An Artificial-Neural-Network-Based Model for Real-Time Dispatching of Electric Autonomous Taxis

Liang Hu
Iowa State University

Jing Dong
Iowa State University, jingdong@iastate.edu

Follow this and additional works at: https://lib.dr.iastate.edu/ccee_pubs

Part of the Transportation Engineering Commons

The complete bibliographic information for this item can be found at https://lib.dr.iastate.edu/ccee_pubs/291. For information on how to cite this item, please visit http://lib.dr.iastate.edu/howtocite.html.
An Artificial-Neural-Network-Based Model for Real-Time Dispatching of Electric Autonomous Taxis

Abstract
This paper presents a real-time dispatching model for electric autonomous vehicle (EAV) taxis that combines mathematical programming and machine learning. The EAV taxi dispatching problem is formulated and solved as an integer linear program that maximizes the total reward for serving customers. The optimal dispatch solutions are generated by simulating electric autonomous taxis that are dispatched by the optimization model. The artificial-neural-network-(ANN)-based model was trained using the optimization model's dispatch solutions to learn the optimal dispatch strategies. Although the dispatch decisions made by the ANN-based model are not optimal, the system's performance is very close to the optimization dispatch model in terms of customer service and taxis' operational efficiency. In addition, the ANN-based dispatch model runs much faster. By comparing with current taxis, it was found that the EAV taxis dispatched by our ANN-based model can improve operational efficiency by reducing empty travel distance. EAV taxis can also reduce fleet size by 15% while maintaining a comparable level of service with the current taxi fleet.

Keywords
Artificial neural network, electric and autonomous vehicle, integer linear program, simulation, taxi dispatch

Disciplines
Transportation Engineering

Comments
An Artificial-Neural-Network-Based Model for Real-Time Dispatching of Electric Autonomous Taxis

Liang Hu and Jing Dong, Senior Member, IEEE

Abstract—This paper presents a real-time dispatching model for electric autonomous vehicle (EAV) taxis that combines mathematical programming and machine learning. The EAV taxi dispatching problem is formulated and solved as an integer linear program that maximizes the total reward for serving customers. The optimal dispatch solutions are generated by simulating electric autonomous taxis that are dispatched by the optimization model. The artificial-neural-network-(ANN)-based model was trained using the optimization model’s dispatch solutions to learn the optimal dispatch strategies. Although the dispatch decisions made by the ANN-based model are not optimal, the system’s performance is very close to the optimization dispatch model in terms of customer service and taxis’ operational efficiency. In addition, the ANN-based dispatch model runs much faster. By comparing with current taxis, it was found that the EAV taxis dispatched by our ANN-based model can improve operational efficiency by reducing empty travel distance. EAV taxis can also reduce fleet size by 15% while maintaining a comparable level of service with the current taxi fleet.

Index Terms—Artificial neural network, electric and autonomous vehicle, integer linear program, simulation, taxi dispatch.

I. INTRODUCTION

Electric vehicles have been adopted in the taxi fleet in many cities around the globe [1]–[5]. Ride-hailing companies, such as Uber and Lyft, have started testing autonomous vehicle (AV) taxis on public roads in the U.S. [6]–[7]. In 2017, Waymo started providing a ride-hailing taxi service in Arizona using a fleet of electric and autonomous vehicles (EAVs) [8].

EAV taxis have significant advantages over traditional taxis and ride-hailing services. Traditional taxis usually cruise on the streets to search for customers, generating many empty miles. With ride-hailing apps, a customer request is sent to nearby drivers. The customer will be picked up by the first driver who accepts the request. Therefore, the match between the customer and driver might not be optimal. By contrast, EAV taxis can operate in a collaborative manner based on the customer (e.g., location, trip distance, and waiting time) and taxi (e.g., location, status, and battery state-of-charge) information and potentially improve operational efficiency and customer experience. In addition, the fleet size of EAV taxis can be more easily adjusted based on customer demand. Replacing fossil-fuel-powered taxis with EAV taxis also reduces tailpipe emissions and saves energy costs [9]. Finally, charging can be better coordinated through vehicle-to-vehicle and vehicle-to-infrastructure communications to avoid an insufficient range to reach a destination and to schedule charging while idling.

Adopting EAVs for different kinds of mobility services has been discussed in only a few previous studies. Chen et al. [10] used agent-based models to simulate the driving and charging activities of shared EAVs. They found that one shared EAV can replace 3.7–6.8 private vehicles. Kang et al. [11] designed an EAV-sharing system and optimized fleet size, charging infrastructure, vehicle assignment, and service fee. Jäger et al. [12] proposed an agent-based simulation approach for a shared EAV on-demand mobility system. It was found that shared EAVs can provide both a high level of service and high vehicle utilization. Iacobucci et al. [13] modeled the operations of EAVs with the one-way car-sharing service in Tokyo, Japan, and showed that EAV car sharing can provide the same level of transport service as private cars, while the fleet size can be reduced by 86%–90%. Dandl and Bogenberger [14] estimated that the fares of an EAV car-sharing system can be reduced by 29%–35% while achieving the same profit as a convenient existing car sharing system in Munich, Germany. The abovementioned studies all involve dispatching EAVs, which is the key to improving operational efficiency in shared mobility services.

The vehicle dispatch problems in mobility services have been studied extensively in the literature and can be classified into four categories. (1)Dispatching the nearest vehicle to the customer was adopted by Liao [15], Jung and Jayakrishnan [16], and Hyland and Mahmassani [17]. (2) Dispatching based on first-come-first-served queueing theory was adopted in Zhang and Pavone [18] and Jäger et al. [12]. (3) The dispatch problem has been formulated and solved as optimization problems with various objectives, such as maximizing profit [19]–[20]; minimizing total operating cost [21]; minimizing total idle travel distance [22]–[23]; minimizing the total pickup distance [17],[24]; and maximizing the number of customers served [25]–[26]. Major ride-hailing platforms,
such as Uber, Lyft and DiDi, have collected requests within a short time window and solved the optimization problem at the end of each time window, called ride request batching [24]. (4) Machine learning has been adopted to solve vehicle dispatching problems. Wen et al. [27] proposed a reinforcement learning approach that adopts a deep Q-network to adaptively move idle vehicles to high-demand areas in a shared on-demand mobility system. This approach performs effectively by reducing the waiting time of passengers and the travel distance of vehicles. Xu et al. [28] used reinforcement learning to solve a large-scale vehicle dispatch problem based on the historical orders of the ride-hailing company DiDi. Their reinforcement-learning model maximizes the expected revenue of the company over the long run. Their model is computationally efficient and has been implemented in DiDi’s real-time vehicle dispatch.

Built on the previous studies, this paper proposes an optimization-based dispatch model that maximizes the total reward for serving customers. By simulating EA V taxi operations, optimal dispatching decisions were generated by our optimization model and used for training an artificial neural network (ANN). This paper compares the performance of the resulting ANN-based dispatch with our optimization-based model in terms of objective function value, customer waiting time, operational efficiency, and computational time.

The main contributions of this paper are twofold. First, we propose an optimization-based dispatch model that considers the trade-off between taxi system efficiency and customer equity. Previous studies have considered multiple objectives associated with taxi operators and customers either in the objective function (e.g., [29]–[30]) or as a constraint [20]. However, the trade-off between taxi travel time for pickup and request waiting time has not been explicitly examined. Second, we propose an ANN-based dispatch model that has learned optimal dispatch strategies generated from an optimization model and is therefore suitable for real-time application.

II. DATA-DRIVEN SIMULATION OF EA V TAXI OPERATIONS

Our simulation framework for EA V taxi operations is presented in this section. The simulation process is illustrated in Fig. 1.

First, real-world taxi operation data were used to initialize EA V taxis’ simulated beginning-of-day status, including their initial locations and an initial battery state-of-charge (SOC) randomly set between 10% and 100%. We assume their electric range is 200 miles and the electricity consumption rate is 0.3 kWh/mile [31].

Our simulation framework includes 5 EA V taxi operational statuses—waiting, called, occupied, going to a charging station, and charging. Each EA V taxi’s initial status is set to waiting, that is, EA V taxis park somewhere and wait for customers.

At time step $T$, the set of available taxis is denoted as $I$ and the set of customer requests as $J$. The available taxis include the ones that are waiting and the ones that are charging and have gained at least 80% SOC. The customer requests consist of the new ones generated in time step $T$ and the unserved requests from the previous time steps. We assume customers’ maximum request waiting time is 15 minutes, beyond which the customers will cancel their request.

The simulation uses a dispatch model to match a taxi $i \in I$ and a customer request $j \in J$. If not dispatched, EA V taxis keep waiting or charging. When an EA V taxi is dispatched, its status changes to called. After the taxi picks up the customer, its status changes to occupied. After dropping off the customer at the destination, the EA V taxi checks whether charging is needed. If battery SOC is greater than or equal to 10%, the taxi keeps waiting near the drop-off location for the next customer. If the battery SOC is less than 10%, the taxi goes to the nearest charging station and starts charging. Once the SOC reaches 80%, the taxi becomes available for picking up new customers. If the taxi is not dispatched before it finishes charging, it keeps waiting near the charging station for incoming requests.

Our simulation framework uses 1-minute time intervals, so it solves the dispatch problem 1440 times per day. For each time interval, all taxis’ status are updated. A customer’s request waiting time increases by 1 minute if the request has not been accepted by any taxi during the previous time step.

The New York City (NYC) taxi trip data from 2013 were used [32] as the input for our simulation. These include a total of 173 million taxi trips or about 450 thousand trips per day on average. For each occupied trip, the timestamp and GPS coordinates of pick-up and drop-off, the travel distance, and the travel time were recorded. The straight-line (Euclidean) distances were computed based on the GPS coordinates and

![Fig. 1. The process of simulating EA V taxi operations.](image-url)
the actual travel distances were used to calibrate (1) [33].

\[ D = 1.4413L + 0.1383(R^2 = 0.9485) \quad (1) \]

where

- \( D \) is the actual travel distance (miles); and
- \( L \) is the straight-line distance calculated based on the GPS coordinates of the trip’s origin and destination (miles).

In addition, the following assumptions were made. First, an expanded charger network with 652 charging stations in NYC as shown in Hu et al. [33] was adopted. We assume 50 kW fast chargers at all stations, as fast and ultrafast chargers are expected to be prevalent in the era of self-driving [34]. Second, the travel distances of nonoccupied trips, which were not provided in the dataset, were estimated based on (1). With the built-in navigation system, we assume EAV taxis will take the shortest path to pick up customers or go to charging stations. Third, we assume taxis wait for new customer requests near the drop-off locations or charging stations. It is possible that taxis are not allowed to park at certain locations for a long time and would need to relocate, but we ignore this cruising distance for simplicity.

III. EAV TAXI DISPATCH MODELS

In this section, we first present our optimization-based dispatch model for EAV taxis. This model maximizes the total reward of matching taxis and customers. Its reward function considers the trade-off between taxi system efficiency and customer equity. Then, we present how this optimization model’s simulation of EAV taxi dispatch operations was used to generate optimal dispatch solutions. Next, our artificial neural network was trained using these solutions to learn optimal dispatch strategies.

A. Optimization-Based Dispatch Model

At a given time step \( T \), there are \( I \) available taxis and \( J \) customer requests. Our optimization-based dispatch model decides which EAV taxi \( i \) should pick up which customer request \( j \) in \( J \). The model’s objective is to maximize the total reward, while serving all the customers. Our optimization model is constrained by customer waiting time, taxi–customer distance, and battery range. The notation is defined as follows.

- \( R_i \) : Remaining range of taxi \( i \), \( \forall i \) (miles).
- \( R_c \) : Remaining range threshold, below which taxis need to charge (miles).
- \( T_{\text{max}} \) : Maximum pickup travel time (min.).
- \( D_i^p \) : Trip distance of customer request \( j \), \( \forall j \) (miles).
- \( D_j^p \) : Distance from the drop-off location of customer request \( j \) to the nearest charging station, \( \forall j \) (miles).
- \( T_{\text{w}}^j \) : Request waiting time of customer \( j \), \( \forall j \), that is, the time that the customer has waited until the request is accepted (min.).
- \( D_{i,j}^p \) : Pickup travel distance between taxi \( i \) and customer \( j \), \( \forall i, j \) (miles).
- \( T_{i,j}^p \) : Pickup travel time between taxi \( i \) and customer \( j \), \( \forall i, j \) (min.), calculated based on the pickup travel distance \( D_{i,j}^p \) and travel speed.
- \( r_{i,j} \) : The reward for taxi \( i \) picking up customer \( j \), \( \forall i, j \).

The EAV taxi dispatch problem is formulated as an integer linear programming (ILP) model. The binary decision variables are \( x_{i,j} \), \( x_{i,j} \) equals 1 when taxi \( i \) is assigned to customer \( j \). The objective function is defined in (2), which maximizes the total reward for picking up all customers in set \( J \).

\[
\begin{align*}
\text{max} & \quad \sum_i \sum_j r_{i,j} x_{i,j} \\
\text{subject to} & \quad \sum_j x_{i,j} \leq 1, \quad \forall i \\
& \quad \sum_i x_{i,j} \leq 1, \quad \forall j \\
& \quad \left( R_i - D_{i,j}^p - D_j^p \right) - \min(D_j^p, R_c) \cdot x_{i,j} \geq 0, \quad \forall i, j \\
& \quad \left(T_{\text{max}} - T_{i,j}^p\right) \cdot x_{i,j} \geq 0, \quad \forall i, j \\
& \quad x_{i,j} \in \{0, 1\}, \quad \forall i, j
\end{align*}
\]

In addition, the reward for dispatching taxi \( i \) to customer \( j \), \( r_{i,j} \), is defined in (8), where \( M \) is a sufficiently large number and \( \eta \) is the weight of the request waiting time \( T_{\text{w}}^j \). \( M \) guarantees reward \( r_{i,j} \) is non-negative, i.e., \( M \geq \max \left(T_{i,j}^p - \eta T_{\text{w}}^j\right) \). Since the maximum pickup travel time is \( T_{\text{max}} \) and customer requests can be accepted without waiting (i.e., \( T_{\text{w}}^j = 0 \)), we have \( \max \left(T_{i,j}^p - \eta T_{\text{w}}^j\right) = T_{\text{max}} \). Let \( M = T_{\text{max}} \). The reward decreases as the pickup travel time \( T_{i,j}^p \) becomes longer, i.e., the taxi system’s efficiency decreases.

The request waiting time \( T_{\text{w}}^j \) acts as the level of emergency for the customer needing to be picked up. Our optimization model gives higher priority to dispatching taxis to the customers who have waited for a longer time, so it is more equitable among customers than many previous optimization models that did not consider request waiting time [19]–[22], [25]; historically, taxi dispatching has not favored customers who have waited a long time, especially during rush hour. In our model, however, the weight \( \eta \) trades off between pickup travel time (or taxi system efficiency) and request waiting time (or customer equity). We explore this trade-off by varying the value of \( \eta \) in Section IV.A. However, let \( \eta = 1 \) if not specified.

\[
r_{i,j} = M - T_{i,j}^p + \eta T_{\text{w}}^j \quad (8)
\]

Our objective of optimal dispatch is subjected to the following constraints. Each taxi will serve at most one customer request, and each customer request will be served by at most one taxi, as shown by the constraint sets (3) and (4) above, respectively. Dispatch of an EAV taxi is also subjected to the taxi’s remaining range. If the remaining range of taxi \( i \) is not enough for picking up customer request \( j \) and dropping off the customer at the destination, taxi \( i \) will not be assigned to that customer. Also, if taxi \( i \) needs charging after drop-off, the remaining range should be enough for traveling to the nearest charging station, as written in constraint set (5) above. Constraint set (6) requires that the pickup travel time should be no more than the maximum pickup time \( T_{\text{max}} \) (assumed as 30 minutes). Constraint set (7) requires the decision variables to be binary.
TABLE I
INPUTS OF AN ANN-BASED DISPATCH MODEL

<table>
<thead>
<tr>
<th>Input variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>The GPS longitude of taxi $i$.</td>
</tr>
<tr>
<td>$X_2$</td>
<td>The GPS latitude of taxi $i$.</td>
</tr>
<tr>
<td>$X_3$</td>
<td>The remaining range of taxi $i$ (miles).</td>
</tr>
<tr>
<td>$X_4$</td>
<td>The GPS longitude of customer request $j$.</td>
</tr>
<tr>
<td>$X_5$</td>
<td>The GPS latitude of customer request $j$.</td>
</tr>
<tr>
<td>$X_6$</td>
<td>The trip distance of customer request $j$ (miles).</td>
</tr>
<tr>
<td>$X_7$</td>
<td>The trip travel time of customer request $j$ (min.).</td>
</tr>
<tr>
<td>$X_8$</td>
<td>The waiting time of customer request $j$ (min.).</td>
</tr>
<tr>
<td>$X_9$</td>
<td>The pickup travel distance from taxi $i$ to customer request $j$ (miles).</td>
</tr>
<tr>
<td>$X_{10}$</td>
<td>The pickup travel time from taxi $i$ to customer request $j$ (min.).</td>
</tr>
<tr>
<td>$X_{11}$</td>
<td>Timestamp, the minutes elapsed relative to the beginning of the day.</td>
</tr>
</tbody>
</table>

B. Artificial-Neural-Network-Based Dispatch Model

The major drawback of any such optimization-based dispatch model is that solving the integer linear program is computationally intensive, especially when a large number of taxis and customer requests are involved. In fact, the optimization dispatch problem can be reformulated as a bipartite matching problem and solved by algorithms with the time complexity of $O(n^3)$ (e.g., Lawler [35], Hung and Rom [36], and Jonker and Volgenant [37]). However, to dispatch EAVs in real time, a fast and accurate dispatch model is desired. Artificial neural networks can take a long time to train, but calculating outputs based on a trained neural network is fast. Therefore, an ANN-based dispatch model is proposed.

When a customer requests a taxi, his/her current location, destination, trip distance, trip travel time, and request waiting time are known. The dispatch system finds all available taxis for this request. For each pair of the request and taxi $(i-j)$, there are 11 input variables to our ANN dispatch model, as listed in Table I. The 11 input variables include the taxi’s status (location and remaining range), the request status (location, trip distance, trip travel time, and waiting time), the spatial and temporal relationship between taxi $i$ and request $j$, and the current time. The neural network calculates the probability of dispatching taxi $i$ to request $j$ ($p_{i,j}$). The ANN consists of one input layer (11 neurons), three hidden layers (128, 64, and 8 neurons, respectively), and one output layer, as shown in Fig. 2. These hyperparameters were selected based on preliminary experiments. Each neuron is connected to all the neurons from the previous layer. Model estimates are computed using forward propagation. The output from a layer is given by

$$ a^{(l+1)} = g^{(l+1)}(W^{(l+1)T}a^{(l)} + b^{(l+1)}) $$  \hspace{1cm} (9)

where $l$ denotes the $l$ th layer in the neural network; $a^{(l+1)}$ is the output vector from layer $l + 1$ and $a^{(1)} = X$, which is the input vector; $W^{(l+1)}$ is the weight matrix associated with the connection between layer $l$ and layer $l + 1$; $b^{(l+1)}$ is the bias vector; and $g^{(l+1)}()$ is the activation function for layer $l+1$.

The three hidden layers are activated by the rectified linear unit (ReLU) function as written in (10). The output layer is activated by the sigmoid function as written in (11). The output is the predicted probability of dispatching taxi $i$ to request $j$, denoted as $p_{i,j}$. If $p_{i,j}$ is less than a threshold (i.e., $p'$), the prediction class is 0. That is to say, the dispatch model will not match the corresponding taxi–request pair. If $p_{i,j} \geq p'$, the prediction class is 1 and the corresponding taxi–request pair is a candidate for matching. In addition, we implemented dropout [38] with the drop rate of 0.2 following the first and second hidden layers to prevent overfitting.

$$ ReLU(x) = \max(0,x) $$  \hspace{1cm} (10)

$$ \sigma(x) = \frac{1}{1 + e^{-x}} $$  \hspace{1cm} (11)

Binary cross-entropy was used as the loss function for this binary classification problem, as shown in (12). The neural network was trained to minimize the loss function.

$$ L(y, p) = -\frac{1}{N} \sum_{k=1}^{N} y_k \log(p_k) + (1 - y_k) \log(1-p_k) $$ \hspace{1cm} (12)

where

- $y_k$ is the true class of the $k$th taxi–customer pair;
- $p_k$ is the predicted probability of dispatching for the $k$th taxi–customer pair; and
- $N$ is the total number of taxi–customer pairs.

The neural network outputs a dispatching probability for each taxi–request pair, so the time complexity is $O(n^2)$. The predicted probability of the neural network cannot alone determine which taxi should be dispatched to which request. As illustrated in Fig. 3, there are overlaps among the taxi–request pairs. For example, the probabilities that taxi 3 is dispatched to request 1, 2, and 3 are 0.3, 0.9, and 0.6, respectively, but taxi 3 can pick up only one customer.

To solve the overlap problem, Algorithm 1 was implemented following the neural network outputs to make the final dispatch decisions. First, find the largest dispatching probability from the outputs. If the largest probability ($p^{max}$) is smaller than $p'$, meaning all the prediction classes are 0, do not dispatch any taxis. If the largest probability is larger than or equal to $p'$, find
For example, Chen has been used in previous work to simulate fleet operations. 5% of the taxis and 5% of served customer requests from model to simulate EA V taxi operations. We randomly drew 23:59, were selected for our optimization-based dispatch optimization-based dispatch model. Three consecutive days C. Training the ANN-Based Dispatch Model in set \( J \) all taxis in set \( J \) have been dispatched or all customer requests that involve taxi \( i \) have been served.

Fig. 3. The overlaps among taxi–request pairs.

\[
\text{Algorithm 1 Dispatch Decisions Following the Neural Network Outputs}
\]

\begin{itemize}
\item \textbf{Input:} available taxis \( I \),
\item \text{customer requests } \( J \),
\item \text{taxi–request pairs } \( i-j \) \( i \in I, j \in J \),
\item \text{probabilities of dispatching } \( p_{ij} \) \( i \in I, j \in J \)
\end{itemize}

\begin{itemize}
\item \textbf{Output:} which taxi \( i \) should be dispatched to pick up which customer request \( j \)
\end{itemize}

\begin{algorithm}
\begin{algorithmic}
\State \textbf{while} \( I \neq \emptyset \) and \( J \neq \emptyset \) \textbf{do}
\State find the largest probability of dispatching \( p_{\max} \)
\If {\( p_{\max} \geq p' \)}
\State Find the corresponding \( i-j \) pair
\State Dispatch taxi \( i \) to request \( j \)
\State Remove all the taxi–request pairs that have \( i \) or \( j \)
\Statex \( I \leftarrow I - i \)
\Statex \( J \leftarrow J - j \)
\Else
\State \textbf{break}
\EndIf
\State \textbf{end while}
\end{algorithmic}
\end{algorithm}

the corresponding pair of taxi \( i \) and request \( j \). Dispatch the taxi \( i \) to pick up the request \( j \). Then remove all the taxi–request pairs that involve taxi \( i \) or request \( j \). Iterate this process until all taxis in set \( I \) have been dispatched or all customer requests in set \( J \) have been served.

C. Training the ANN-Based Dispatch Model

Our ANN-based dispatch model learned the optimal dispatch decisions from our previously described optimization-based dispatch model. Three consecutive days from September 10, 2013, 00:00, to September 12, 2013, 23:59, were selected for our optimization-based dispatch model to simulate EAV taxi operations. We randomly drew 5% of the taxis and 5% of served customer requests from this 3-day dataset to run the simulation. A similar technique has been used in previous work to simulate fleet operations. For example, Chen et al. [10] used 10% of all trip demand in a metropolitan area to simulate a fleet of shared EAVs, and Fagnant et al. [39] simulated 5% shared AV serving 5% of all vehicle trips on a network with 5% capacity.

At each ANN iteration, optimal dispatch decisions were computed and the dispatch data (as listed in Table I) for each taxi–request pair were generated. If taxi \( i \) is dispatched to request \( j \), the dispatch decision is labeled as 1; otherwise, it is labeled as 0.

Since our optimization-based dispatch model does not allow taxis to pick up customers who are far away, we removed the dispatch data in which the taxi-to-customer travel time is beyond \( T_{\max} \), which left about 10 million data samples. 90% of these data were used for training the neural network and 10% for validation. Only 1% of the training data were labeled as 1. We used the up-sampling method to keep a balance of the classes, so the numbers of major and minor classes in the training data were equal.

To evaluate our model estimates, we employed two metrics—accuracy and recall. In (13) and (14), \( TP \) stands for true positive (true class = 1, prediction = 1), \( TN \) stands for true negative (true class = 0, prediction = 0), \( FP \) stands for false positive (true class = 0, prediction = 1), and \( FN \) stands for false negative (true class = 1, prediction = 0). Accuracy is the percentage of the data that are correctly classified. Recall, or sensitivity, is the percentage of the data with the true class of 1 that are predicted as such.

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (13)
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \quad (14)
\]

IV. RESULTS

A. Taxi System Efficiency and Customer Equity

The reward for dispatching a taxi to a customer is calculated using pickup travel time and request waiting time. If request waiting time is not taken into account, the optimization model will minimize total pickup travel time, achieving the highest taxi system efficiency. However, this may cause some customers to wait for a long time or not be served at all. If a higher dispatch priority is given to customers who have waited longer, the optimization model will tend to pick up these customers even if they are further away, thus creating greater equity among customers. There is a trade-off between taxi system efficiency and customer equity. To explore the trade-off between efficiency and equity, we varied the weight \( \eta \) of request waiting time in the reward function.

Taxi system efficiency was represented by the average pickup travel time of taxis. A higher pickup travel time indicates lower system efficiency. Customer equity was represented by the Gini coefficient of the customer’s waiting time before his/her request is accepted. The Gini coefficient is a measure of statistical dispersion (ranging from 0 to 1), which is commonly used for a country’s income inequity [40]. It has also been used to study the equity of transportation systems [41]–[43]. A lower Gini coefficient indicates a more equitable system. We used one computation instance of dispatching as an example and solved the optimization problem with different weights in the reward function. Fig. 4 shows that if request waiting time is not considered (\( \eta = 0 \), the taxi system’s efficiency is highest but customer equity is the worst. As the weight \( \eta \) increases, meaning higher priority is given to
customers who have waited longer, taxis spend more time to pick up customers on average, but the waiting time is more equally distributed.

B. Training Results of an ANN-Based Dispatch Model

The Adaptive Moment Estimation (Adam) optimizer [44] was used to train our neural network, which minimizes the loss function as defined in (12). The number of epochs was 100. The batch size was 256. The training and validation loss are shown in Fig. 5. The validation losses do not show signs of overfitting.

After training and validating for 100 epochs, the validation loss was 0.1888. Different threshold values of $p^i$, ranging from 0 to 1, were tested on the validation dataset. The receiver operating characteristic (ROC) curve is plotted in Fig. 6. Sensitivity plus specificity achieve their maximum values when the threshold equals 0.5 (i.e., $p^i = 0.5$). The validation accuracy is 0.9075. The recall on the validation dataset is 0.93, indicating that our ANN-based model is very unlikely to mistakenly predict a taxi–customer pair that should be matched as not matched.

C. Performance of Dispatch Models

To compare the performance of our ANN-based dispatch model and optimization-based dispatch model, we used trips from October 16, 2013, to simulate EAV taxis dispatched by the two models, respectively. We drew 5% of the taxis (650 taxis) and the corresponding customer requests they served as the samples for this analysis. We generated optimal dispatch data for this day and used them as an independent testing dataset. Applying our ANN dispatch model to this testing dataset, the ANN model’s accuracy was 0.8653 and the recall was 0.92.

Objective value measures how well an ANN model has learned the optimal strategies. For our October 16 one-day simulation, we employed both models to solve the dispatch problem every minute, respectively, and calculated the objective values. Our ANN model always had smaller or equal objective values compared to the optimization model, as shown in Fig. 7. The two methods generated dispatch solutions with equal objective values in 26% of cases, indicating that the ANN model solutions were indeed optimal. Among the other 74% of cases, the objective values of the ANN model were within 20% of the optimal values in 69% of cases, and in only about 5% of cases, the ANN model generated significantly worse solutions. Therefore, our ANN model is capable of generating optimal or near-optimal solutions.

Furthermore, the optimization and the ANN-based dispatch models have similar performance in terms of the number of served customers, serving 99.3% and 99.2% of the customer requests, respectively. For both models, less than 1%
TABLE II
PERFORMANCE OF AN EAV TAXI FLEET UNDER DIFFERENT DISPATCH MODELS

<table>
<thead>
<tr>
<th></th>
<th>Optimization model</th>
<th>ANN model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. occupied trips</td>
<td>36.2</td>
<td>36.1</td>
</tr>
<tr>
<td>Avg. travel distance (miles)</td>
<td>134.9</td>
<td>130.4</td>
</tr>
<tr>
<td>Avg. empty travel distance (miles)</td>
<td>34.1</td>
<td>29.8</td>
</tr>
<tr>
<td>Avg. travel distance occupancy</td>
<td>0.758</td>
<td>0.777</td>
</tr>
<tr>
<td>Avg. time spent charging (min.)</td>
<td>44.7</td>
<td>42.7</td>
</tr>
</tbody>
</table>

Fig. 8. Histograms of computation time for different dispatch models.

of customers cannot be served by an EAV taxi within the allowed waiting time. Using the optimization-based dispatch model, 76% and 92% of customers can be picked up within 5 and 10 minutes, respectively, while this is 83% and 93% using the ANN model. Over 90% of customer requests can be immediately accepted by taxis under both dispatch models. In sum, as shown in Table II, the performance of the EAV taxi fleet under the two models is similar. Therefore, the ANN model has, in fact, learned the optimal dispatch solutions and its performance is similar to the more computation-time-intensive optimization model.

D. Computation Time for Dispatch Models

Our simulation of EAV taxi operations using different dispatch models ran on a workstation with an Intel Xeon E5-1620 CPU and 16GB RAM. The optimization-based dispatch models were solved using Gurobi 8.0.1. We recorded the time spent on obtaining dispatch solutions at each time step during the simulation. Fig. 8 shows the histograms of both models’ computation time. The optimization dispatch model took 0–120 seconds to solve the dispatch problem; the average computation time was 43 seconds. The ANN dispatch model took, on average, only 10 seconds.

For faster computation, the above simulation drew only 5% of samples from the taxi population. For real-world EAV dispatching problems, the size of the slowdown problem could be much larger. Therefore, we compared the average computation time with a 95% confidence interval for different sample sizes, as seen in Fig. 9. When dispatching a large fleet, the average computation time of the optimization model increases more quickly than for the ANN model. In Fig. 10, we show the two models dispatching 100% of samples at different times of the day. This reveals that the ANN dispatch model can make dispatch decisions within a reasonable time frame throughout the day. Thus, the computation time advantage of the ANN dispatch model is more significant for larger-scale problems.

Overall, since the ANN dispatch model provides near-optimal dispatch solutions and is much faster to compute, it is suitable for solving real-time EAV taxi dispatching problems.

E. Improvements in Taxis’ Operational Efficiency

The operational efficiency of the EAV taxis dispatched by our ANN-based model is compared that of the current taxi fleet in terms of total travel distance, empty travel distance, and travel distance occupancy ratio, as shown in Fig. 11, 12, and 13, respectively.

Fig. 11 compares the distributions of the total travel distances of the current taxis and the EAV taxis dispatched by our ANN-based model. A $t$-test was conducted to show that the average travel distance of EAV taxis dispatched by our ANN-based model ($\bar{d}_1 = 130.4$ miles) is significantly shorter than that of current taxis ($\bar{d}_2 = 157.8$ miles). This is mainly due to reduction in unoccupied trip distance, as can be seen in Fig. 12, which compares the distributions of unoccupied travel distances. A $t$-test shows that the average unoccupied travel distance of EAV taxis dispatched by our ANN-based model ($\bar{d}_{1n} = 29.8$ miles) is significantly shorter than that of
TABLE III
LEVEL OF SERVICE AND OPERATIONAL EFFICIENCY OF EAV TAXIS AT DIFFERENT FLEET SIZES

<table>
<thead>
<tr>
<th></th>
<th>650 EAVs</th>
<th>618 EAVs</th>
<th>585 EAVs</th>
<th>553 EAVs</th>
<th>520 EAVs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reduction in fleet size</td>
<td>0%</td>
<td>5%</td>
<td>10%</td>
<td>15%</td>
<td>20%</td>
</tr>
<tr>
<td>Unserved requests</td>
<td>0.8%</td>
<td>0.9%</td>
<td>1.0%</td>
<td>1.3%</td>
<td>2.1%</td>
</tr>
<tr>
<td>Taxi pickup time ≤5 min</td>
<td>83%</td>
<td>82%</td>
<td>82%</td>
<td>81%</td>
<td>82%</td>
</tr>
<tr>
<td>Requests accepted immediately</td>
<td>93%</td>
<td>90%</td>
<td>87%</td>
<td>83%</td>
<td>75%</td>
</tr>
<tr>
<td>Avg. occupied trips</td>
<td>36.1</td>
<td>38.0</td>
<td>40.1</td>
<td>42.3</td>
<td>44.6</td>
</tr>
<tr>
<td>Avg. travel distance (miles)</td>
<td>130.4</td>
<td>137.8</td>
<td>146.2</td>
<td>154.7</td>
<td>163.1</td>
</tr>
<tr>
<td>Avg. empty travel distance (miles)</td>
<td>29.8</td>
<td>32.1</td>
<td>34.9</td>
<td>37.5</td>
<td>39.9</td>
</tr>
<tr>
<td>Reduction in fleet travel distance</td>
<td>17.4%</td>
<td>17.0%</td>
<td>16.6%</td>
<td>16.6%</td>
<td>17.3%</td>
</tr>
<tr>
<td>Avg. time spent charging (min.)</td>
<td>42.7</td>
<td>44.7</td>
<td>48.8</td>
<td>53.1</td>
<td>54.1</td>
</tr>
</tbody>
</table>

![Fig. 11. Histograms of total travel distance for current taxis and EAV taxis.](image1)

![Fig. 12. Histograms of empty travel distance for current taxis and EAV taxis.](image2)

![Fig. 13. Histograms of travel distance occupancy for current taxis and EAV taxis.](image3)

F. Reduction in Fleet Size

EAV taxis have the potential to match current taxi fleet operations with fewer vehicles. We therefore experimented with different EAV taxi fleet sizes dispatched by our ANN model. Each fleet size’s level of service and operational efficiency are compared with the performance of current taxi fleet shown in Table III.

EAV taxis become busier with a smaller fleet size in terms of more occupied trips, longer travel distances, and longer time spent charging. The percentage of unserved requests increases from 0.7% to 2.1% when the fleet size is reduced from 650 EAVs to 520 EAVs. However, over 80% of customers can still be picked up within 5 minutes, though the percentage of immediately accepted requests drops with the reduced fleet size from 93% to 75%.

When the fleet size is reduced by 15% to 553 EAV taxis, each travels 154.7 miles daily on average, which is close to the performance of the current taxi fleet. However, the average empty travel distance drops by 15.1 miles compared to the current taxi fleet, ultimately saving 16.6% of total fleet travel distance. Therefore, EAV taxis can reduce fleet size by 15% while maintaining a comparable level of service and traveling fewer miles. In other words, 1 EAV taxi can replace about 1.2 current taxis.

V. CONCLUSION

This paper studies the EAV taxi dispatching problem. We first designed a simulation framework that can implement different dispatch models to simulate the operations of EAV taxis. Then, we proposed two EAV taxi dispatch models—the optimization-based model, which maximizes the total reward for serving customers and the ANN-based model, which learned the optimal dispatch strategies from the optimization model.
Three consecutive days in 2013 were selected for simulating the operations of the New York City EAV taxis dispatched by our optimization model. The optimal dispatch solutions were generated during this simulation process. Our ANN dispatch model was then trained using these generated data, in order to learn the optimal dispatch strategies.

To compare the performance of the two dispatch models, another day was selected for simulation. The results show that our ANN dispatch model has very close performance to our optimization model. The optimal dispatch solutions were generated during this simulation process. Our ANN dispatch model was then trained using these generated data, in order to learn the optimal dispatch strategies.

EAV taxis dispatched by our ANN model could improve the operational efficiency of current taxi fleets. On average, EAV taxis can reduce travel distance by 17%, reduce empty travel distance by 43%, and increase travel distance occupancy from 67% to 78%, all while serving 99.2% of customer requests. By experimenting with different fleet sizes, it was also found that EAV taxis can reduce fleet size by 15% while maintaining a comparable level of service as the current taxi fleet.

Our trained ANN can handle supply and demand pattern changes at different times of day and from day to day. However, if significant changes in the demand and supply occur during special events, over a long time period, or due to the introduction of disruptive transportation technologies, retraining the model would be necessary. In future research, online learning techniques can be adopted to retrain the model to dynamically adapt to new demand and supply patterns [45]–[46]. The frequency of need for retraining would depend on when new data are available and when the performance of the current model on new data begins deteriorating.

REFERENCES


Authorized licensed use limited to: Iowa State University. Downloaded on May 07,2021 at 23:48:21 UTC from IEEE Xplore. Restrictions apply.


Liang Hu received the B.S. degree in automotive engineering from China Agricultural University in 2013, the M.S. degree in automotive engineering from Tongji University in 2015, and the Ph.D. degree in civil engineering from Iowa State University in 2019. His research interests include the transportation problems of plug-in electric vehicles, autonomous vehicles, ride-sharing, taxi operations, and eco-driving.

Jing Dong (Senior Member, IEEE) received the B.S. degree in automation and M.S. degree in systems engineering from Tsinghua University in 2001 and 2003, respectively, and the Ph.D. degree in civil and environmental engineering from Northwestern University in 2008. She is currently an Associate Professor with the Department of Civil, Construction, and Environmental Engineering, Iowa State University. Her research interests include network modeling and optimization, traffic flow theory, intelligent transportation systems, and transportation energy analysis.