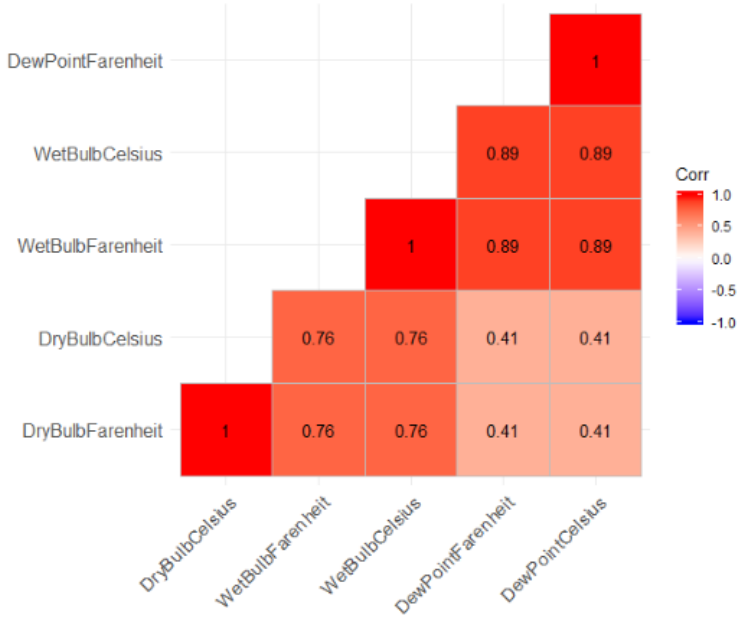


Figure 8. Correlation matrix of temperature features.

Moreover, we adopted a universal encoding strategy for discrete and categorical features as shown in Figure 9. More specifically, the discrete and categorical variables were split up into

two types – high-cardinality and low-cardinality based on the threshold of 50. Then, we encoded the features using frequency encoding and one-hot encoding, respectively. One-hot encoding enables the representation of the discrete feature to be mapped into Euclidean space, and a certain value of the discrete feature corresponds to a point in the Euclidean space so that it is more reasonable to calculate the distance between them. However, it would cause the curse of dimension if we encode high-cardinality features using one-hot encoding. Hence, we used frequency encoding, which counts and sorts the occurrence of values, to address this issue.

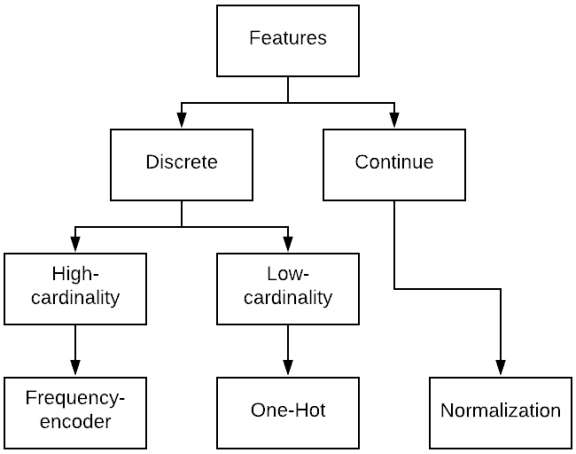


Figure 9. Encode and normalize features.

In more detail, frequency encoding sorts the categories according to the frequency of each category within the feature in the training set. Compared with standard frequency encoding which gives the same encoding to different categories having the same frequency, our method has a specific advantage that it is not sensitive to outliers. Normalization, or feature scaling is important for some machine learning algorithms where Euclidean distance is used for calculating objective functions. That is because some features with a broad range of values are likely to dominate the objective functions so that the model cannot work properly. Moreover, it also

facilitates the convergence of models using gradient descent for updating weights. Three normalization methods were considered:

(1) **Min-Max Scaler** is the simplest method which rescales the original feature range to [0,1]. The formula is shown below.

$$x' = \frac{x - min(x)}{max(x) - min(x)}$$

(2) **Standard Scaler** makes the values of each feature in the data have zero-mean and unit-variance by subtracting the mean and being divided by standard deviation as shown in the below formula.

$$x' = \frac{x - \bar{x}}{\sigma}$$

(3) **Normalizer** scales each component of a vector to unit length by dividing individuals by the Euclidean length of the vector as shown in the formula below. It is worth noting that in this case, this normalizer directly performs to each record instead of each feature.

$$x' = \frac{x}{\|x\|}$$

After the data was cleansed, we need to integrate these multi-source datasets. Considering all of them are tabular data including spatial and temporal indexes, we used the spatio-temporal as the matching indicator. In addition to existing features in the datasets, we defined the congestion indices of ground and airspace based on flight data and trajectory data, respectively. The ground congestion index was computed by counting departures and arrivals within a time bin. In a similar way, congestion index of airport airspace (low-attitude) can be obtained but we used ADS-B data because it contains both commercial airplanes and private jets that make the

results more accurate. To be specific, the flight records whose aircraft-to-airport distance is less than 200 KM and altitude ranges from 1,200 to 10,000 feet above the mean sea level (MSL) were counted. The distance was obtained by Haversine formula given the longitude and latitude of an airport and an airplane:

$$d = 2r\,sin^{-1}(\sqrt{sin^2(\frac{\Delta\phi}{2}) + cos(\phi)\,sin^2(\frac{\Delta\psi}{2})}$$

$$\Delta\phi = \phi_2 - \phi_1, \Delta\psi = \psi_2 - \psi_1$$

$$cos(\phi) = cos(\phi_1)\,cos(\phi_2)$$

where d is the great-circle distance between two points on a sphere with longitude and latitude (ϕ, ψ) and r is the radius of the Earth. Besides, we split the airspace at 18,000 feet above MSL into equal-sized sectors and the number of aircraft in each sector in a time bin is defined as the congestion index of en-route airspace.

Air Mobility Prediction

Originally, air mobility referred to the ability to transport military troops and supplies in and out of combat areas by means of aircraft (Tolson, 1973). In recent years, this concept has been gradually introduced to the aviation industry, which can be described as a safe and efficient aviation transportation system. Moreover, air mobility in aviation is an important research topic toward the development of Advanced Aerial Mobility (AAM) (NASEM, 2020), a newly emerging field that aims at safe and responsible operation in an integrated National Airspace System (NAS). Air mobility in aviation is crucial to multiple stakeholders including passengers, airlines, airports, and air traffic authorities, as a deterioration of air mobility can lead to

significantly undesired consequences like severe flight delays, cancellations of flights, and congestion of airports, thus disrupting the aviation system and resulting in a huge economic loss. Therefore, accurate estimates of air mobility are requisite for modeling a reliable air transportation network in dynamical environments. In the following sections, multiple machine learning models are proposed to predict air mobility such as airflow/airspace capacity and flight/taxi delay.

Airflow/Airspace Capacity Prediction

A smoother and more robust mechanism to avoid overloads and maximize the use of airspace is necessary because of increasing customer demand and number of aircraft that have brough higher complexity to air traffic management nowadays. Airspace capacity estimation is vital for decision-making of route design and flight dispatches. However, traditional airspace capacity model is either based solely on handoff workload and fixed procedural constrains using queueing principle such as Monitor Alert Parameter (MAP) (Welch, 2015) or a weighted combination of air traffic density and task-based controller workload like Dynamic Density (Laudeman, 1998). However, it cannot reflect the real-world situation because of limitations on the underlying assumption of mathematical model (Richardson, 1979) and a limited number of parameters. Therefore, we firstly applied machine learning algorithms to evaluate airspace capacity, as measured by the total airflow in the airspace. Moreover, human factors and climate information were considered to build tree-based models which provide importance scores for each variable.

Classification and Regression Tree (CART) is a classical supervised learning method. For regression problems, we stratified the predictor space into multiple distinct and no-overlapping

regions and then the predicted value in a specific region is represented by mean or mode of the corresponding training observation. To split a predictor space, we need to find the regions $R_1, R_2, \dots, R_M$ that minimize $J = \sum_{i=1}^{M} \sum_{j \in R_m} (y_j - \widehat{y_{R_m}})^2$ where $\widehat{y_{R_m}}$ denotes average values of the training observations within the m-th region. Ensemble methods such as bigging and boosting help improve machine learning results by combining a few predictors into an even better one. The aggregation function is typically the average for regression. Random Forest is one of the bagging methods in which each tree (predictor) uses the same algorithm, but training set is sampled randomly with replacement (bootstrapping).

Flight/Taxi Delay Prediction

Accurate prediction of aviation delay for commercial flights is a key component to improve safety, capacity, efficiency in air traffic management, and airline business. However, as a dynamic system with uncertainties, civil aviation faces multiple unexpected events frequently that can result in flight delays or cancellations. In the case of the flight delay propagation between initially delayed flight and flights downstream, the impact can grow exponentially if the air traffic control cannot adopt an appropriate resource re-allocation strategy to optimize mobility in time. Hence, estimating delay as accurately as possible within a controllable time window is critical. To achieve this goal, we provided a forecasting framework with both micro and macro level prediction models. Implementation details are described as follows:

(1) Micro-level flight delay prediction based on convolutional neural network

We firstly evaluated four common machine learning models, namely support vector machine (SVM), decision tree (DT), random forest (RF), and multilayer perceptron (MLP) using

the fused dataset of LCD data and AOTP data. In the training stage, k-fold cross validation (k=5) was employed, where the training stage was divided into k folds, then each fold was treated as a validation once while the k-1 remaining folds from the training set. As for hyper-parameters tuning, a grid search method was applied combining with aforementioned k-fold cross validation so that a set of relatively good hyper parameters can be selected after validation, and the best model with its parameters can be evaluated by the test set. Table 3 evaluates the performance of models with different scaling methods. The results of using Min-Max scaler are left out since the accuracies are low in general with the best result that is up to 63%. It is clear that normalizer actually has better performance on SVM, DT, and RF models by a large amount increase of accuracies compared with the standard scaler. However, the standard scaler works better on MLP which is based on neural network method and the increase of accuracy is marginal. To summarize, MLP has better adaptability on both scalers. The feature encoding methods are also evaluated as shown in Table 4. It is clear that the models using all of three

Model	Standard Scaler	Normalizer
Support Vector Machine	73.98%	**80.14%**
Decision Tree	71.24%	**76.42%**
Random Forest	73.95%	**80.21%**
Multilayer Perceptron	**89.07%**	87.83%

Table 3. Prediction accuracy and feature scaling

Model	LabelCount	Binary	Hybrid
Support Vector Machine	**80.14%**	78.36%	79.15%
Decision Tree	76.35%	76.38%	**76.42%**
Random Forest	**80.21%**	78.87%	79.33%
Multilayer Perceptron	**89.07%**	88.55%	88.82%

Table 4. Prediction accuracy and feature scaling

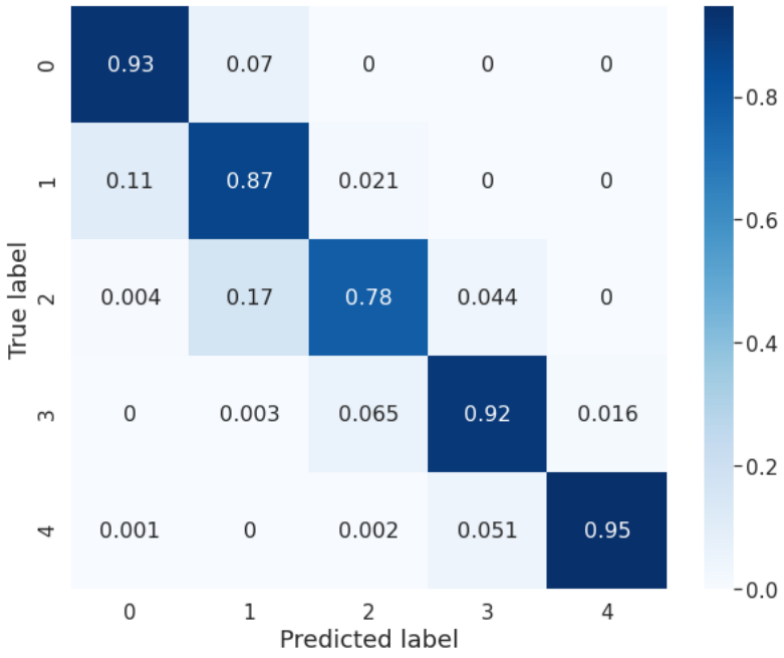


Figure 10. Confusion matrix of flight delay prediction

chosen encoding methods have good accuracies on prediction with minor differences. In general, LabelCount encoder has a better performance on tested models than that of others because of the highest prediction accuracies on SVM, RF, and MLP, while hybrid encoder suits DT best. It is worth noting that the binary encoder shows an increase of feature dimension without deteriorating much on the accuracy, which might be due to its characteristics separating categories well on a high dimensional feature space. According to the above evaluation, the conclusion can be drawn that MLP achieves the best results among all choices of feature engineering compared with other machine learning models, where the highest prediction accuracy reaches 89.05%.

Figure 10 presents the confusion matrix that evaluates the performance of the built model with the existence of skew data. It is obvious that the prediction model works well on classifying non-delay class, as well as high delay and severe delay classes with at least 92% accuracy. For

the slight delay class where a delay is less than 60 minutes' arrival, the performance is also good enough with 87% prediction accuracy. However, for medium delay class, the prediction model shows relatively weak ability distinguishing it from the slight delay class with only 78% accuracy. In general, the model is most likely to classify the delay class to the true one or adjacent ones, which suggests good robustness on prediction tasks.

From the above results and evaluations, it can be easily found that the neural network method has a significant advantage on this dataset since its best accuracy is 8.86%, higher than that of second-best one. It also suggests a potential to use deep neural networks for this task. Conventional neural network is one of the deep learning methods which are most used in visual images tasks with satisfactory results. This method uses many filters to extract spatial information and construct feature maps for the next layer of network so that it can capture high dimensional features, which is suitable for complex tasks. Inspired by this idea, we treated features as a pattern and rearranged its shape as a map so that it can be used as the input of convolutional neural networks (CNNs), expecting that useful patterns can be found by the built model. For feature engineering, standard scaler was used due to the better performance on neural network models and LabelCount encoder proved to be the only encoding method suiting this model due to a large increase of accuracy (over 23%) compared with others in experiments. This actually makes sense since CNNs rely on the spatial characteristic of data, while encoding

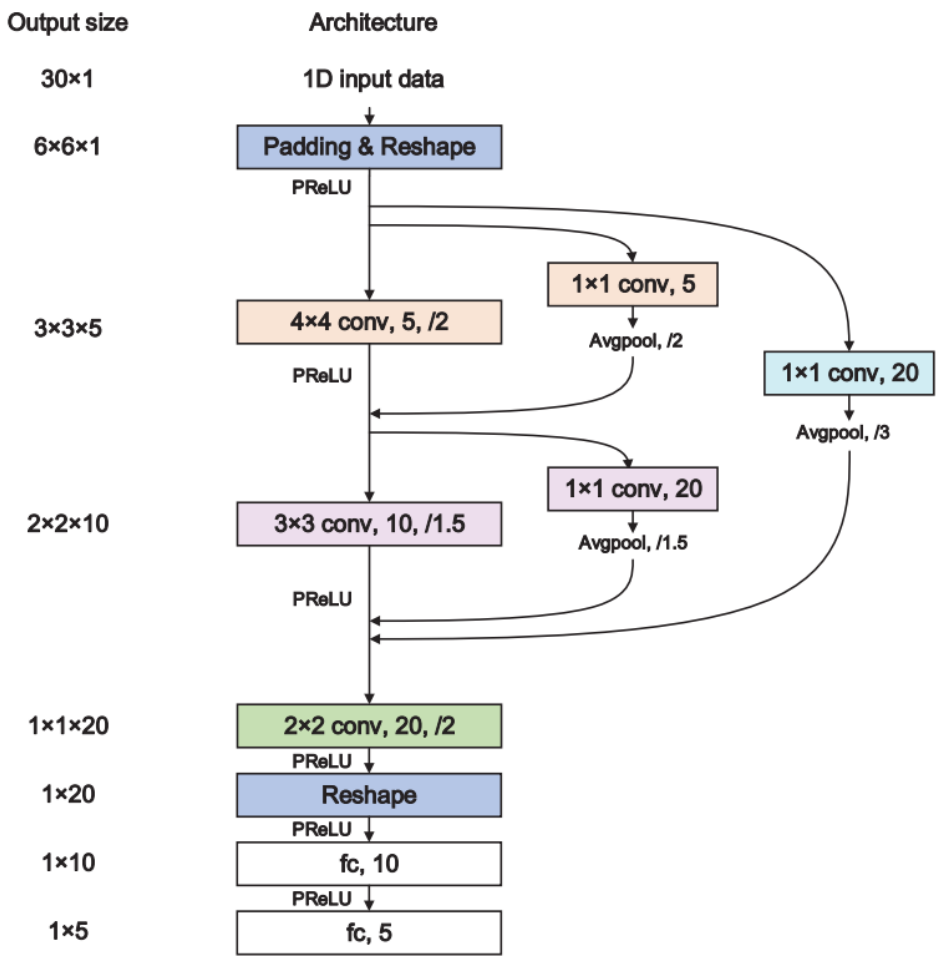


Figure 11. Proposed convolutional neural network architecture

method that increases the dimensions of features ruffle the spatial information and impede feature learning if the feature map is not carefully designed previously. The architecture of this model is depicted in Figure 11. Dense connections between convolutional layers are used along with 1x1 convolution kernels so that the features can be reused in different levels and the training of neural networks will be easier, where the original idea is from RestNet (He, 2016) and DenseNet (Huang, 2017). Moreover, the accuracy using this architecture is about 0.4% higher than that of a plain network according to our experiments, suggesting the advantages of a feature learning process that is just mentioned. The training and validating configurations are the same

as previous models. Besides, the batch size was 256, the initial learning rate was set as 0.0035, and the number of training epochs was 130. Cross entropy was selected as the loss function with Adam optimizer. PReLU (He, 2015) was used as the activation function instead of ReLU where the slope of function can be learned to fit the negative part of the scaled input data with a negligible increase in training cost. The final prediction accuracy for this model is 89.32% which is slightly better than that of MLP, suggesting the advantages of applying neural network methods and feasibility of further research based on deep learning models.

(2) Macro-level flight delay prediction based on long short-term memory network

Aviation delay is affected by multiple factors. Some of them are unpredictable such as military training activities, equipment failures at the airport, and extreme weather conditions. However, there are still valuable temporal contexts that can be retrieved for delay prediction. For example, the late arrival or departure of a previous flight will affect the on-time departure and arrival of succeeding flights. Such a pattern motivates us to model the aviation delay as a multivariate time series problem. Inspired by the considerable success of recurrent neural network (RNN) and its variants applied in sequential event prediction such as natural language processing, few researchers implemented RNN to capture temporal correlation of factors which may potentially influence aviation for more accurate delay prediction from the perspective of airports (Kim, 2016; McCarthy, 2019; Ayhan, 2018). However, a lot of valuable attributes in spatial and temporal domains are not or are rarely considered in these studies. Therefore, we presented a spatio-temporal data mining framework based on stacked LSTM networks for aviation delay prediction to bridge this gap. By combining spatio-temporal features from available datasets containing flight path information, airspace characteristics and weather, our model can learn representations of spatio-temporal sequences with multiple levels of abstraction

to predict subsequent aviation delay of an airport. To alleviate the overfitting problem, a regularization technique called Dropout (Zaremba, 2014) was applied in the middle of two adjacent stacked LSTM layers. Compared with previous work based on RNN, we made three main contributions in this project:

- Unlike other systems that predict delay before departure or make an implicit assumption that the journey of each aircraft does not vary significantly, we fully considered the indeterminacy of air transportation system to figure out the aviation delay prediction problem pointedly over a time horizon of one hour.
- We presented a complete workflow of data manipulation in this project. Some work provides comparative analysis for delay prediction using several models including RNNs, but they focus on more on non-time-series methods especially boosting and bagging methods for tree models, whereas the input format for RNNs is far away from the input format of other algorithms. Moreover, though the authors (Kim, 2016) described a data processing pipeline in detail, they created day-to-day sequences rather than flight-to-flight or minute-to-minute sequences like us, which does not make sense in real-time air traffic control.
- Multi-source data were integrated and extended to generate a spatio-temporal dataset with richer information in both spatial and temporal domains.

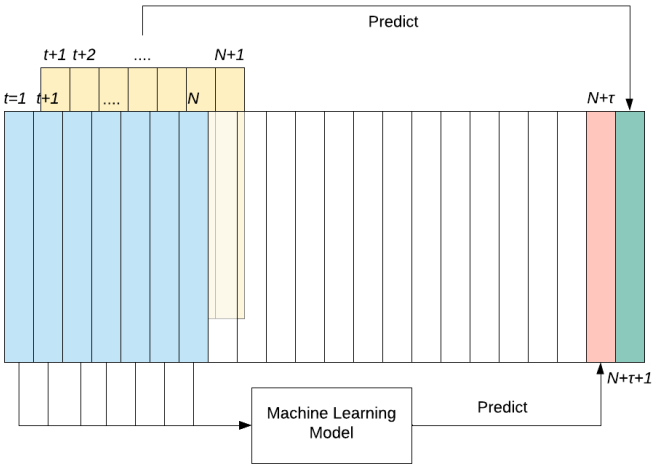


Figure 12. Transform time series data into a supervised learning problem.

The overview of our problem setting is shown in Figure 12. We used a sequence starting at $t = 1$, the information in a certain period ($t \in [1, N]$ and $t \in \mathbb{Z}$) would be used to predict the delay at the time stamp $N + t$. Figure 13 shows the basic structure of a LSTM cell. The gate mechanism is a key component in the LSTM structure. Gates are tracked to optionally let information (hidden state h_t and cell state C_t) through. There are three gates in a LSTM cell – forget gate, input gate, and output gate, and their functions are represented by the following equations:

$$f_t = \sigma(W_t \cdot [h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_t \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * tanh(C_t)$$

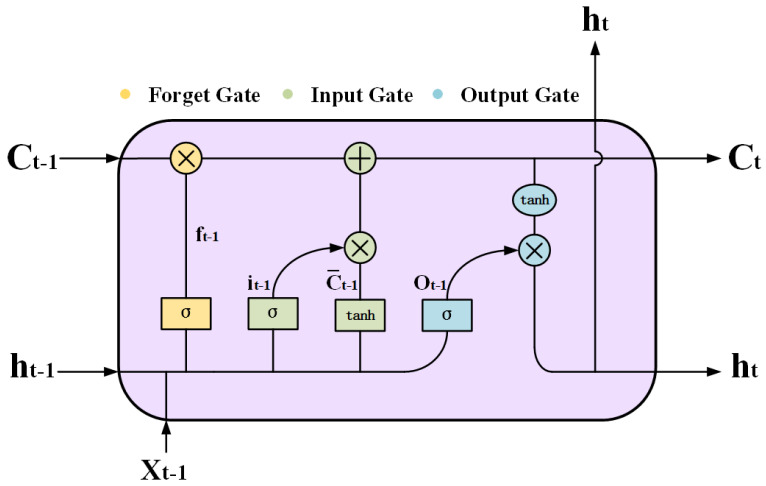


Figure 13. A LSTM cell.

where σ is the sigmoid activation function which observes sates at the previous step and outputs a number between zero and one to control the degree of remaining information flow. i_t is the input gate that is combined with the candidate value, $\tilde{C}_t$ after tanh layer to update the state, then the old cell state C_t representing long-term memory can be replaced by the new one as shown in the 4th equation. h_t is the final output calculated by the multiplication of o_t and $tanh(C_t)$ which could be the input for next LSTM cell.

For macro-level flight delay prediction component, we used data of the first day of each month for the period of July 2016 through February 2017 in Coordinated Universal Time (UTC). Note that, to guarantee the spatio-temporal continuity of data, only records whose *Arrival Airport* is Hartsfield-Jackson Atlanta International Airport (ATL) were selected. The outline descriptions of features extracted from each domain are illustrated in Figure 14.

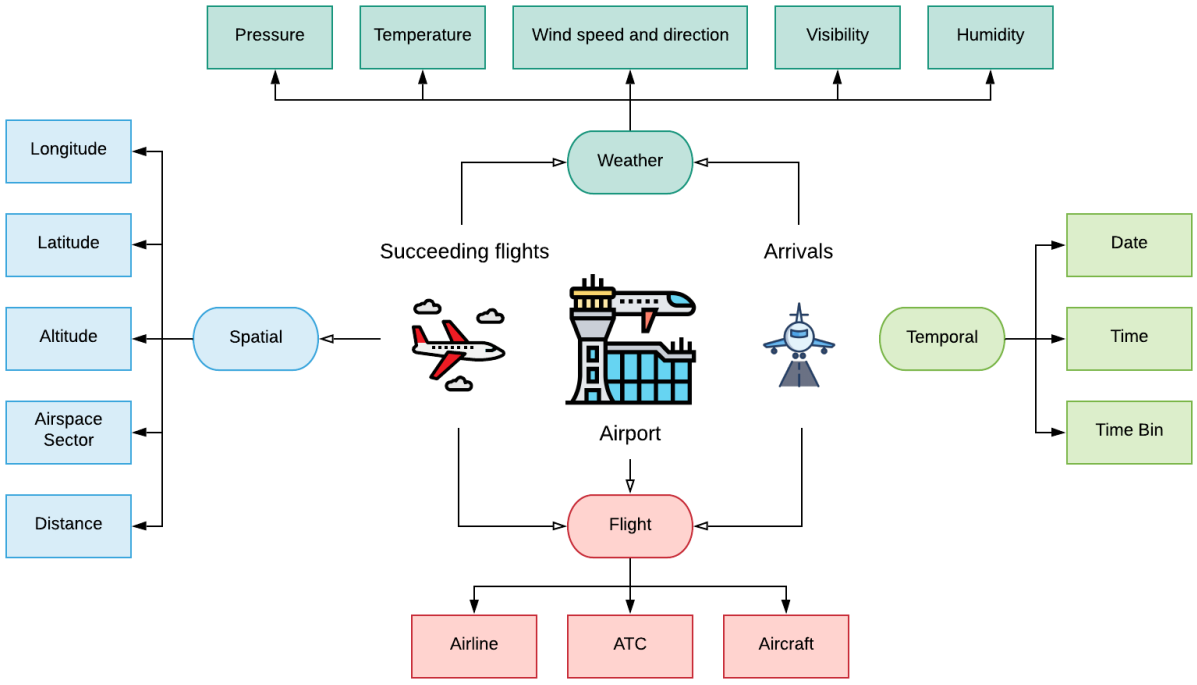


Figure 14. Overview of features. Once a flight arrives, it triggers the prediction mechanism based on airport information and the corresponding succeeding flight information. Various features in each domain can be combined concerning spatio-temporal characteristics. For example, combining weather and spatial features of succeeding flights produces a series of indicators that reflect the weather conditions of each aircraft's geographic locations. Then, if we consider temporal elements further, the synthetic features describe variations of flights over space and time synchronously.

We adopted stacked LSTM architecture for our aviation delay prediction as shown in Figure 15, since RNNs can benefit from depth in space by stacking multiple recurrent hidden layers on top of each other (Graves, 2013). There is a sequence with length N, and each output of the second LSTM layer is jointed with a fully connected (FC) layer. We expected the model to look backwards at historical states and attempt to capture potential temporal characteristics since

some important hidden representations may be lost in the last LSTM cell. Besides, FC layer with more neurons has better expressivity for a complex function. In the process of our experiments, we found that the loss converges too quickly and tends to zero which is proven to be the overfitting problem; thus, we add a Dropout regularization between two LSTM layers.

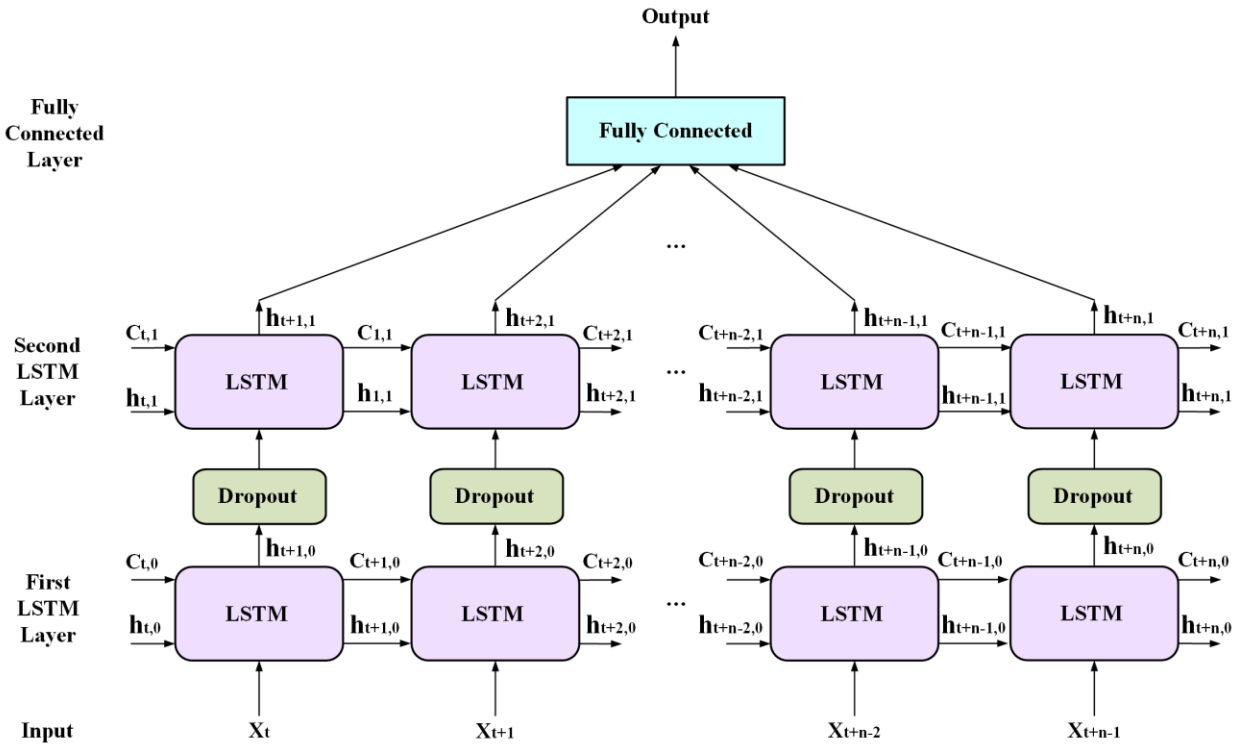


Figure 15. Stacked LSTM architecture with Dropout.

The inputs for LSTM are three-dimensional sequential data. Figure 16 shows the progress of constructing multivariate time series. It should be noted that our data only contains records of the first day of each month from July 2016 to February 2017 in UTC, so we had to split the original data into eight days by time matching firstly and then slice them separately to create a series of time blocks with a length of N. Next, the training set and test set were generated by

stacking these arrays in sequences vertically, which keeps the time continuity within a time block. The order of inputs will affect the training results since the former samples will get larger gradients in general, while the neighboring sequences in the time series have a similar distribution. Hence, we shuffled the order in which sequences are fed to the LSTM to guarantee the generation of the trained model. Note that we did not shuffle the ordering of elements within individual sequences.

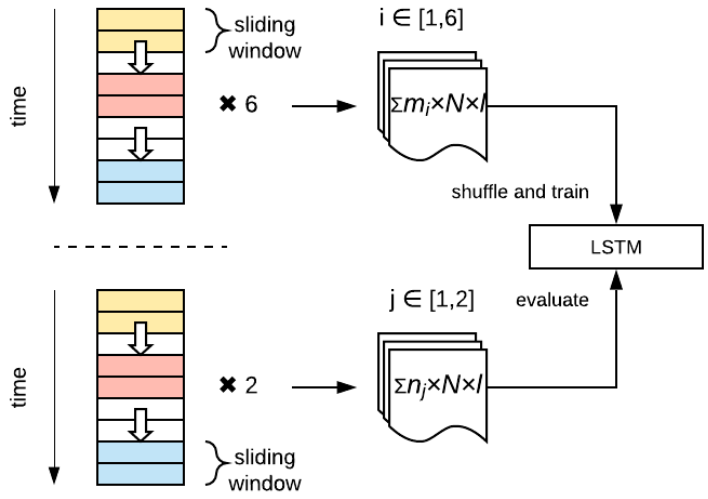


Figure 16. Array slicing for constructing time series. m and n denote the number of instances of daily data after slicing for training and testing, respectively. N represents the length of sliding window over time, and l is the number of features.

Two types of datasets were used for training and evaluation. The first dataset with 66 features contains only flight and weather information in the airport, while the other dataset with 203 features contains not only the first dataset but also spatio-temporal information of succeeding flights. We let the sequence length N be 30, 60, 90, and 120, respectively to generate datasets with

different time steps. Mean Square Error (MSE, in *minute*^2) was chosen as the loss function in the training. Our experiment was divided into three steps:

- Compare the performance of our model with other powerful ensemble algorithms which are utilized by most research such as Random Forest Regression (RF), a bagging method; and Gradient Boosting Regression Tree (GBRT), a boosting method. RF and GBRT are not designed for sequential prediction, so we just feed all data into the model instead of slicing data according to *N* like LSTM. Besides, other commonly used baseline models such as Linear Regression (LR), Support Vector Regression (SVR), Regression Tree (RT) and MLP were also used for comparison.
- The second step is to use two datasets separately to train our stacked LSTM model based on different sequence length, *N*. Therefore, we can analyze the impact of sequence length and en-route spatio-temporal information for our model.
- We applied the model for different airports to verify the validity. The airports include large hub airports such as Los Angeles International Airport (LAX), O'Hare International Airport (ORD); a mid-size airport, Orlando International Airport (MCO); and a small airport, Daytona Beach International Airport (DAB).

Table 5 compares performance of different model settings based on *T* data. *T* represents the dataset without information on subsequent flights, on the contrary, *ST* represents the entire dataset. The best score is highlighted. In a quick glance, it seems clear that our LSTM model outperforms other models. The second-best schema is the ensemble method, in which GBRT works best. These conclusions can also be drawn from Table 6. LR perform poorly on both *T* and *ST* that illustrates there is no apparent linear relation between features and target. Generally

speaking, the performance of models with richer spatio-temporal features is better than models that take only temporal correlations into account. Besides, we observed that there is a trend of decline for MSE for test set as the length of the sliding window increases. We believe that the reason for this phenomenon is that the LSTM learns more hidden patterns of prior knowledge from longer sequences. Although longer sequences may introduce noise to the model during training, the gate mechanism prevents the abuse of long-term dependencies.

Figure 17 depicts the gap between the predicted value and the ground truth over time in a more intuitive way. It can be clearly seen that there is a very slight difference in amplitude, and their trends are consistent basically which also verifies that our proposed model is able to predict aviation delay accurately.

Method	**Model**	**Test MSE**
Linear	LR	218.9779
Non-linear	RT	111.2693
	SVR	150.2401
Ensemble	RF	84.3054
	GBRT	68.9447
Neural Network	MLP	119.0170
	LSTM-30	64.1153
	LSTM-60	62.3336
	LSTM-90	60.3336
	LSTM-120	**52.7232**

Table 5. Comparison of prediction accuracy using *T* data and different sequence length.

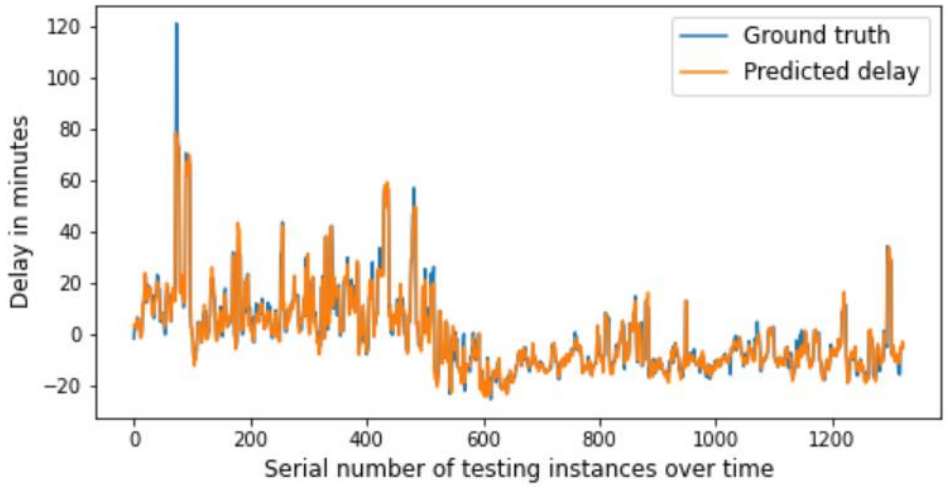


Figure 17. Predicted delay vs. ground truth for *ST* test set (LSTM-120).

Table 7 shows the MSE values of test sets for different airports. The distributions of data for different airports vary dramatically no matter from spatio-temporal perspective (e.g. weather varies in different cities) or from flight's perspective (e.g. an airline has different flight arrangements for different airports). Therefore, we used the data at various airports to re-generate time series with spatio-temporal features and retrained the LSTM-120 model. It demonstrates the effectiveness of our proposed LSTM model and the corresponding feature engineering for large hub airports. However, this model does not make sense for mid- and small-size airports due to the relatively few flights, which are hard to generate representative sequences.

Method	Model	Test MSE
Linear	LR	202.3007
Non-linear	RT	77.2012
	SVR	57.7634
Ensemble	RF	52.0770
	GBRT	43.8420
Neural Network	MLP	56.8772
	LSTM-30	43.9821
	LSTM-60	39.5061
	LSTM-90	35.7900
	LSTM-120	**25.6320**

Table 6. Comparison of prediction accuracy using *ST* data and different sequence length.

Airport	Test MSE
ATL	25.6320
LAX	27.3342
ORD	46.7891
MCO	73.5672
DAB	110.6825

Table 7. Comparison of prediction accuracy of LSTM-120 using *ST* data at different airports.

Large-Scale Flight Dispatch for Disaster Evacuation: A Learning and Planning Approach

The effectiveness of resource allocation under emergencies, especially hurricane disasters, is crucial. However, most researchers focused on emergency resource allocation in ground transportation systems. In this project, we present a novel flight dispatch algorithm in large-scale evacuation scenarios, namely Learning-to-Dispatch (L2D), a reinforcement learning

(RL) based air route dispatching system, that aims to add additional flights for hurricane evacuation while minimizing the airspace's complexity and air traffic controller's workload. Given a bipartite graph with weights that are learned from the historical flight data using RL in consideration of short- and long-term gains, we formulated the flight dispatch as an online maximum weight matching problem. Different from the conventional order dispatch problem, there is no actual or estimated index that can evaluate how the additional evacuation flights influence the air traffic complexity. Then we proposed a multivariate reward function in the learning phase and compare it with other univariate reward designs to show its superior performance. The experiments using the real-world dataset for Hurricane Irma demonstrated the efficacy and efficiency of our proposed schema.

Overall Structure of L2D framework

L2D system consists of three modules: offline simulation, offline learning, and online matching, as illustrated in Figure 18 below. Although it is computationally expensive for learning the value functions, it could be completed before disasters approach. Thus, offline process is not time-sensitive in practice.

Finite Markov Decision Process Modeling

The Markov Decision Process (MDP) is typically used to model sequential decision-making problems (Xu, 2018). In an MDP, an agent behaves in an environment regarding a policy

that specifies how the agent selects actions at each state of the MDP. The agent's goal is to maximize its gain $G_t = \sum_{i=t}^{T} R_{t+1}$, the expected return starting from time t. A common objective to solve and MDP is to learn the value functions, e.g., the state-value function $V_\pi(s) = \mathbb{E}[G_t \mid s_t = s]$ and the action-value function $A_\pi(s) = \mathbb{E}[G_t \mid s_t = s, a_t = a]$. In

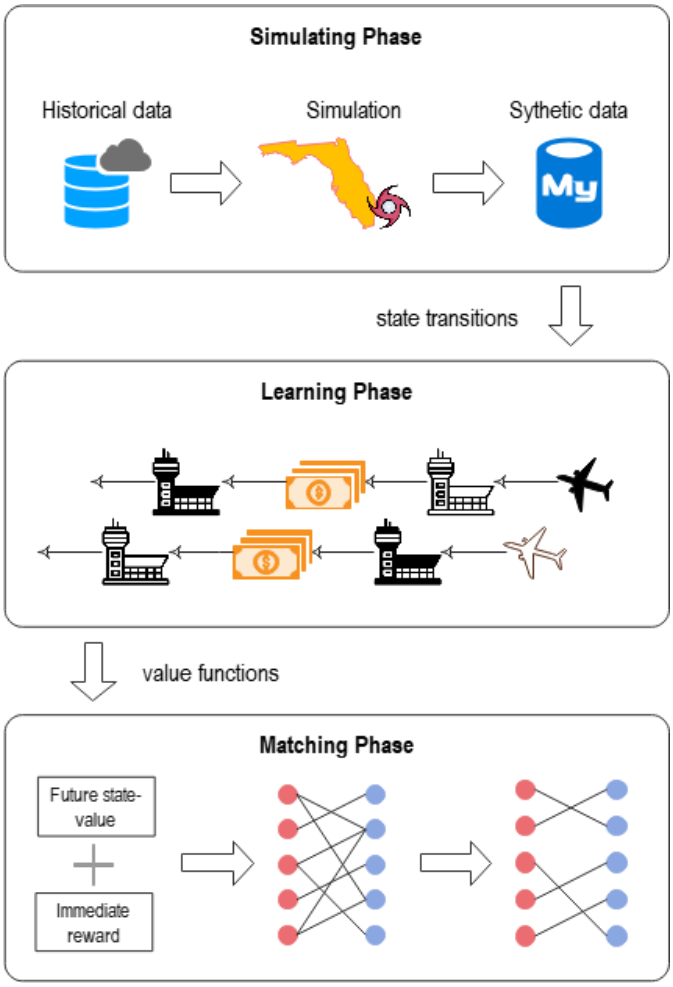


Figure 18. Overview of L2D system.

this project, we built the MDP in a local view in terms of the flight dispatch problem: each independent aircraft was modeled as an agent. And we regarded each agent as the same so that

each agent performs the same policy. We defined components in an MDP defined by $M = \langle \mu_0, S, P, A, R, \gamma \rangle$ as follows and illustrated them in Figure 19.

- **Initial States** μ_0 is the set of initial states of all the aircraft.
- **State** S is determined by a spatio-temporal vector $\langle g, t \rangle$, where g is the airport index where the aircraft is located and t is the time index.
- **Action** A is defined as a set $\{0,1\}$, where 0 and 1 represent that the aircraft is assigned to a flight or not in a time slot, respectively.
- **State Transition** $P(s'|s,a)$ specifies the probability of the state transitioning to s' upon taking action a in state s, which depends on the estimated future information such as flight delay and air flow. We use the predictive models based on machine learning that are mentioned in the "Air Mobility Prediction" Section to acquire such estimates.
- **Reward** R is the immediate gain that an aircraft can get if it performs an action to accept an evacuation or not from current state s with the next state being s'. It is defined as an index with hybrid factors called ATCI that we think can reflect the influence of an evacuation flight to the corresponding airports.

$$\text{ATCI}=\frac{100}{\lambda \times w + \mu \times et + v \times d}$$

where w is an indicator for air traffic controller's workload. et denotes the elapsed time of a flight, d represents the distance while λ, μ, v are hyperparameters.

- **Discount Rate** γ determines how important future rewards to the current state. We set $\gamma = 0.9$ in our application. For a flight which lasts for T time steps with ATCI R. The final reward is given by $R_\gamma = \sum_{t=0}^{T-1} \gamma^t \frac{R}{T}$.

To make the process of sequential decision-making clearer, it was assumed that an aircraft in airport **A** receives a flight plan from **A** to evacuation zone **B**, and then goes to destination **C** at time 10:00. The trip is composed of two airways that are estimated to be completed in 1 hour and 2 hours, respectively. And the cost for each airport is 3, 6, and 3 (unit is omitted here), respectively. We segment time slots into 1-hour windows (starts at 10:00) and this plan will make the aircraft transit from current state s to final state s' with a reward r, where $s = (A, 0)$ and $s' = (C, 3)$ (1 for support time, 2 for delivery time), and $r = 4 + 4 * 0.9 + 4 * 0.9^2 = 10.84$.

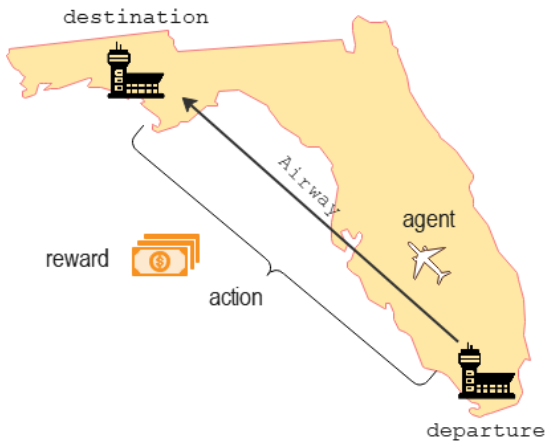


Figure 19. MDP formulation in the L2D. An agent represents an aircraft/pilot. The action in flight dispatch represents assigning the aircraft/pilot to server a particular evacuation flight.

Offline Learning

(1) Generate Simulated Evacuation Flight Transactions

The flight transactions in the historical database record only a series of one-way itineraries such as from Miami to New York City. However, in the context of disaster evacuation, an evacuation airway could be a multi-way itinerary. Therefore, we generated simulated evacuation flight transactions based on predicted air mobility indicators from machine learning models that are trained using historical real-world data. These transactions will be utilized to learn the value functions later.

Algorithm 1 Evacuation Flight Transaction Productor

Input: Historical flight data **F** and human-factor data **H**; predefined number of transactions **N**. The airports k and time t are extracted randomly from the historical datasets in each iteration.

Output: Synthetic airway set $\{\langle I_o, S_D, A_o, C_t^k, S_A\rangle\}$

1: **for** $i = 0$ to $i =$ **N do**
2: Generate a potential airway tuple $\langle I_o, S_D, A_o, S_A\rangle_i$ but without the ATCI.
3: $atc_t^k \leftarrow$ number of air traffic controllers $\times$ the corresponding training time
4: $air_t^k \leftarrow$ the average of delay $\times$ the number of flights
5: $et_t^k \leftarrow$ sample from the historical elapsed time distribution complying with Gaussian
6: $d_t^k \leftarrow$ flight length
7: $C_t^k = 100/[\lambda \times (air_t^k/atc_t^k) + \mu \times et_t^k + \nu \times d_t^k]$
8: Take the C_t^k as the ATCI in the airway tuple to construct $\langle I_o, S_D, A_o, C_t^k, S_A\rangle_i$.
9: **end for**

Algorithm 1 describes the progress of producing synthetic transactions. In each episode, we randomly selected departure and arrival states including airport and time from the historical AOTP flight dataset and treated the availability index as either 0 (not available) or 1 (available).

At a time t, workload $w_t^k = air_t^k/atc_t^k$, elapsed time et_t^k, and distance d_t^k could measure a one-way flight as well as a multi-way flight. Let's use an example to illustrate the

process. There is a transaction from k_1, supporting k_2 to transport people to k_3, thus w_t^k should be the accumulated workload when an aircraft takes off or lands in these three airports at the corresponding time step, which is calculated as $w_t^k = w_{t1}^{k1} + w_{t2}^{k2} + w_{t3}^{k3}$, where t_1 represent the time of departure and t_2, t_3 represent the estimated time of arrival.

Algorithm 2 Hyperparameter Selection

Input: A pre-defined hyperparameter candidate set $\{\mathcal{H}_i\}$ where i is the index of each candidate, $i \in [0, N_{hyper}]$.
Output: An optimal hyperparameter tuple $\langle \lambda^*, \mu^*, \nu^* \rangle$.
Initialize ρ_{hyper} and $\mathcal{H}^* = \langle \lambda^*, \mu^*, \nu^* \rangle$ of all zeros.
for $i = 0$ to $i = N_{hyper}$ **do**
  $\rho_i^w, \rho_i^d, \rho_i^t \leftarrow$ calculate the Spearman rank-order correlation coefficient of ATCI and workload, flight length, elapsed time in $\mathcal{H}_i$, respectively.
  if $\rho_i \leq 0.4$ for each coefficient **then**
    $\rho_i^{neg} = |\rho_i^w + \rho_i^d + \rho_i^t|$
    if $\rho_i^{neg} > \rho_{hyper}$ **then**
      $\rho_{hyper} \leftarrow \rho_i^{neg}$, $\mathcal{H}^* \leftarrow \mathcal{H}_i$
    end if
  end if
end for

Algorithm 2 is the hyperparameter selection strategy for computing ACTI. The intuitive behind the hyperparameter selection algorithm is to let every variable be negatively correlated with the ACTI as mu as possible so that the reward could not be heavily biased toward either factor.

(2) Policy Evaluation

Based on the definitions of MDP and ATCI, we broke down historical data into a set of evacuation flight transactions (s, a, s', r) including evacuation actions and standby actions. Note that we did not distinguish individual aircraft and regard them as the same agent, thus the simulated transactions for all aircraft can form a single dataset for computing value functions.

We can also call this process Policy Evaluation. In practice, we assumed that the policy remains unchanged over the learning period, so we omit the subscript π.

We then defined the specific Temporal-Difference (TD) update rule for each action in the following and the schematic diagram is illustrated in Figure 20. In a standby transaction, the agent cannot get the immediate reward, so the TD update rule is $V(s) \leftarrow V(s) + \alpha[0 + \gamma V(s') - V(s)]$, where s is the current state of the driver and s' is the next state afterwards. On the other hand, in an evacuation transaction, the agent receives immediate reward and triggers a state transition. The update rule is $V(s) \leftarrow V(s) + \alpha\left[R_\gamma + \gamma^{\Delta t} V(s'') - V(s)\right]$, where s'' is the estimated state when an aircraft completes an evacuation flight (that is, it arrives at the destination airport), and it can be represented by $s'' = (t + \Delta t, g_{dest})$. To be specific, Δt denotes the cumulation of estimated time spans of supporting and evacuating process. The reward should be discounted and calculated using the aforementioned discount rate.

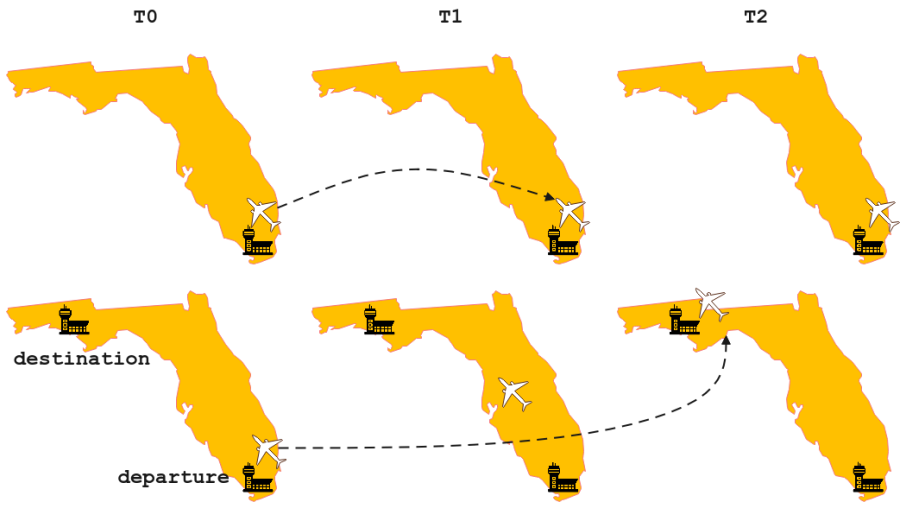


Figure 20. Arbitrary-step TD update rule. An agent may perform two kinds of actions: standby (the top figure) and deliver.

Algorithm 3 State-value Lookup Generator

Input: Collect state transitions $\{\langle s_i, a_i, r_i, s_i' \rangle\}$ from synthetic evacuation transactions . State is represented as a spatio-temporal tuple: $s_i = \langle g_i, t_i \rangle$, $s_i' = \langle g_i', t_i' \rangle$. T is the time of last state in the finite-horizon MDP.

Output: Value function $V(s)$ for all states.

Initialize $V(s)$ and $N(s)$ of all zeros

for $t = T - 1$ to $t = 0$ **do**
  $\{\langle s_i, a_i, r_i, s_i' \rangle\}^t \leftarrow$ A subset of state transitions in which $t_i = t$
  for each instance in the subset **do**
    $N(s_i) \leftarrow N(s_i) + 1$
    TD error: $\delta_i \leftarrow \gamma^{\Delta t(a_i)} V(s_i') + R(a_i) - V(s_i)$
    $V(s_i) \leftarrow V(s_i) + \frac{1}{N(s_i)} \delta_i$
  end for
end for

In practice, we assume a perfect model of environment. In other words, the state transition probability of each state-action pair can be accurately modeled in advance. Therefore, we can perform an implementation based on dynamic programming to compute the value functions, where an episode records the transactions of an aircraft in a day. The DP algorithm is conducted in an inverted order of time slots, as described in Algorithm 3. The resultant state-value function captures spatio-temporal patterns of the air route network dynamics. To make it clearer, as a special case, when using no discount and an episode-length of a day, the state-value function in fact corresponds to the expected influence that this aircraft will make on average to the whole network from the current time until the end of the day.

Online Planning

(1) Advantage Function Trick

Recall the problem formulation section, the goal of the online dispatch is to determine the best matching between aircraft and evacuation flights in each time slot. W(i,j) denotes the value of taking an action in a state that corresponds to the concept of the action-value function $Q(s,a)$. In practice, we use the advantage trick (Xu et al., 2018) to reduce computational complexity. We explain this process in detail in the following paragraphs.

According to the definition of value functions including state-value function $V_\pi(s)$ and action-state function (Sutton & Barto, 2018):

$$\begin{aligned} V_\pi(s) &= \mathbb{E}[G_t|S_t = s] \\ &= \mathbb{E}[R_{t+1} + \gamma G_{t+1}|S_t = s] \\ &= \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)[r + \gamma V_\pi(s')] \end{aligned}$$

$$\begin{aligned} Q_\pi(s) &= \mathbb{E}[R_{t+1} + \gamma V_\pi(S_{t+1})|S_t = s, A_t = a] \\ &= \sum_{s',r} p(s',r|s,a)[r + \gamma V_\pi(s')] \end{aligned}$$

The advantage function is defined by:

$$A_\pi(s) = Q_\pi(s) - V_\pi(s)$$

In fact, the state transition is deterministic, and the online policy remains unchanged, so we can omit the policy subscript and get $V(s) = \sum_{s',r} p(s',r|s,a)[r + \gamma V(s')]$, and $Q(s) = r + \gamma V(s')$. Note that we adopt an arbitrary-step TD update rule instead of one-step like the

traditional definition of value functions. Then, we get $V(s) = \sum_{s',r} p(s', r|s, a)[r + \gamma^{\Delta t} V(s')]$, and $Q(s) = r + \gamma^{\Delta t} V(s')$. Now we rewrite the advantage function as follows:

$$A(s) = r + \gamma^{\Delta t} V(s') - V(s)$$

After introducing the advantage function trick, the objective function of flight dispatch in one timestamp becomes:

$$\underset{a_{ij}}{argmax} \sum_{i=0}^{m} \sum_{j=0}^{n} A(i,j) a_{ij}$$

$$s.t. \sum_{i=0}^{m} a_{ij} \leq 1, j = 1,2,3 \dots, n$$

$$\sum_{j=0}^{n} a_{ij} \leq 1, i = 1,2,3 \dots, m$$

$$A(i,j) = \gamma^{\Delta t_j} V(s'_{ij}) - V(s_i) + R_\gamma(j)$$

For an edge between an aircraft i and a transaction j, the advantage function is computed as the expected gain of serving the transaction (receiving a discounted reward and ending at a state determined by the transaction's destination) minus the expected value of remaining in the same state (value function of the aircraft's current state). Figure 21 illustrates this process.

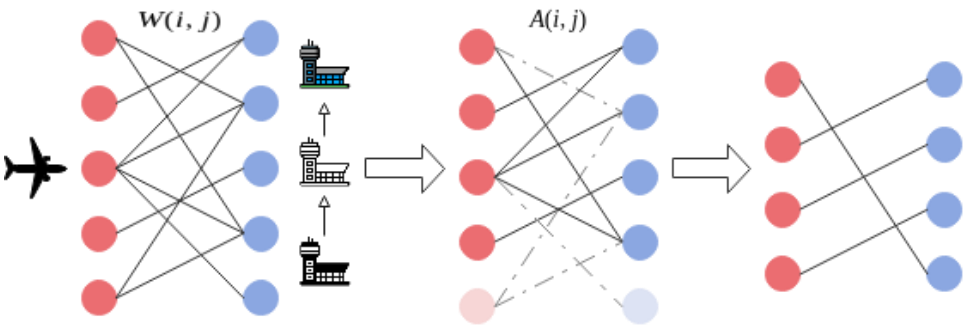


Figure 21. Arbitrary-step TD update rule. An agent may perform two kinds of actions: standby (the top figure) and deliver.

(2) Online Transaction Evaluation

Different from offline transaction generation, the ground truth or samples from the normal distribution based on historical data of air mobility indicators are not known in the process of online transaction generation. As a result, we had to use predictive models we have mentioned in the last section, *Air Mobility Prediction*. We can employ the trained machine learning models to acquire the estimated air mobility indicators such as workload, elapsed time and distance, and calculate the corresponding ATCI to construct the online flight candidates using the way described in the Algorithm 1.

Furthermore, considering that evacuation flights may happen to be delayed, and each delay will cause a sequence of delays which influences future dispatch. Thus, a dependable flight delay predictor is required in the generation process of online transactions for computational reduction. To achieve it, we divided the historical flight delay information into three categories regarding the delay in minutes: less than 0, greater than 0 but less than 15, greater than 15 minutes, and we train a *Random Forest* classifier to predict the delay categories. The attributes we used are the historical information of aircraft departure and arrival location, aircraft departure time, and departure airport weather information including wind speed, wind direction, temperature, precipitation, and visibility. Based on the different delay categories, the transaction value will be multiplied by 1.1, 1, and 0, respectively. In other words, this delay prediction model encourages the online planning system to dispatch a flight that is estimated as “not delayed” since the transaction which will be delayed more than 15 minutes will be “removed”.

After introducing the "filter" on online transactions, the resultant bipartite graph can be built and the objective of flight dispatch in a timestamp (the interval is 1 hour by default) can be achieved in the following steps:

- Load the state-value lookup table and delay classifier trained offline.
- Collect available aircraft and online flight transactions (airways) (the output of classifier is “not delayed”).
- Compute A_π for each valid aircraft-airway pair using $A_\pi(s_{ij}, a_{ij}) = Q_\pi(s_{ij}, a_{ij}) - V_\pi(s_{ij}, a_{ij})$.
- Build a completely weighted bipartite graph where every possible edge between aircraft and airways exists.
- Solve the following equation by Hungarian algorithm.

$$\arg\max_{a_{ij}} \sum_{i=0}^{m}\sum_{j=0}^{n} W(i,j)a_{ij}$$
$$s.t. \sum_{i=0}^{m} a_{ij} = 1, j = 1,2,3\ldots,n$$
$$\sum_{j=0}^{n} a_{ij} = 1, i = 1,2,3\ldots,m$$

Note that,

$$W(i,j) = A_\pi(i,j)$$
$$= \gamma^{\Delta t_j} V(s'_{ij}) + R(s_i, a_i, s'_{ij}) - V(s_i)$$

- Forward the machine to the relevant aircraft and dispatch the corresponding flights. Remaining aircraft will go to the next round for matching new generated transactions.

(3) Online Transaction Allocation Strategy

Assume an airport is the nearest airport to the supporting airport and not as busy as other evacuation airports. Therefore, the transaction whose departure region is that airport will always have a higher value. In other words, the dispatch system will match more aircraft and flights which take off from that airport if we do not specify configuration in the process of online transaction generation. To alleviate this issue, we established a set of balance weights $\{a, b, c, d\}$ such that:

$$N_1 : N_2 : N_3 : N_4 = a : b : c : d$$

where N_1, N_2, N_3, and N_4 represent the number of evacuation transactions in which the departure airport is C_1, C_2, C_3, and C_4, respectively. We follow two special strategies to set up the control variables: central allocation and uniform allocation that are described as follows.

- **Central Allocation** is a strategy that satisfies different evacuation demands for evacuees in different areas.
- **Uniform Allocation** is a strategy that arranges flights roughly equally to each evacuation airport. Such a strategy is used to disperse flight capacity pressure uniformly.

Experimental Setup

We assume that residents in evacuation zones (Florida in our study) would like to evacuate by flight six hours before the arrival of the hurricane T_h. We used historical data of Hurricane Irma to evaluate our system. Specifically, the eye of Irma was close to Southern Florida in the morning on September 10, 2017 (National Weather Service, 2017). To coordinate

evacuation flights in a safer manner, we let the evacuation flight dispatching start at 12 AM and end at 6 AM.

Four experimental scenarios were designed in our study:

Scenarios 1: Only MIA is set as the evacuation airport while ALT, BNA, CLT, DFW and JFK are set as supporting or destination airports. The time step (dispatch interval) in our system is one hour. $M = 200$ evacuation flights are generated at each timestamp, and the evacuation plan must be completed before T_h with a fixed number of evacuation-oriented aircraft $N = \{30,60,120\}$, which are evenly assigned to the relevant airports. We assumed that aircraft could transport people immediately without waiting for overhaul, and passenger boarding and landing.

Scenarios 2: Only MIA is set as the evacuation airport while the ATL, BNA, CLT, DCA, DFW, DTW, and JFK are set as supporting and destination airports. The time step is one hour. $M = 500$ with fixed number of aircraft candidates $N = \{3,5,7\}$ which are arranged at each airport at the first three timestamps. In the remaining time slots, the aircraft dispatched previously will continue to be responsible for the following evacuation tasks until T_h. We also assumed that aircraft could transport people immediately without waiting for overhaul, passenger boarding, and landing.

Scenarios 3: MIA, FLL, TPA, MCO are set as the evacuation airports. Other settings are the same as described in Scenario 4.

Scenarios 4: We kept the same configuration as the third scenario, whereas we let ATL airport be closed due to an emergency starting at the third hour of the evacuation (3 AM). Such a setting is designed to evaluate the robustness of our system.

Evaluation Objectives: Our proposed ATC impact factor is the multiplicative inverse for workload, flight length and elapsed time, thus the larger it is, the lower incremental air traffic complexity is. We also consider two more intuitive evaluation metrics: "*How many evacuation flights can be completed within six hours?*" and "*How many evacuation flights each airport has to coordinate with?*" The fourth scenario is adopted to answer, "*Could the system adjust original flight plan in time if an emergency occurs at the corresponding supporting airport?*". We also evaluate the performance using different allocation strategies in terms of the number of dispatched flights taking off from each evacuation airport.

Results Analysis and Discussions

Detailed results of experiments are illustrated in this section, which provide a more complete understanding of the efficacy and efficiency of our proposed framework. We will discuss the results of different experimental setups one by one in the following paragraphs.

(1) Scenario 1 Results

Figure 22 shows comparison results of the four reward function settings regarding the number of completed evacuation flights. Note that the transit flights via MIA are not counted and there are a total of five departure flights. It is not surprising that the time-based and distance-based methods tend to dispatch evacuation flights to ATL and CLT because they are the closest airports to MIA compared to the other four airports, while the elapsed time is usually positively

correlated with flight length. On the contrary, very few evacuation flights will take off or land from ATL and CLT using the workload-based method. Such a result reconciles with the real conditions since they are two of the world's busiest airports as well as the nearest airports to MIA, and the evacuation flights may increase airspace complexity in a short time. Besides, airports in the experimental setting are almost large hub/core airports in the US. Therefore, it is easy to explain that why the workload-based method produces far fewer airway-aircraft pairs than others, the same result is also shown in Figure 22. It is worth pointing out that a more balanced flight allocation plan is obtained with our designed reward function. The extra workload caused by evacuation is nearly uniformly shared by all airports.

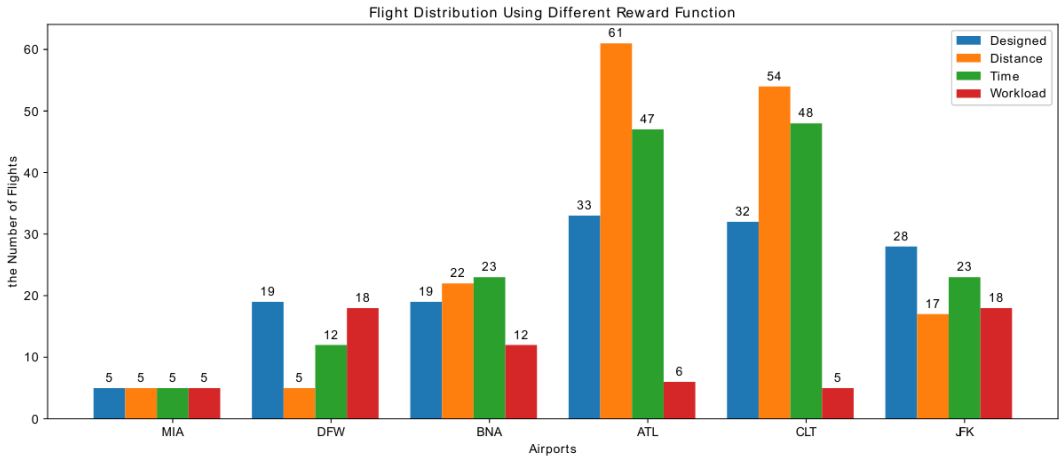


Figure 22. Flight distribution with different reward functions (N = 30). The transit flights via MIA are not counted.

The comparison results for different airway-aircraft ratios are shown in Figure 23. Here the "200" in the airway-aircraft ratio indicates that 200 potential airways are generated at each time step, then they can be paired with available aircraft. As the ratio increases, there is a convergent phenomenon in the evaluation metrics depending on average elapsed time and flight

length. In other words, all strategies tend to achieve similar performance because of sufficient aircraft supply. Otherwise, when the supply is not enough such as $N = 30$ or $N = 60$, the workload-based method results in more flight time and flight distance for each evacuation flight compared to the other three methods.

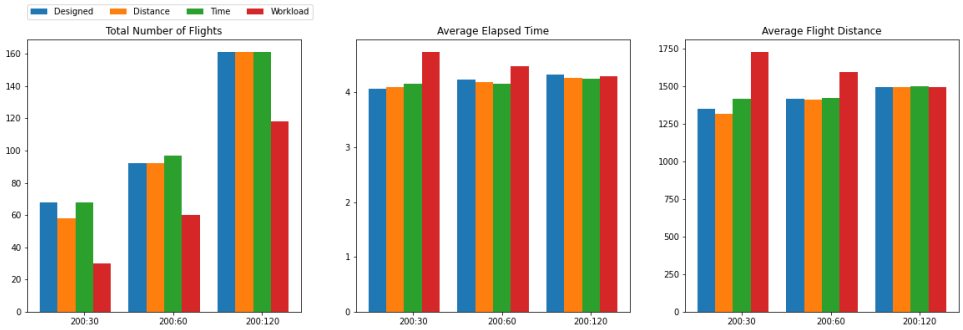


Figure 23. Comparison of distance-based method, time-based method, workload-based method and the ATCI-based method in three metrics on the Hurricane Irma example. X-axis stands for the ratio of potential airway-aircraft at each time step.

Assume that an evacuation aircraft can transport 400 people at a time, approximately 28,000 people can evacuate towards other cities within 6 hours even if only 30 aircraft serve for evacuation. Therefore, the ATCI-based method can produce not only good but also unbiased evacuation flight plans.

(2) Scenario 2~3 Results

Figure 24 shows the result of the second experimental setting while we used $N = \{3,5,7\}$. The boxplot presents the number of aircraft taking off from each airport every hour. Generally speaking, either the multi-airport or the single-airport tends to dispatch the flights to the nearest

and less busy airport. This dispatch method based on the transaction value can reduce the additional pressure on the airspaces or airports caused by the evacuation flights. Meanwhile, regardless of a multi-airport evacuation or a single-airport evacuation, it is not astonishing that the number of aircraft taking off at the supporting airport per hour is much less than that of the evacuation airport. The most important reason is that all flights need to go to the evacuation airport to pick evacuees up. In the multi-airport setting, the dispatch system tends to dispatch evacuation flights from TPA since TPA is the closest evacuation airport to the supporting airport compared to MIA and FLL, and TPA is not as busy as MCO, which means the evacuation transaction in which the arrival region is TPA has the higher transaction value. However, in the single evacuation airport setting, only MIA is used as an evacuation airport, so a large number of aircraft is concentrated in MIA. In other words, when more airports participate as evacuation airports, the negative impact of additional flights on the evacuation airport capacity can be significantly reduced. Especially, when the N is 7, if only a single airport participates as an evacuation airport, this airport should be able to handle 30 aircraft per hour, and the specific number of aircraft which take off at different airports at each time stamp is shown in Table 8. If MIA acts as the only one evacuation airport, it has to handle approximately 50 airplanes in the last two hours. To put it in another way, the takeoff interval of adjacent flight needs to be controlled within approximately one minute. Considering the delay and error of the estimated arrival time of the flight, this dispatch plan is unreasonable.

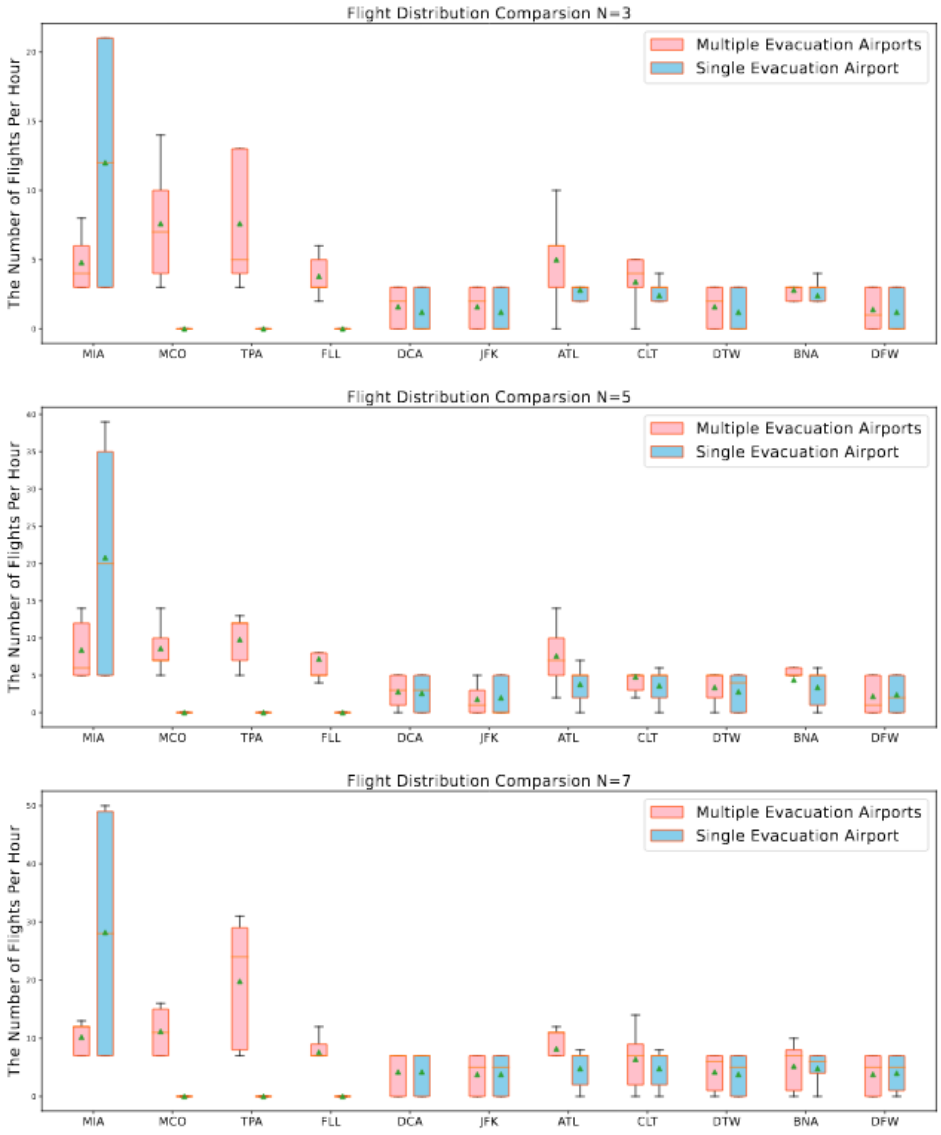


Figure 24. Comparison of hourly departure flight distribution under single-airport and multi-airport scenarios.

Evacuation Situation	Timestamp	MIA	MCO	TPA	FLL	DCA	JFK	ATL	CLT	DTW	BNA	DFW
Singel Evacuation Airport without Emergency	1	7	0	0	0	7	7	7	7	7	7	7
	2	7	0	0	0	7	7	8	7	7	7	7
	3	28	0	0	0	7	5	7	8	5	5	5
	4	50	0	0	0	0	0	2	2	0	4	1
	5	49	0	0	0	0	0	0	0	0	0	0
Multiple Evacuation Airports without Emergency	1	7	7	7	7	7	7	7	7	7	7	7
	2	7	11	8	7	7	7	11	9	7	8	7
	3	12	15	24	9	7	5	12	14	6	10	5
	4	8	7	29	3	0	0	11	2	1	1	0
	5	13	16	31	12	0	0	0	0	0	0	0
Multiple Evacuation Airports with Emergency	1	7	7	7	7	7	7	7	7	7	7	7
	2	7	10	7	7	7	7	10	7	7	7	7
	3	10	10	23	7	4	5	0	12	5	5	2
	4	7	11	15	10	0	0	0	12	0	12	0
	5	9	6	16	12	0	0	0	0	0	0	0

Table 8. Specific number of planes taking off at each airport every hour in different evacuation situations.

(3) Scenario 4 Results

In the fourth experimental setting, N is set as 7, the emergency happens in the third timestamp, and the major supporting airport ATL shuts down due to some emergencies. Under this situation, there are two special circumstances that will affect the evacuation flight dispatch plans. Firstly, due to the sudden airport closure of ATL, the flight which will arrive at the ATL airport at the third timestamp will not be able to participate in the subsequent flight dispatch plans. According to the results of the experiment, there are 15 aircraft that cannot participate in the subsequent dispatch plans. Secondly, the flight which will arrive at the ATL airport after the third timestamp and departure before the third timestamp from other airports will be dispatched to the airport which is closest to the current aircraft position. As shown in Figure 25, most of these kinds of flights will be redirected to the CLT, which is the best real-time dispatch modification because CLT is the closest airport to the ATL.

Table 8 shows the specific number of flights that take off from each airport hourly. In the emergency evacuation result, the aircraft will not take off from ATL at the third time step.

Furthermore, the most redirected aircraft cause an increasing number of flights that are taken off from the CLT and BNA. Finally, those 15 aircraft that cannot participate in the subsequent dispatch plans are the principal reason for the decrease in the number of total evacuation flights. The result also turns out that the dispatch system can generate timely and effective evacuation plans to deal with such emergencies.

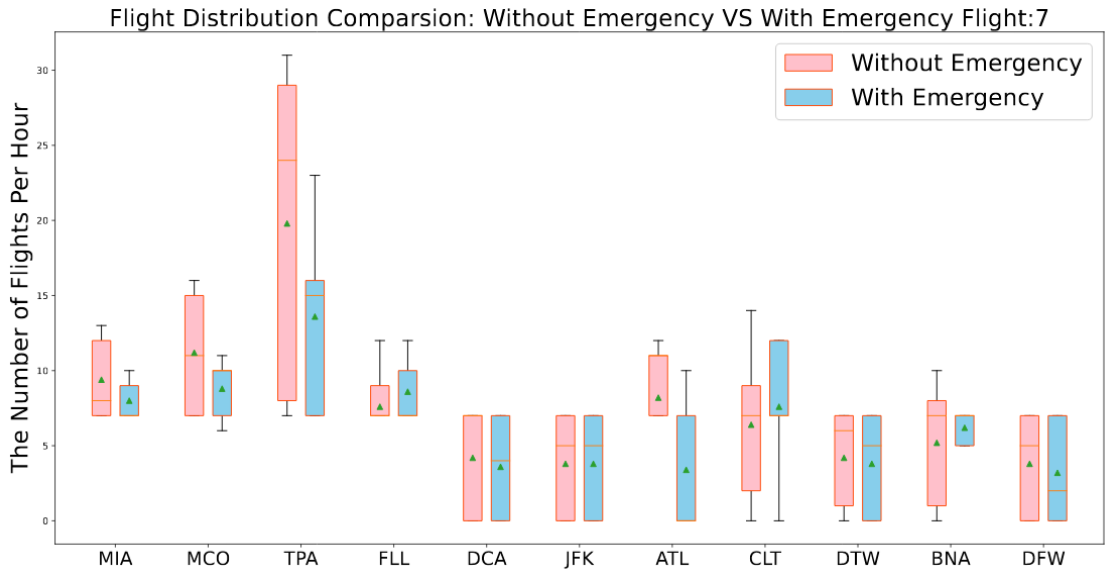


Figure 25. Comparison of hourly departure flight distribution with emergency or without emergency.

(4) Central Allocation vs. Uniform Allocation

The comparison results for different allocation strategies in the third setting are illustrated in Figure 26. The evacuation airports are assumed to act as different roles during the evacuation process, the dispatch system adopts different evacuation flight generation strategies. Therefore, the third setting is utilized to test our online allocation strategy methods. Figure 26 shows that when the dispatch system does not utilize any optimizations, the flights are more likely to be allocated to airports that are closer to the supporting airport and have less negative workload

impact on the supporting airports. However, the evacuation airport may play different roles in the evacuation process. Firstly, the Central Allocation is adopted when Miami airport is the center of the hurricane, the flights should be arranged to focus on the Miami airport and the other evacuation airport is used to disperse the capacity pressure of the Miami airport. Since the FLL and MIA locate near to the Miami, the desired result of the flights which are dispatched to take off from MIA and FLL accounted for 60%, and MIA takes 50% flights. The rest of the flights should be arranged to MCO and TPA, which is based on the population. Then the balance weight is set as $0.5: 0.15: 0.25: 0.1$. Secondly, Uniform Allocation is adopted when all evacuation airports play the same important role in the evacuation process, so the flights should be dispatched almost equally to all evacuation airports. Then the balance weight is set as $0.2: 0.25: 0.15: 0.4$. Figure 26 shows that when the airport evacuation dispatch system takes different allocation strategies, the flight distribution satisfies the system's expectation of actual demand, which can produce centralized or equalized flight plans.

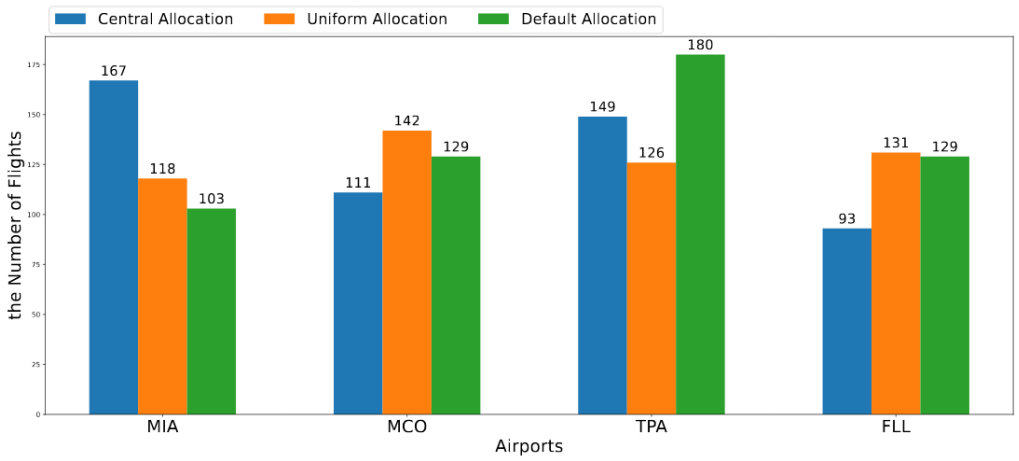


Figure 26. Comparison of central allocation, uniform allocation and default allocation.

Conclusion

In this project, we proposed a novel evacuation flight dispatching framework under emergency that aims to optimize the air route network's long-term efficiency, as well as satisfying instant evacuation demands. The flight dispatch is modeled as an MDP, where the value of the aircraft-airway pair is obtained by the flight's utility and the future expected air traffic control index value (transaction value) learned from synthetic data based on historical records. Matching between multiple aircraft and potential evacuation airway is then solved by the Hungarian algorithm. The real-world case study based on Hurricane Irma demonstrates the effectiveness and efficiency of the proposed L2D and ATCI-based reward function compared with the baseline univariate methods. To tackle the uncertainty problem during evacuation process, we developed the corresponding machine learning models that were utilized to predict the air mobility indicators in the online planning stage. The results show its potential to be deployed in the real-world disaster management in the future.

Our proposed L2D is the first work on the evacuation flight planning under emergency. To further improve the utility of our framework, we need to address the following limitations from the perspective of system design and methodologies.

Limitation of System Design

The optimal solution cannot be obtained because the start state and end state of online transaction candidates are generated randomly. Data analysis on demand variations during hurricane evacuation helps understand evacuee's behavior such as "when people start to evacuate". Also adding predictive modeling to better understand the future demand could add

more value while being applied in a real-world case like emergency management. Another concern of this study is the lack of operation data such as runway capacity and the maintenance time.

Limitation of Methodology

We used a table that stores spatiotemporal values for each aircraft. It has two major problems. Firstly, only the available states in the table can be extracted for further planning phase. In other words, the generalization of tabular method cannot be guaranteed. Considering the strong representational power of neural networks, deep reinforcement learning, in which deep neural networks are used for value function approximation, may improve the scope of applicability to the real world.

For future work, we are interested in developing a holistic large-scale flight dispatching framework based on passenger demand and weather fluctuation, which could be applied in nationwide evacuation planning. Meanwhile, it is beneficial to investigate the approach which can better capture the dynamics of the air route network during disaster evacuation.

References

Ayhan, S., Costas, P. & Samet, H. (2018, July). Predicting estimated time of arrival for commercial flights. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 33—42. https://doi.org/10.1145/3219819.3219874

Baker, E. J. (1991). Hurricane evacuation behavior. *International Journal of Mass Emergencies and Disasters*, *9*(2), 287—310. https://training.fema.gov/emiweb/downloads/ijems/articles/hurricane%20evacuation%20behavior.pdf

Barrett, B., Ran, B. & Pillai, R. (2000). Developing a dynamic traffic management modeling framework for hurricane evacuation. *Transportation Research Record*, *1733*(1), 115—121. https://doi.org/10.3141/1733-15

Barto, G. A. (1995). Reinforcement learning and dynamic programming. *IFAC Proceedings Volumes*, *28*(15), 407—412. https://doi.org/10.1016/B978-0-08-042370-8.50010-0

Bayram, V. (2016). Optimization models for large scale network evacuation planning and management: A literature review. *Surveys in Operations Research and Management Science*, *21*(2), 63—84. https://doi.org/10.1016/j.sorms.2016.11.001

Bayram, V., Tansel, B.Ç. & Yaman, H. (2015). Compromising system and user interests in shelter location and evacuation planning. *Transportation Research Part B: Methodological*, *72*, 146—163. https://doi.org/10.1016/j.trb.2014.11.010

Bazzan, A.L. & Grunitzki, R. (2016, July). A multiagent reinforcement learning approach to en-route trip building. In *2016 International Joint Conference on Neural Networks (IJCNN)*, 5288—5295. https://doi.org/10.1109/IJCNN.2016.7727899

Beckmann, M., McGuire, C. B., & Winsten, C. B. (1956). Studies in the Economics of Transportation: Technical Report.

Bhaskar, U., Fleischer, L. & Anshelevich, E. (2015). A Stackelberg strategy for routing flow over time. *Games and Economic Behavior*, *92*, 232—247. https://doi.org/10.1016/j.geb.2013.09.004

Bliemer, M.C., Raadsen, M.P., Brederode, L.J., Bell, M.G., Wismans, L.J. & Smith, M.J. (2017). Genetics of traffic assignment models for strategic transport planning. *Transport Reviews*, *37*(1), 56—78. https://doi.org/10.1080/01441647.2016.1207211

Centers for Disease Control and Prevention. (2018). Preparedness and safety messaging for hurricanes, flooding, and similar disasters. Atlanta, GA: U.S. Department of Health and Human Services.

Chan, W.K.V., Son, Y.J. & Macal, C.M. (2010, December). Agent-based simulation tutorial-simulation of emergent behavior and differences between agent-based simulation and discrete-event simulation. *In Proceedings of the 2010 Winter Simulation Conference*, 135—150. https://doi.org/10.1109/WSC.2010.5679168

Chen, X., Meaker, J.W. & Zhan, F.B. (2006). Agent-based modeling and analysis of hurricane evacuation procedures for the Florida Keys. *Natural Hazards*, *38*(3), 321—338, https://doi.org/10.1007/s11069-005-0263-0

Chow, H.F.A. (2007). System optimal traffic assignment with departure time choice. University of London, University College London (United Kingdom).

Coppola, D.P. (2006). Introduction to international disaster management. Elsevier.

Correa, J.R., Schulz, A.S. & Stier-Moses, N.E. (2007). Fast, fair, and efficient flows in networks. *Operations Research*, *55*(2), 215—225. https://doi.org/10.1287/opre.1070.0383

Coutinho-Rodrigues, J., Tralhão, L. & Alçada-Almeida, L. (2012). Solving a location-routing problem with a multiobjective approach: the design of urban evacuation plans. *Journal of Transport Geography*, *22*, 206—218.

Cova, T.J. & Johnson, J.P. (2003). A network flow model for lane-based evacuation routing. *Transportation research part A: Policy and Practice*, *37*(7), 579—604. https://doi.org/10.1016/S0965-8564(03)00007-7

Daganzo, C.F. (1994). The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological*, *28*(4), 269—287. https://doi.org/10.1016/0191-2615(94)90002-7

Daganzo, C.F. (1995). The cell transmission model, part II: network traffic. *Transportation Research Part B: Methodological*, *29*(2), 79—93. https://doi.org/10.1016/0191-2615(94)00022-R

Daskin, M.S. (1985). Urban transportation networks: Equilibrium analysis with mathematical programming methods, Prentice Hall. https://www.jstor.org/stable/25768196

De Silva, F.N. & Eglese, R.W. (2000). Integrating simulation modelling and GIS: spatial decision support systems for evacuation planning. *Journal of the Operational Research Society*, *51*(4), 423—430. https://doi.org/10.1057/palgrave.jors.2600879

Dixit, V. & Wolshon, B. (2014). Evacuation traffic dynamics. *Transportation Research Part C: Emerging Technologies*, *49*, 114—125. https://doi.org/10.1016/j.trc.2014.10.014

Edara, P., Sharma, S. & McGhee, C. (2010). Development of a large-scale traffic simulation model for hurricane evacuation—methodology and lessons learned. *Natural Hazards Review*, *11*(4), 127—139. https://doi.org/10.1061/(ASCE)NH.1527-6996.0000015

Faturechi, R. & Miller-Hooks, E. (2014). Travel time resilience of roadway networks under disaster. *Transportation Research Part B: Methodological, 70*, 47—64. https://doi.org/10.1016/j.trb.2014.08.007

Fu, J., Kumar, A., Nachum, O., Tucker, G. & Levine, S. (2020). D4rl: Datasets for deep data-driven reinforcement learning. arXiv preprint arXiv:2004.07219

Fu, H. & Wilmot, C. G. (2004). Sequential logit dynamic travel demand model for hurricane evacuation. *Transportation Research Record*, *1882*(1), 19—26. https://doi.org/10.3141/1882-03

Fu, H. & Wilmot, C. G. (2007). Static versus dynamic and aggregate versus disaggregate: A comparison between practice and research in hurricane evacuation travel demand modeling (No. 07-0646).

Fu, H., Wilmot, C. G. & Baker, E.J. (2006). Sequential logit dynamic travel demand model and its transferability. *Transportation Research Record*, *1977*(1), 17—26. https://doi.org/10.1177/0361198106197700103

Galindo, G. & Batta, R. (2013). Review of recent developments in OR/MS research in disaster operations management. *European Journal of Operational Research*, *230*(2), 201—211. https://doi.org/10.1016/j.ejor.2013.01.039

Gianazza, D. (2017, June). Learning air traffic controller workload from past sector operations. In *ATM Seminar*, 12th USA/Europe Air Traffic Management R&D Seminar, Seattle, USA.

Graves, A., Mohamed, A. R. & Hinton, G. (2013, May). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, 6645—6649. https://doi.org/10.1109/ICASSP.2013.6638947

Grunitzki, R., de Oliveira Ramos, G. & Bazzan, A. L. C. (2014, October). Individual versus difference rewards on reinforcement learning for route choice. In *2014 Brazilian Conference on Intelligent Systems*, 253—258. https://doi.org/10.1109/BRACIS.2014.53

Hasan, S. & Ukkusuri, S.V. (2011). A threshold model of social contagion process for evacuation decision making. *Transportation research part B: methodological*, *45*(10), 1590—1605. https://doi.org/10.1016/j.trb.2011.07.008

He, K., Zhang, X., Ren, S. & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770—778.

He, K., Zhang, X., Ren, S. & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, 1026—1034.

Hofinger, G., Zinke, R. & Künzer, L. (2014). Human factors in evacuation simulation, planning, and guidance. *Transportation Research Procedia*, *2*, 603—611. https://doi.org/10.1016/j.trpro.2014.09.101

Hu, J. & Wellman, M. P. (1998, June). Multiagent reinforcement learning: theoretical framework and an algorithm. *ICML*, *98*, 242—250. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.138.2589&rep=rep1&type=pdf

Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700—4708.

International Federation of Red Cross and Red Crescent Societies. (n.d.). About disaster management. Retrieved May, 2021, from https://www.ifrc.org/disaster-preparedness

Jahn, O., Möhring, R. H., Schulz, A. S. & Stier-Moses, N. E. (2005). System-optimal routing of traffic flows with user constraints in networks with congestion. *Operations Research*, *53*(4), 600—616. https://doi.org/10.1287/opre.1040.0197

Jiang, Y., Liu, Y., Liu, D. & Song, H. (2020, August). Applying Machine Learning to Aviation Big Data for Flight Delay Prediction. In *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on*

Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), 665—672. https://doi.org/10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00114

Jha, M., Moore, K. & Pashaie, B. (2004). Emergency evacuation planning with microscopic traffic simulation. *Transportation Research Record*, *1886*(1), 40—48.

Kim, Y.J., Choi, S., Briceno, S. & Mavris, D. (2016, September). A deep learning approach to flight delay prediction. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, 1—6. https://doi.org/10.1109/DASC.2016.7778092

Kim, S., Lewis, M. E. & White, C. C. (2005). Optimal vehicle routing with real-time traffic information. *IEEE Transactions on Intelligent Transportation Systems*, *6*(2), 178—188. https://doi.org/10.1109/TITS.2005.848362

Kim, G., Ong, Y. S., Cheong, T. & Tan, P. S. (2016). Solving the dynamic vehicle routing problem under traffic congestion. *IEEE Transactions on Intelligent Transportation Systems*, *17*(8), 2367—2380. https://doi.org/10.1109/TITS.2016.2521779

Kopardekar, P. (2000). Dynamic density: A review of proposed variables. FAA WJHTC internal document. Overall conclusions and recommendations, Federal Aviation Administration.

Kumar, A., Fu, J., Tucker, G. & Levine, S. (2019). Stabilizing off-policy q-learning via bootstrapping error reduction. arXiv preprint arXiv:1906.00949

Kumar, S., Solanki, V. K., Choudhary, S. K., Selamat, A. & González Crespo, R. (2020). Comparative Study on Ant Colony Optimization (ACO) and K-Means Clustering Approaches for Jobs Scheduling and Energy Optimization Model in Internet of Things (IoT). *International Journal of Interactive Multimedia & Artificial Intelligence*, *6*(1), 107—116. https://doi.org/10.9781/ijimai.2020.01.003

Lämmel, G., Grether, D. & Nagel, K. (2010). The representation and implementation of time-dependent inundation in large-scale microscopic evacuation simulations. *Transportation Research Part C: Emerging Technologies*, *18*(1), 84—98. https://doi.org/10.1016/j.trc.2009.04.020

Laudeman, I. V., Shelden, S. G., Branstrom, R. & Brasil, C. L. (1998). Dynamic density: An air traffic management metric, NASA/TM-1998-112226.

Lindell, M. K., Kang, J. E. & Prater, C. S. (2011). The logistics of household hurricane evacuation. *Natural Hazards*, *58*(3), 1093—1109. https://doi.org/10.1007/s11069-011-9715-x

Lindell, M. K. & Perry, R. W. (1987). Warning mechanisms in emergency response systems. *International Journal of Mass Emergencies and Disasters*, *5*(2), 137—153.

Lindell, M. K. & Prater, C. S. (2007). Critical behavioral assumptions in evacuation time estimate analysis for private vehicles: Examples from hurricane research and planning. *Journal of Urban Planning and Development*, *133*(1), 18—29. https://doi.org/10.1061/(ASCE)0733-9488(2007)133:1(18)

Lindsey, A. B., Donovan, M., Smith, S., Radunovich, H., & Gutter, M. (2021, February). Impacts of technological disasters. University of Florida.

Li, Z. & Zhao, X. (2008). Integrated-equilibrium routing of traffic flows with congestion. *IFAC Proceedings*, *41*(2), 16065—16070. https://doi.org/10.3182/20080706-5-KR-1001.02715

Liu, Y. & Hansen, M. (2018). Predicting aircraft trajectories: a deep generative convolutional recurrent neural networks approach. arXiv preprint arXiv:1812.11670

Lou, P., Liu, Q., Zhou, Z., Wang, H. & Sun, S. X. (2012). Multi-agent-based proactive–reactive scheduling for a job shop. *The International Journal of Advanced Manufacturing Technology*, *59*(1), 311—324. https://doi.org/10.1007/s00170-011-3482-4

Luo, S. (2020). Dynamic scheduling for flexible job shop with new job insertions by deep reinforcement learning. *Applied Soft Computing*, *91*(21), 106208. https://doi.org/10.1016/j.asoc.2020.106208

Macal, C. & North, M. (2010). Tutorial on agent-based modeling and simulation. *Journal of Simulation*, *4*(3), 151—162. https://doi.org/10.1109/WSC.2005.1574234

Mao, C. & Shen, Z. (2018). A reinforcement learning framework for the adaptive routing problem in stochastic time-dependent network. *Transportation Research Part C: Emerging Technologies*, *93*, 179—197. https://doi.org/10.1016/j.trc.2018.06.001

Matignon, L., Laurent, G. J. & Le Fort-Piat, N. (2012). Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems. *The Knowledge Engineering Review*, *27*(1), 1—31. https://doi.org/10.1017/S0269888912000057

McCarthy, N., Karzand, M. & Lecue, F. (2019, July). Amsterdam to Dublin eventually delayed? LSTM and transfer learning for predicting delays of low cost airlines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*(1), 9541—9546. https://doi.org/10.1609/aaai.v33i01.33019541

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G. & Petersen, S. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529—533. https://doi.org/10.1038/nature14236

Mogford, R. H., Guttman, J. A., Morrow, S. L. & Kopardekar, P. (1995). The Complexity Construct in Air Traffic Control: A Review and Synthesis of the Literature, Technical note, DOT/FAA/CT-TN95/22.

Montz, T. & Zhang, Z. (2015). Modeling regional hurricane evacuation events: calibration and validation. *Natural Hazards Review*, *16*(4), 04015007. https://doi.org/10.1061/(ASCE)NH.1527-6996.0000181

Murray-Tuite, P., Yin, W., Ukkusuri, S. V. & Gladwin, H. (2012). Changes in evacuation decisions between Hurricanes Ivan and Katrina. *Transportation Research Record*, *2312*(1), 98—107. https://doi.org/10.3141/2312-10

Na, H. S. & Banerjee, A. (2015). A disaster evacuation network model for transporting multiple priority evacuees. *IIE Transactions*, *47*(11), 1287—1299. https://doi.org/10.1080/0740817X.2015.1040929

Na, H. S. & Banerjee, A. (2019). Agent-based discrete-event simulation model for no-notice natural disaster evacuation planning. *Computers & Industrial Engineering*, *129*, 44—55. https://doi.org/10.1016/j.cie.2019.01.022

National Centers for Environmental Information. (n.d.). *Global forecast system*. Retrieved May, 2021, from https://www.ncei.noaa.gov/products/weather-climate-models/global-forecast

National Oceanic & Atmospheric Administration. (2019, June). North Atlantic Hurricane Basin (1851-2018) Comparison of Original and Revised HURDAT (Hurricane Database).

National Weather Service. (2017, September). *Hurricane Irma local report/summary*. https://www.weather.gov/mfl/hurricaneirma

Nguyen, T. T., Nguyen, N. D. & Nahavandi, S. (2020). Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications. *IEEE Transactions on Cybernetics*, *50*(9), 3826—3839. https://doi.org/10.1109/TCYB.2020.2977374

Pan, X., Han, C. S., Dauber, K. & Law, K. H. (2007). A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations. *AI& Society*, *22*(2), 113—132. https://doi.org/10.1007/s00146-007-0126-1

Pel, A. J., Bliemer, M. C. and Hoogendoorn, S. P. (2012). A review on travel behaviour modelling in dynamic traffic simulation models for evacuations. *Transportation*, *39*(1), 97—123. https://doi.org/10.1007/s11116-011-9320-6

Pelechano, N. & Malkawi, A. (2008). Evacuation simulation models: Challenges in modeling high rise building evacuation with cellular automata approaches. *Automation in Construction*, *17*(4), 377—385. https://doi.org/10.1016/j.autcon.2007.06.005

Pidd, M., De Silva, F. N. & Eglese, R. W. (1996). A simulation model for emergency evacuation. *European Journal of Operational Research*, *90*(3), 413—419. https://doi.org/10.1016/0377-2217(95)00112-3

Prandini, M., Piroddi, L., Puechmorel, S. & Brázdilová, S. L. (2011). Toward air traffic complexity assessment in new generation air traffic management systems. *IEEE Transactions on Intelligent Transportation Systems*, *12*(3), 809—818. https://doi.org/10.1109/TITS.2011.2113175

Ramos, G. D. O., Bazzan, A. L. & da Silva, B. C. (2018). Analysing the impact of travel information for minimising the regret of route choice. *Transportation Research Part C: Emerging Technologies*, *88*, 257—271. https://doi.org/10.1016/j.trc.2017.11.011

Richardson, B. C. (1979). *Limitations on the use of mathematical models in transportation policy analysis*. University Microfilms International, Ann Arbor, Michigan.

Ritchie, H. (n.d). OFDA/CRED International Disaster Data. Our World in Data.

Ritchie, H. & Roser, M. (2020). Natural Disasters. Our World in Data.

Rogers, G. O. & Sorensen, J. H. (1991). Diffusion of emergency warning: comparing empirical and simulation results. In *Risk analysis*, *8*, 117—134. https://doi.org/ 10.1007/978-1-4899-0730-1_14

Roughgarden, T. A. (2002). *Selfish routing*. Cornell University.

Radio Technical Commission for Aeronautics (RTCA) Force (1995). Final Report Task Force 3 Free Flight Implementation. United States: RTCA.

Schatz, K., Schlittenlacher, J., Ullrich, D., Rüppel, U. & Ellermeier, W. (2014). Investigating human factors in fire evacuation: a serious-gaming approach. In *Pedestrian and Evacuation Dynamics 2012*, 1113—1121. Springer, Cham. https://doi.org/10.1007/978-3-319-02447-9_91

Scheidegger, A. P. G., Pereira, T. F., de Oliveira, M. L. M., Banerjee, A. & Montevechi, J. A. B. (2018). An introductory guide for hybrid simulation modelers on the primary simulation methods in industrial engineering identified through a systematic review of the literature. *Computers & Industrial Engineering*, *124*, 474—492. https://doi.org/10.1016/j.cie.2018.07.046

Schulz, A. S. & Stier-Moses, N. E. (2006). Efficiency and fairness of system-optimal routing with user constraints. *Networks: An International Journal*, *48*(4), 223—234. https://doi.org/10.1002/net.20133

Sheu, J. B. & Pan, C. (2014). A method for designing centralized emergency supply network to respond to large-scale natural disasters. *Transportation Research Part B: Methodological*, *67*, 284—305. https://doi.org/10.1016/j.trb.2014.05.011

Shi, X., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K. & Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, 802—810. https://proceedings.neurips.cc/paper/2015/file/07563a3fe3bbe7e3ba84431ad9d055af-Paper.pdf

Shou, Z. & Di, X. (2020). Multi-agent reinforcement learning for dynamic routing games: A unified paradigm. arXiv preprint arXiv:2011.10915

Siebers, P. O., Macal, C. M., Garnett, J., Buxton, D. & Pidd, M. (2010). Discrete-event simulation is dead, long live agent-based simulation! *Journal of Simulation*, *4*(3), 204—210. https://doi.org/10.1057/jos.2010.14

Songchitruksa, P., Henk, R., Venglar, S. & Zeng, X. (2012). Dynamic traffic assignment evaluation of hurricane evacuation strategies for the Houston–Galveston, Texas, region. *Transportation Research Record*, *2312*(1), 108—119. https://doi.org/10.3141/2312-11

Sorensen, J. H. (2000). Hazard warning systems: Review of 20 years of progress. *Natural Hazards Review*, *1*(2), 119—125. https://doi.org/10.1061/(ASCE)1527-6988(2000)1:2(119)

Southworth, F. (1991). *Regional evacuation modeling: A state-of-the-art review, technical report, Department of Defense*. (Report No. ORNL/TM-11740). Department of Defense.

Sridhar, B., Sheth, K. S. & Grabbe, S. (1998, December). Airspace complexity and its application in air traffic management. In *2nd USA/Europe Air Traffic Management R&D Seminar*, 1—6.

https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.208.8837&rep=rep1&type=pdf

Sutton, R. S. & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Tolson, J. J. (1973). *Airmobility, 1961-1971*. *90*(4). Department of the Army.

Ukkusuri, S. V., Hasan, S., Luong, B., Doan, K., Zhan, X., Murray-Tuite, P. & Yin, W. (2017). A-rescue: An agent based regional evacuation simulator coupled with user enriched behavior. *Networks and Spatial Economics*, *17*(1), 197—223. https://doi.org/10.1007/s11067-016-9323-0

United Nations. (2021, July). *Global SDG indicators database* [Data set]. Statistics Division from Department of Economic and Social Affairs. https://unstats.un.org/sdgs/indicators/database/

United States Department of Transportation. (2021). *Bureau of Transportation statistics – on-time: Reporting carrier on-time performance (1987-present)*. https://www.transtats.bts.gov/Fields.asp?gnoyr_VQ=FGJ

Wagner, N. & Agrawal, V. (2014). An agent-based simulation system for concert venue crowd evacuation modeling in the presence of a fire disaster. *Expert Systems with Applications*, *41*(6), 2807—2815. https://doi.org/10.1016/j.eswa.2013.10.013

Wardrop, J. G. (1952). Some theoretical aspects of road traffic research (Road Paper No. 36). *Proceedings of the Institution of Civil Engineers*, *1*(3), 325—362. https://doi.org/10.1680/ipeds.1952.11259

Weiszer, M., Chen, J. & Locatelli, G. (2015). An integrated optimisation approach to airport ground operations to foster sustainability in the aviation sector. *Applied Energy*, *157*, 567—582. https://doi.org/10.1016/j.apenergy.2015.04.039

Welch, J. D. (2015). *En route sector capacity model final report* (Report No. ATC-426). National Technical Information Service. https://www.ll.mit.edu/sites/default/files/publication/doc/2018-12/Welch_2015_ATC-426.pdf

Wu, H. C., Lindell, M. K. & Prater, C. S. (2012). Logistics of hurricane evacuation in Hurricanes Katrina and Rita. *Transportation Research Part F: Traffic Psychology and Behaviour*, *15*(4), 445—461. https://doi.org/10.1016/j.trf.2012.03.005

Wu, S., Shuman, L., Bidanda, B., Kelley, M., Sochats, K. & Balaban, C. (2008). Agent-based discrete event simulation modeling for disaster responses. *IIE Annual Conference Proceedings*, 1908—1913. https://www.researchgate.net/profile/Cd-Balaban/publication/266497864_Agent-based_Discrete_Event_Simulation_Modeling_for_Disaster_Responses/links/543bdbd90cf204cab1db42d1/Agent-based-Discrete-Event-Simulation-Modeling-for-Disaster-Responses.pdf

Xu, Z., Li, Z., Guan, Q., Zhang, D., Li, Q., Nan, J., Liu, C., Bian, W. & Ye, J. (2018, July). Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 905—913. https://doi.org/10.1145/3219819.3219824

Yamada, T. (1996). A network flow approach to a city emergency evacuation planning. *International Journal of Systems Science*, *27*(10), 931—936. https://doi.org/10.1080/00207729608929296

Yang, Y., Luo, R., Li, M., Zhou, M., Zhang, W. & Wang, J. (2018, July). Mean field multi-agent reinforcement learning. In *International Conference on Machine Learning*, 5571—5580. arXiv:2108.02731

Yazici, M. A. (2010). *Introducing uncertainty into evacuation modeling via dynamic traffic assignment with probabilistic demand and capacity constraints*, Rutgers, The State University of New Jersey. https://doi.org/10.7282/T3Q52PBV

Yazici, M. A. & Ozbay, K. (2008). Evacuation modelling in the United States: Does the demand model choice matter? *Transport Reviews*, *28*(6), 757—779. https://doi.org/10.1080/01441640802041812

Yin, W., Murray-Tuite, P., Ukkusuri, S. V. & Gladwin, H. (2014). An agent-based modeling system for travel demand simulation for hurricane evacuation. *Transportation Research Part C: Emerging Technologies*, *42*, 44—59. https://doi.org/10.1016/j.trc.2014.02.015

Yu, Y., Han, K. & Ochieng, W. (2020). Day-to-day dynamic traffic assignment with imperfect information, bounded rationality and information sharing. *Transportation Research Part C: Emerging Technologies*, *114*, 59—83. https://doi.org/10.1016/j.trc.2020.02.004

Yuan, S., Chun, S. A., Spinelli, B., Liu, Y., Zhang, H. & Adam, N. R. (2017). Traffic evacuation simulation based on multi-level driving decision model. *Transportation Research Part C: Emerging Technologies*, *78*, 129—149. https://doi.org/10.1016/j.trc.2017.03.001

Zaremba, W., Sutskever, I. & Vinyals, O. (2014). Recurrent neural network regularization. arXiv:1409.2329

Zhang, B., Chan, W. K. V. & Ukkusuri, S. V. (2014). On the modelling of transportation evacuation: an agent-based discrete-event hybrid-space approach. *Journal of Simulation*, *8*(4), 259—270. https://doi.org/10.1057/jos.2014.3

Zhang, K., Liu, Y., Wang, J., Song, H. & Liu, D. (2020, September). Tree-Based Airspace Capacity Estimation. In *2020 Integrated Communications Navigation and Surveillance Conference (ICNS)*, 5C1-1—5C1-5. https://doi.org/10.1109/ICNS50378.2020.9222986

Zhang, K., Jiang, Y., Liu, D. & Song, H. (2020, November). Spatio-Temporal Data Mining for Aviation Delay Prediction. In *2020 IEEE 39th International Performance Computing and Communications Conference (IPCCC)*, 1—7. https://doi.org/10.1109/IPCCC50635.2020.9391561

Zhou, B. & Li, X. (2012, June). User equilibrium with length constrained users. In *2012 Fifth International Joint Conference on Computational Sciences and Optimization*, 494—499. https://doi.org/10.1109/CSO.2012.114

Zhou, B., Song, Q., Zhao, Z. & Liu, T. (2020). A reinforcement learning scheme for the equilibrium of the in-vehicle route choice problem based on congestion game. *Applied Mathematics and Computation*, *371*, 124895. https://doi.org/10.1016/j.amc.2019.124895

Zhu, Y., Ozbay, K., Xie, K. & Yang, H. (2019, October). Modeling of Incident-Induced Capacity Loss for Hurricane Evacuation Simulation. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 613—618. https://doi.org/10.1109/ITSC.2019.8917250

Ziliaskopoulos, A. K. (2000). A linear programming model for the single destination system optimum dynamic traffic assignment problem. *Transportation Science*, *34*(1), 37—49. https://doi.org/10.1287/trsc.34.1.37.12281

Zinke, R., Hofinger, G. & Künzer, L. (2014). Psychological aspects of human dynamics in underground evacuation: Field experiments. In *Pedestrian and Evacuation Dynamics 2012*, 1149—1162. Springer, Cham. https://doi.org/10.1007/978-3-319-02447-9_94