IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 June 2023, accepted 3 July 2023, date of publication 11 July 2023, date of current version 27 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3294345

RESEARCH ARTICLE

Deep Encoder Cross Network for Estimated Time of Arrival

KAI-FUNG CHU[1], (Member, IEEE), ALBERT Y. S. LAM[2,3,4], (Senior Member, IEEE), KA HO TSOI[4], ZHIRAN HUANG[4,5], AND BECKY P. Y. LOO[4,6]

[1]Department of Computing, The Hong Kong Polytechnic University, Hong Kong
[2]Fano Laboratories, Hong Kong
[3]Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong
[4]Department of Geography, The University of Hong Kong, Hong Kong
[5]Department of Building and Real Estate, The Hong Kong Polytechnic University, Hong Kong
[6]School of Geography and Environment, Jiangxi Normal University, Nanchang 330022, China

Corresponding author: Albert Y. S. Lam (ayslam@eee.hku.hk)

ABSTRACT Estimated time of arrival (ETA) is essential to enable various intelligent transportation services and reduce passenger waiting time. Estimating the time of arrival of public transport in a highly dynamic and uncertain transportation system could be challenging. Many indirect factors beyond the remaining travel distance could dramatically deviate the time of arrival from the original schedule. Existing distance-based estimation methods disregarding those factors usually result in inaccurate estimations. In this paper, we propose a new deep learning model, called Deep Encoder Cross Network (DECN), to improve the ETA prediction based on multiple non-distance-based factors such as weather, road speed and congestion, and traffic composition. Unlike most regression tasks that output the target directly, we predict the ETA residual over the location-based ETA prediction. To effectively learn in the large and sparse input feature space, we use a new neural network structure consisting of three main components. First, a deep neural network is responsible for modeling explicit feature interactions. Second, an encoder network is constructed to reduce the input feature dimensionality. Third, a cross-network is introduced to learn from the implicit feature interactions. We conduct extensive experiments on a large real-world bus ETA dataset of Hong Kong, which contains about 2.95×10^8 rows with 27 different features on an 84-dimensional space. The results show that the deep learning approach with the DECN model can improve the ETA error by 11% on average, and 49% for late arrival. The proposed approach can be further improved and extended to estimate other traffic information by incorporating non-distance-based related information.

INDEX TERMS Estimated time of arrival, deep learning, neural network.

I. INTRODUCTION

Traffic information is essential to an intelligent transportation system. Many advanced and next-generation transportation concepts, such as reversible lane [1], optimal vehicle routing [2], traffic signal control [3], etc, can only be realized with the support of a huge amount of information and the corresponding information processing techniques. Sensing [4] and vehicular communication [5] facilitate traffic data collection. Although different kinds of data can be sensed and collected by multiple types of sensors, the data only represent the information of current and past events. The ability to predict future events is crucial to many applications. Predictive techniques, such as traffic flow [6], and travel demand [7] predictions, can be applied to historical and real-time data to get future trends. One traffic information that represents future events is the estimated time of arrival (ETA). ETA usually refers to the arrival time of the next transport to a location. The ACM SIGSPATIAL GIS CUP in 2021 [8] focused on ETA prediction, which revealed the importance of ETA to intelligent transportation systems. For example, traffic congestion was predicted with driving time, indicating

The associate editor coordinating the review of this manuscript and approving it for publication was Tao Huang.

the linkage between traffic congestion and ETA [9]. Bus trajectories can also be inferred from the ETA records [10]. For passengers, ETA indicates how long they have to wait at a specific location [11], or how long they have to wait before arriving at the next location if they are already on board. Revealing the ETA and waiting time to passengers may result in less perceived waiting time [12], positive psychological effects, and higher customer satisfaction [13].

With mature hardware and software technologies, computing the ETA of any location and revealing them on mobile applications is not a difficult task nowadays. The most direct method to predict ETA is to use location data such as global positioning system (GPS) data in [14] and [15]. While GPS may be inaccurate or blocked by tall buildings in densely populated cities, researchers try to incorporate other data, in addition to the GPS-based ETA, to improve the estimation, such as passenger counter [16], fare collection [17], smart card [18], and crowdsourced smartphone [19], etc. However, there are many unpredictable events in the highly dynamic transportation system. Even though many transport operators have already been revealing the ETA, computing an accurate GPS-based ETA is still very challenging. Many studies show that the error between ETA and actual arrival time is associated with many factors. For example, crowdedness and the collective travel behaviors of the travelers affect operating speed, waiting time, travel time reliability, and route and bus choice [20]. Moreover, estimating the time of arrival can be difficult in poor weather conditions [21]. Hence, an accurate ETA system considering those factors is still desired by the public.

To improve estimation accuracy, researchers put effort into information technologies such as data-driven and machine learning approaches. In [22], the authors predicted the bus arrival time using support vector machines (SVM). The segment, current segment travel time, and latest next segment travel time were inputted to the SVM, and the travel time of the preceding/current bus on links was used to estimate traffic conditions of links. In [23], artificial neural networks (ANNs) trained by link-based and stop-based data were integrated with an adaptive algorithm to improve the prediction accuracy. In [24], a hybrid scheme was used to enhance the GPS-based ETA by combining it with the inference results of a neural network based on the decision rules of a Kalman filter. In [25], a hybrid machine learning model was proposed for short-term ETA prediction using a multi-cells neural network-based model. By far, most researchers have focused on predicting ETA *per se* without considering the ETA residual. From the sense of machine learning feasibility, predicting a residual could be much easier than predicting the actual output. For example, a famous deep learning model called ResNet [26] proved the concept of learning just a residual could be much more effective and efficient than learning the actual output. A similar idea can be applied to the ETA prediction using deep learning models. Hence, predicting the ETA residual, instead of ETA *per se*, could be promising research.

In this paper, we aim to obtain an accurate ETA by learning to predict the ETA residual using proxy data without location information. The actual ETA can be determined by adding the ETA residual to the location-based ETA. To project the high dimensional proxy data feature domain into ETA residual domain, we use a new neural network structure called Deep Encoder Cross Network (DECN) that can effectively reduce the input feature dimensionality and learn on the large and sparse feature space, which is suitable for the ETA residual prediction problem in this paper. We conduct extensive experiments on a large real-world bus ETA dataset of Hong Kong, which contains about 2.95×10^8 rows with 27 different features on an 84-dimensional space. Our experimental results show that the ETA residual approach using DECN effectively improves the overall ETA error by 11%, and 49% for late arrival.

The main contributions of this paper are summarized as follows:

1. We propose a deep learning approach to improve ETA prediction accuracy by regulating with ETA residual. The approach predicts ETA residual, instead of the actual ETA value, which is based on proxy features to avoid relying on GPS-based location information;
2. We propose a new neural network structure DECN with three main components: deep neural network, pre-trained encoder network, and cross network. The deep neural network is responsible for modeling the explicit feature interactions. The pre-trained encoder network is constructed to reduce the input feature dimensionality. The cross network is introduced to learn from the implicit feature interactions;
3. We construct a big ETA dataset of Hong Kong containing 2.95×10^8 rows with 27 different features on an 84-dimensional space to evaluate the proposed approach. Results indicate that the ETA error is improved by 11% on average, and 49% for late arrival.

The rest of this paper is organized as follows. Related work is presented in Section II. Section III formulates the ETA system and the ETA residual learning problem. The methodology is discussed in Section IV and experimental results are presented in Section V. Finally, the paper is concluded in Section VI.

II. RELATED WORK

Accurate ETA information is vital to drivers and passengers to reduce their stress and waiting times. Transportation operators can use this information to track and manage the system better [27]. In [28], ETA was predicted by learning from a large number of historical bus GPS trajectories. The authors built a time-dependent graph to model the properties of dynamic road networks, in which the graph represented road segments between two adjacent bus stops and it allowed a clustering approach to estimate the distribution of travel time on the graph. In [29], without relying on GPS data, the system used radio-frequency identification (RFID) to update the latest bus stop that has been passed to provide a

rough knowledge of the real-time bus location between two bus stops. In [30], the authors predicted the ETA of each bus route using the bus running times of multiple routes. Machine learning models, including SVM, ANN, k-nearest neighbors algorithm, and linear regression, were adopted for the prediction. Nevertheless, ETA is influenced by the complex real-time traffic factors, which suggests the potential improvement of incorporating other proxy data in the prediction. [31] enhanced the accuracy by considering patterns of traffic flow and driver travel behavior. To mitigate the effect of dwell time in the ETA prediction, authors of [32] proposed a segment-based approach to predict bus ETA that separately predicts the bus ETA and dwell time with two models by using different impact traffic factors. Reference [33] combined driver information in ETA prediction by constructing a multi-task learning framework that learned an embedding of the personalized driving information. The authors in [34] modeled the traffic links and crosses jointly to learn their spatial-temporal dependencies in the route, which integrated into the Neural Factorization Machines for memorizing the historical patterns and a multiple layer perceptron for integrating various heterogeneous influencing factors. We can see that many research works use proxy data to improve the location-based ETA and this is in line with the current trends of big data research.

The big data cannot be used effectively unless we have a powerful data mining technique such as deep learning. Deep learning has been applied in intelligent transportation systems dramatically in recent years. For example, traffic flow can be accurately predicted by a well-trained deep neural network [6], [35]. A novel deep learning model, called Multi-Scale Convolutional LSTM Network, was proposed to predict travel demand and origin-destination flows by considering temporal and spatial correlations and high-level prediction results of the historical traffic data [36]. Wheel defects can be detected by machine learning method using the measured wheel vertical force [37]. Knowing the current and past observations of the surrounding environment is important to autonomous driving applications, which motivates vehicle behavior prediction [38]. For computer vision tasks, convolutional neural network is a commonly used neural network structure to capture vision patterns from images. Reference [39] detected lane boundaries in traffic scenes with the help of a deep convolutional network. Traffic signs can also be recognized by a deep convolutional network for advanced driver-assistance and autonomous systems as in [40]. For autonomous driving, deep reinforcement learning [41] is an outstanding technology to address the computational challenges in the real-world deployment of autonomous driving agents.

Among the deep learning research works, a groundbreaking neural network architecture, called residual networks (ResNet) [26], was proposed with extraordinary performance on image recognition task, which won the ImageNet Large Scale Visual Recognition Challenge [42] in 2015.

TABLE 1. Notation summary.

Notations	Meaning
$\mathcal{R}$	Set of routes
r	Route $r \in \mathcal{R}$
N_r	Number of stop on route r
$\mathcal{S}$	Set of stops
s	Stop $s \in \mathcal{S}$
M_s	Number of route passing stop s
τ	Timestamp
t_{rs}^{τ}	ETA ground truth of route r, stop s and time τ
$\hat{t}_{rs}^{\tau}$	Predicted ETA of route r, stop s and time τ
$\tilde{t}_{rs}^{\tau}$	Location-based ETA of route r, stop s and time τ
ε_{rs}^{τ}	ETA residual ground truth of route r, stop s and time τ
$\hat{\varepsilon}_{rs}^{\tau}$	Predicted ETA residual of route r, stop s and time τ
$\mathcal{D}_{\text{proxy}}$	Proxy dataset
$\mathcal{D}_{\text{ETA}}$	ETA dataset
$\mathcal{D}_{\text{truth}}$	ETA dataset with ground truth and residual
x_i^{proxy}	Proxy data $x_i^{\text{proxy}} \in \mathcal{D}_{\text{proxy}}$ of feature i
L_p	p-norm loss function

The concept of ResNet is developed based on deep residual learning, which learns a residual function $F(x_i) = x_{i+1} - x_i$ between layers instead of learning the whole target output $F(x_i) = x_{i+1}$ directly, where F and x_i is the forward function of layers and input of layer i, respectively. Since then, deep residual learning and its variants have been widely applied. For example, the authors in [43] used deep residual learning to speed up the training process and boost the performance of image denoising. In [44], the image super-resolution problem was addressed with better performance by a deep residual learning-based approach. Deep residual learning can also be used in the magnetic resonance image (MRI) acquisition domain as presented in [45]. Moreover, MRI scans are applicable to predict the progression to Alzheimer's disease using deep residual learning [46]. Deep residual learning can also address other problems beyond computer vision-based tasks. For example, deep residual learning is applied to wireless communication and networking studies to address clustering and beamforming [47] and channel estimation problems [48]. For transportation, deep residual learning can detect traffic congestion in traffic videos [49]. All the above examples reveal that deep residual learning is a popular and promising data processing technique extendable to different fields of study.

III. PROBLEM FORMULATION

In this section, we define the public transport ETA residual prediction problem. The notation used thereafter is summarized in Table 1.

Consider a public transport network with a set of pre-defined routes $\mathcal{R}$ and a set of stop $\mathcal{S}$. We aim to obtain an accurate public transport estimated time of arrival $\hat{t}_{rs}^{\tau}$ for every stop $s \in \mathcal{S}$ and route $r \in \mathcal{R}$ at time τ. The ETA $\hat{t}_{rs}^{\tau}$

refers to the time when a transport service arrives[1] at stop s serving for route r. Each route r consists of N_r stops and each stop s may serve for M_s routes.

Let t_{rs}^{τ} and $\tilde{t}_{rs}^{\tau}$ be the ground truth of arrival time and ETA, respectively, in which the ETA is given by the public transport service provider based on location sensor such as GPS installed on each public transport. The difference between t_{rs}^{τ} and $\tilde{t}_{rs}^{\tau}$ is defined as ε_{rs}^{τ}, which is dynamic residual based on real-time road situation. Our objective is make a prediction on the residual term represented by $\hat{\varepsilon}_{rs}^{\tau}$ based on a set of given proxy data $\mathcal{D}_{\text{proxy}}$ so as to minimize the L_p-norm between t_{rs}^{τ} and $\tilde{t}_{rs}^{\tau}$:

$$L_p = ||t_{rs}^{\tau} - (\tilde{t}_{rs}^{\tau} + \hat{\varepsilon}_{rs}^{\tau})||_p, \quad \forall s \in \mathcal{S}, r \in \mathcal{R} \tag{1}$$

where $t_{rs}^{\tau} = \tilde{t}_{rs}^{\tau} + \varepsilon_{rs}^{\tau}$, and $||\cdot||_p$ is the norm given by the pth root of the sum of the p-powers of the absolute value. The proxy data $\mathcal{D}^{\text{proxy}}$ represents the road factors influencing the ETA residual such as traffic flow, weather, time of day, etc. GPS data is not included in $\mathcal{D}_{\text{proxy}}$ as it has already been involved to calculate $\tilde{t}_{rs}^{\tau}$. We do not restrict the data type and class here to preserve the generality of problem as long as location information is not included by our definition. Further discussion of the proxy data used in this work will be given in Section V.

To predict $\hat{\varepsilon}_{rs}^{\tau}$, a modeling function f with trainable parameters θ is used to project the high dimensional input proxy data $\mathcal{D}_{\text{proxy}}$ to the residual domain such that

$$\hat{\varepsilon}_{rs}^{\tau} = f(x_i^{\text{proxy}}, \theta, r, s, \tau), \tag{2}$$

where $x_i^{\text{proxy}} \in \mathcal{D}$. $f(x_i^{\text{proxy}}, \theta, r, s, \tau)$ can be any function such as linear regression, support vector regression, decision tree, random forest, deep neural networks, etc.

Formally, the public transport estimated time of arrival rectification problem is defined as:

Problem 1 (Public Transport Estimated Time of Arrival Residual Prediction Problem):

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & \sum_{s\in\mathcal{S}, r\in\mathcal{R}, \tau} \|t_{rs}^{\tau} - \tilde{t}_{rs}^{\tau} - \hat{\varepsilon}_{rs}^{\tau}\| \\ \text{subject to} \quad & x_i^{\text{proxy}} \in \mathcal{D}_{\text{proxy}}. \end{aligned}$$

Fig. 1 shows the system architecture of the ETA residual prediction.

IV. METHODOLOGY

A. OVERVIEW

The transportation system is a highly dynamic system and the proxy data domain for the prediction can be large and sparse. Predicting ETA residual with sparse proxy data would be a challenging task using traditional data mining methods. Therefore, we propose a new deep learning model, i.e., DECN, which can effectively reduce the input feature dimensionality and learn in the large and sparse feature space.

[1]Here we use the general term "time of arrival" as it is widely used by the public transport service provider to indicate the time of the next service. For the service departing at the first stop, the time of arrival refers to the departure time of the transportation service as the vehicles usually arrive and wait to depart. Without loss of generality, we assume the dwell time is zero so that the departure time at each stop is equivalent to the arrival time.

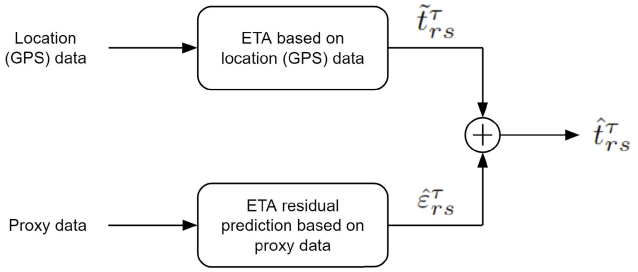


FIGURE 1. System architecture of the ETA residual prediction.

B. DEEP ENCODER CROSS NETWORKS

Traditionally, feature engineering is essential to the accuracy of a prediction model. However, there is no clear guideline for this process and it may require a tremendous amount of manual work and exhaustive searching. The aim of the cross network is to avoid task-specific feature engineering prior to the prediction model training. A pre-trained encoder network can project the input feature space to a lower dimensional space without significant information loss so that only the major combined features are considered by the network model. Together with a deep neural network that has the model capability to capture complex relationship across the input features and inference target, DECN can capture feature interactions and achieve outstanding performance.

DECN is composed of five components, namely an embedding and stacking layer, an encoder network, a deep neural network, a cross network, and an output layer. The input data is first fed to the embedding and stacking layer and encoder network for dimension reduction. Then, the hidden output of the embedding and stacking layer is fed to the cross network and deep neural network in parallel for feature crossing and extraction. Finally, the outputs of the the deep neural network, encoder network, and cross network are stacked to go through the output layer. The overall architecture of DECN is shown in Fig. 2.

1) EMBEDDING AND STACKING LAYER

The input proxy data of the ETA residual problem contain sparse and dense features. For example, the categorical features, such as district and day of the week, are encoded as one-hot vectors, and thus these data are very sparse. On the other hand, numeric features, such as traffic flow, are represented by their own (normalized) values, which are dense. Therefore, an embedding is used to transform the sparse features into dense embedding vectors given as

$$x_i^{\text{embedding}} = w_i^{\text{embedding}} x_i^{\text{sparse}}, \tag{3}$$

where x_i^{sparse} and $x_i^{\text{embedding}}$ are the sparse input and embedding vectors of the i-th categorical features, respectively,

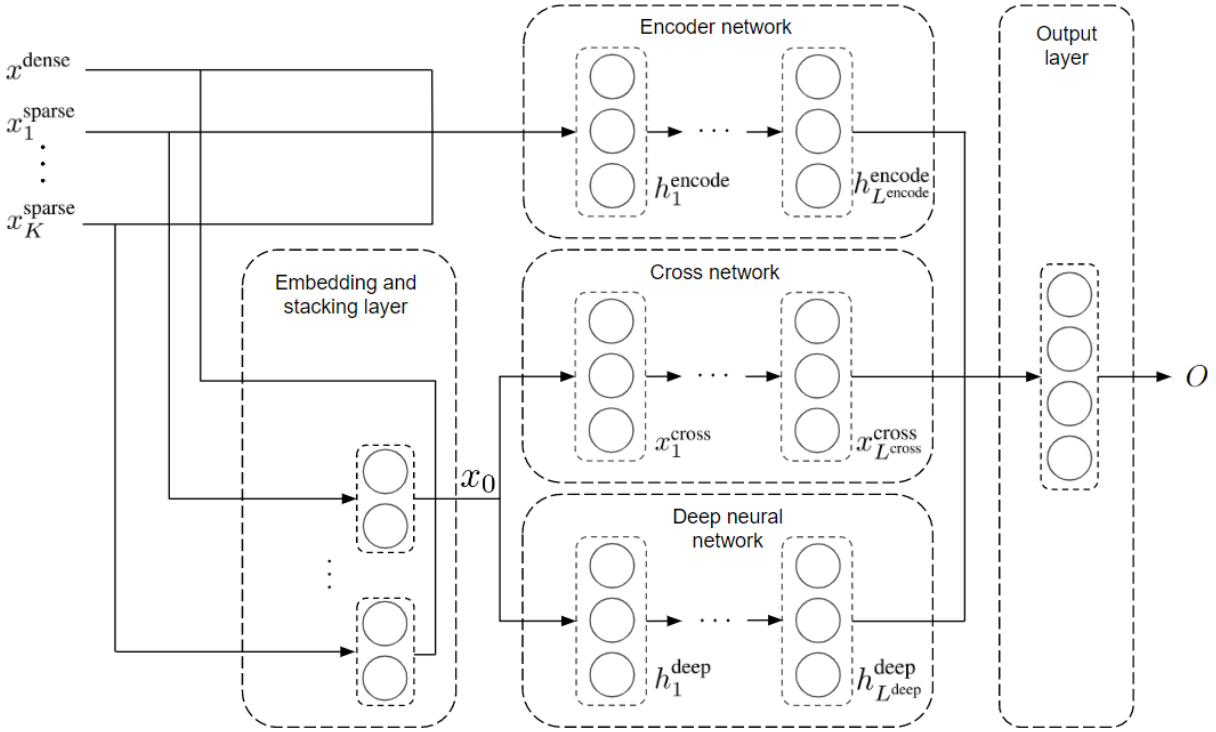


FIGURE 2. DECN architecture.

and $w_i^{\text{embedding}}$ is the trainable embedding parameters to be adjusted during model training.

The output x_0 of the embedding and stacking layer is the concatenation of embedding vectors and the dense vector for the numeric features, and it is specified by

$$x_0 = \left[x_1^{\text{embedding}}; \ldots; x_K^{\text{embedding}}; x^{\text{dense}}\right] \tag{4}$$

for input data with K categorical features, where x^{dense} is the dense vector for the numeric features.

2) ENCODER NETWORK

An encoder is used to reduce the dimensionality of both sparse and dense inputs, which can be represented by

$$h_{l+1} = \phi_l^{\text{encode}}(w_l^{\text{encode}} h_l + b_l^{\text{encode}}), \tag{5}$$

where h_l is the hidden state of the l-th hidden layer, w_l^{encode} and b_l^{encode} are the trainable weight and bias of the l-th hidden layer, respectively, and ϕ_l^{encode} is the activation function of l-th hidden layer.

The encoder network is extracted from an autoencoder pre-trained using the input data of the training dataset. By training the autoencoder to reconstruct the input data as accurately as possible, the encoder can learn to extract meaningful features from the input data and reduce its dimensionality.

3) DEEP NEURAL NETWORK

The deep neural network is a classic multi-layer perceptron and each hidden layer is represented by:

$$h_{l+1} = \phi_l^{\text{deep}}(w_l^{\text{deep}} h_l + b_l^{\text{deep}}), \tag{6}$$

where h_l is the hidden state of the l-th hidden layer, w_l^{deep} and b_l^{deep} are the trainable weight and bias of the l-th hidden layer, respectively, and ϕ^{deep} is the activation function of l-th hidden layer. One may note that the formulas of the Encoder and Deep Neural Network are the same. However, there are two main differences. The first difference is that their hyperparameters can be different. The second difference is that their training mechanisms are different. The Encoder Network is a pre-trained network extracted from an autoencoder pre-trained using the input data of the training dataset. The weight of neurons will be fixed during the training. For the Deep Neural Network, the weight of neurons was randomly initialized and updated during the training.

4) CROSS NETWORK

The cross network is composed of cross layers, which perform crossing among the stacked vector x_0. Each cross layer can be represented by

$$x_{l+1} = x_0 x_l^T w_l^{\text{cross}} + b_l^{\text{cross}} + x_l = f_c(x_l, w_l^{\text{cross}}, b_l^{\text{cross}}) + x_l, \tag{7}$$

where x_l are the intermediate vector outputted from the l-th cross layer, and w_l^{cross} and b_l^{cross} are the trainable weight and

bias of the l-th cross layer, respectively. There are two main differences between the cross layer and the fully connected layer. The first difference is the feature crossing with x_0, which allows the network to discover high order interaction and association across features. The second is that each layer adds back its input, which imitates the network structure of ResNet [26], to facilitate the training process.

5) OUTPUT LAYER

The output layer combines the intermediate outputs from the cross network and deep neural network. The intermediate outputs are stacked and fed to the final layer as the final output of the DECN as:

$$O = \phi^{\text{out}}(w^{\text{out}}[h^{\text{encode}}_{L^{\text{encode}}}; x^{\text{cross}}_{L^{\text{cross}}}; h^{\text{deep}}_{L^{\text{deep}}}] + b^{\text{out}}), \tag{8}$$

where w^{out} and b^{out} are the trainable weight and bias of the output layer, respectively, L^{encode}, L^{cross} and L^{deep} are the number of encoder, cross and deep neural network hidden layers, respectively, and ϕ^{out} is the activation function of the output layer.

To train the model, we use mean squared error (MSE) as the loss function given by:

$$\text{MSE}(t^{\tau}_{rs}, \hat{t^{\tau}_{rs}}) = \frac{1}{B_N}\sum_{i=1}^{B_N}(t^{\tau}_{rs} - \hat{t^{\tau}_{rs}})^2 \tag{9}$$

where B_N is the mini-batch size.

V. EXPERIMENTS

A. DATA COLLECTION AND PRE-PROCESSING

We perform experiments with real-world ETA data of bus services in Hong Kong to evaluate our proposed approach. The ETA data can be accessed and downloaded from the Transport Department,[2] which was provided by The Kowloon Motor Bus Company (1933) Limited. It provides real-time estimated next bus arrival times of all routes and bus stops, including detailed information such as bus stop name, stop location, and generated timestamp. It can be represented by the 4-tuple $\langle r, s, \tilde{t}^{\tau}_{rs}, \tau\rangle$, where r is the bus route, s is the stop index of the bus route, $\tilde{t}^{\tau}_{rs}$ is the GPS-based ETA in absolute time unit computed by the provider, and τ is the timestamp of the record. Fig. 3 shows an example ETA of a bus route and bus stops at timestamp equals 15:01:00. There are 506 bus routes in the dataset. To simplify the formulation, a bus route with opposite directions constitutes two different routes. So there are 1012 bus routes in the dataset. The dataset was streamed every minute for all bus routes and stops during the period of 21 July 2021 to 21 September 2021, which contains 2.95×10^8 ETA records. The distribution of bus stops across the region of Hong Kong is shown in Fig. 4. Most of the bus stops are located in the urban regions with a higher population in Hong Kong, which is an excellent testbed for studying the effect in an urban city. Most regions with fewer bus stops are non-residential and non-commercial areas such as mountains and beaches with low population.

[2]https://data.gov.hk/en-data/dataset/hk-td-tis_21-etakmb

Moreover, we supplement various additional features to the dataset for the prediction task. 27 features in total were added based on the bus route, stop and timestamp. The features are: 1) district, 2) land use, 3) day of week, 4) weather type, 5) total rainfall, 6) route type, 7) sequence of stop, 8) change of road category, 9) peak hour, 10) total route distance, 11) travel distance from the previous stop, 12) wind speed, 13) maximum rainfall, 14) traffic speed, 15) congestion index, 16) bus occupancy, 17) number of routes at a stop, 18) annual average daily traffic 2019, 19) annual average daily traffic 2020, 20) hourly flow, and 21-27) vehicle mix (including shares of motorcycles, cars, taxis, light buses, light duty vans, med high duty trucks, and buses).

For the traffic-related factors, the annual average daily traffic, hourly traffic flow, hourly traffic composition and bus occupancy rates were directly extracted from the Annual Traffic Census conducted by the Transport Department [50]. Real-time hourly traffic speed at each road segment were extracted from TomTom API based on millions of consumer GPS tracking records [51]. Then, the congestion level is calculated with reference to the real-time speed and speed limit following the method proposed by Loo and Huang. Peak-hours are defined as 8am - 10am (morning peak) and 5pm - 7pm (evening peak) [52]. Then, spatial datasets of bus routes and bus stops provided by the bus company as well as the road network (available at data.gov.hk) were used to derive the bus-specific factors, including route distance, types of bus routes, sequence of bus stop, travel distance from the previous stop, whether there is a change of road functional class between stops, and the number of bus routes at a stop. Moreover, administrative boundaries were used to derive the districts and land uses where a bus stop is located. For weather-related variables, the general weather conditions and wind speed were extracted from. The district-based hourly rainfall was compiled by the Hong Kong Observatory.

The GPS-based ETA $\tilde{t}^{\tau}_{rs}$ in the dataset is not the same with the actual bus arrival time t^{τ}_{rs}, as the actual arrival time may be affected by many factors such as traffic congestion and weather instead of the location only. Hence, we need to first determine the actual arrival time from the dataset. The ETA $\tilde{t}^{\tau}_{rs}$ is similar to a staircase function which changes after a bus leaves the bus stop. The time period within the sudden changes usually represents the same next bus. Intuitively, the last record before the change is the most updated and accurate information we can get from the dataset, as the distance between the bus and bus stop is the smallest. Therefore, we assume that the last ETA at each stop for each bus to be the ground truth of arrival time.

To determine whether a record is the ground truth, we detect the sudden changes based on the following three conditions:

- whether the difference between each arrival time is smaller than a threshold Eq. (10);
- whether the difference between the arrival time and generated timestamp is smaller than a threshold Eq. (11); and

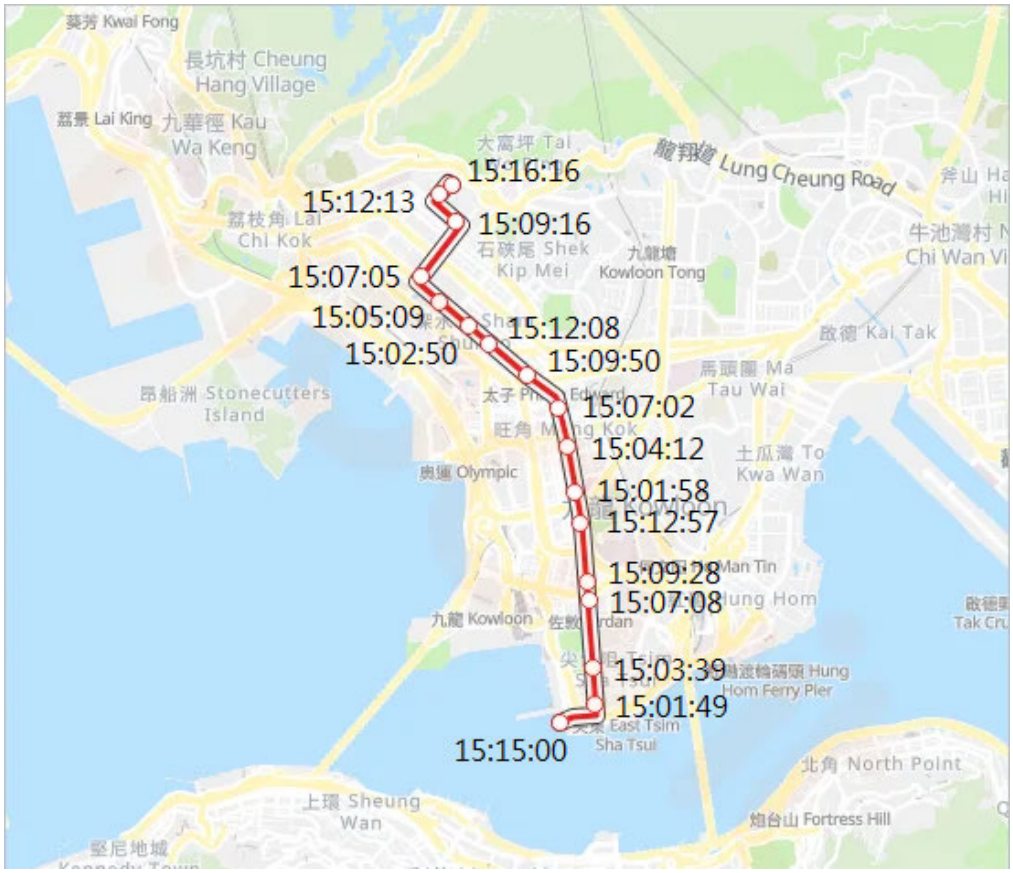


FIGURE 3. An example of ETA of a bus route and bus stops at 15:01:00. The buses move from south to north.

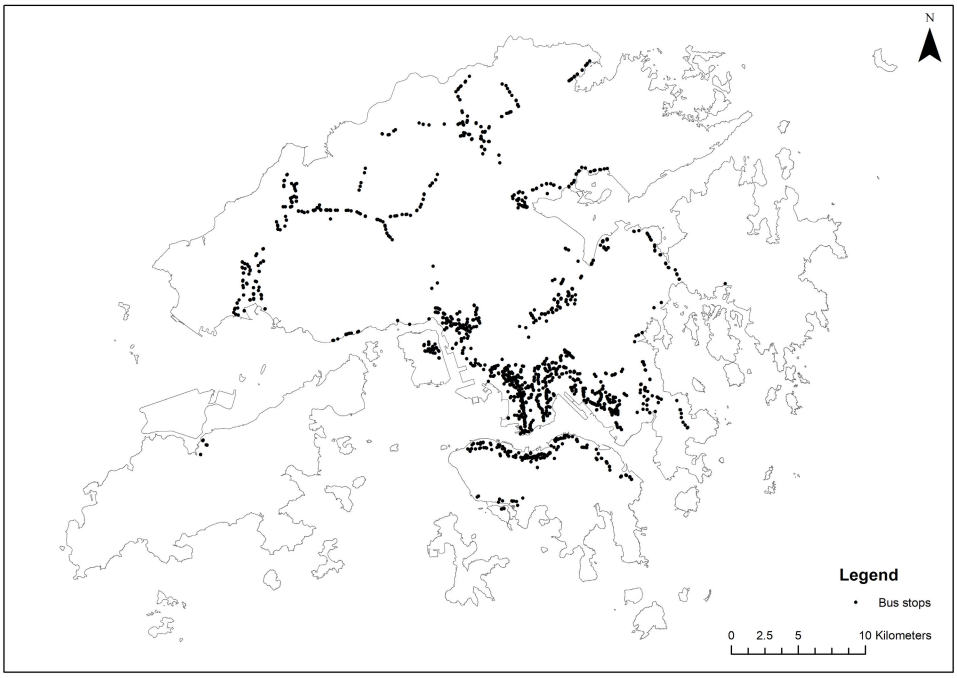


FIGURE 4. Distribution of bus stops.

- whether the difference of the difference between each arrival time is larger than a threshold Eq. (12).

The conditions are:

$$\left|\tilde{t}_{rs}^{\tau} - \tilde{t}_{rs}^{\tau+1}\right| \leq \delta_1 \tag{10}$$

$$\left|\tilde{t}_{rs}^{\tau} - \tau\right| \leq \delta_2 \tag{11}$$

$$\left|\left|\tilde{t}_{rs}^{\tau-1} - \tilde{t}_{rs}^{\tau}\right| - \left|\tilde{t}_{rs}^{\tau} - \tilde{t}_{rs}^{\tau+1}\right|\right| \geq \delta_3 \tag{12}$$

We define the ETA residual as:

$$\varepsilon_{rs}^{\tau} = t_{rs}^{\tau} - \tilde{t}_{rs}^{\tau}. \tag{13}$$

Algorithm 1 shows the algorithm to determine the ground truth and bus ETA residual.

Algorithm 1 Bus Arrival Time Error Analysis

Input: ETA dataset $\mathcal{D}_{ETA}$
Output: ETA dataset with ground truth and residual $\mathcal{D}_{truth}$

```
Sort D_ETA based on r, s, τ
for each r ∈ R and s ∈ S do
    for each d ∈ D_ETA in descending order do
        if Eqs. (10), (11), and (12) are satisfied then
            t_rs := t̃_rs^τ
        end if
        ε^τ = t_rs − t̃_rs^τ
        d := [d, t_rs, ε^τ]
    end for
end for
D_truth := D_ETA
return D_truth
```

After determining the ground truth, we prepare the dataset for model training. We first transform all categorical features into one-hot vectors. Then, we randomly split the dataset by the ETA records (rows) into training, validation, and testing sets in the ratio of 0.64 : 0.16 : 0.2. To be more specific, we use the training set to train the model and the validation set is used to validate the error for better parameters update during training. The testing set is untouched until the model training completed, which produces the resulting testing errors. The ratio is just a rule of thumb in machine learning that splits the training and testing sets into an 8:2 ratio, and the same ratio for training and validation sets, which results in $0.8 \times 0.8 : 0.8 \times 0.2 : 0.2$. Some may also use a ratio of 7:1:2 for simplicity. In general, the lower the training-to-testing ratio, the harder the model to be generalized. Therefore, we try not to take advantage of a larger training-to-testing ratio and set it to 0.64: 0.16: 0.2. Finally, we fit a min-max scaler using only the training set to scale all non-categorical input features to the range between zero to one. The non-categorical input features of the validation and testing set are scaled using the same min-max scaler. Note that the scaling is performed after the split as the testing set should not be included in the feature normalization process, which is a common practice to handle a dataset.

B. BASELINE MODELS AND SETTINGS

We use the following models as baseline models to evaluate the efficiency of DCN:

- Raw: The distance-based ETA alone without regulation by ETA residual.
- Historical Average (HA): The historical average of all ETA residuals is used to be the prediction value.
- Linear Regression (LR): A set of input independent variable x_i and the set of output dependent variable y are used to fit a linear equation $y = \beta_i x_i + c$ with the smallest error.
- K-nearest Neighbors (k-NN) Regressor: KNN is a supervised machine learning algorithm for the categorical target variable. It can be used for regression tasks in which the target variable is numeric.
- Random Forest (RF): An ensemble learning method for regression by combining multiple decision trees.
- Fully-Connected Neural Network (FCNN): An universal model inspired by biological neural networks that perform non-linear mapping from input to output by the weights and biases of the neural network.

We chose the methods for comparison because they are well-known and standard machine learning models, which can be good benchmarks for the prediction task.

We use two common metrics, root mean square error (RMSE) and mean absolute error (MAE), to evaluate the models, which are computed as follows:

$$\text{RMSE}(y, \hat{y}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y - \hat{y})^2} \tag{14}$$

$$\text{MAE}(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N}|y - \hat{y}|, \tag{15}$$

where y_i and $\hat{y}_i$ are the ground truth and predicted values, respectively, and N is the number of samples. Since our goal is to compare the computed values of the prediction model with the raw distance-based ETA which also contains error per se. We define a benchmark metric, error difference, to indicate the difference between the computed values of compared model and the raw distance-based model as follows:

$$E_{diff} = E_{raw} - E_i, \tag{16}$$

where E_i is the RMSE or MAE of model i.

Table 2 summarized the parameters of the models used in the experiments. We run experiments with different parameters and the current settings shown in Table 2 give the best performance.

C. EXPERIMENT RESULTS

1) COMPARISON WITH BASELINE MODELS

We test the ETA improvement with the ETA residual predicted by the baseline models. Figs. 5 and 6 show the normalized ETA improvement compare to raw ETA using

TABLE 2. Parameter settings.

Model	Parameter	Value
k-NN	Number of neighbors	5
	Weight function used	Uniform weights
	Leaf size	30
Random Forest	Number of trees	100
FCNN	Number of neural network layers	512
	Number of neurons of each layer	8
	Batch size	512
	Learning rate	0.0001
	Number of training epochs	200
DECN	Number of neural network layers	8
	Embedding size	32
	Number of encoder layers	4
	Number of cross layers	8
	Number of neurons of each layer	512
	Batch size	512
	Learning rate	0.0001
	Number of training epochs	200

different models for training and testing set, respectively. The ETA improvement value is calculated by one minus the normalized error value of Raw for better visualization. Since the value shown is the one minus percentage error, the higher the value, the better the performance. In general, regulation with ETA residual may not always be beneficial to the ETA error. We can see that there are values lower than zero, which means regulating the ETA with residual decreases accuracy. In other words, a poorly predicted ETA residual could increase the overall ETA error. We can see how the approaches fit the models from the training results. In Fig. 5, HA increases both the RMSE and MAE caused by the negative effect from the average ETA residual. For LR, the performances are similar to Raw. An interesting observation from LR is that the RMSE is above zero while MAE is below zero. That means LR has both positive and negative effects if we consider RMSE and MAE, respectively. This result is mainly due to the mathematical square of the error gap between the samples. This can be interpreted as LR increasing the average absolute error, but it reduces the gap between the samples, especially for the samples with large errors. k-NN has improvement in both RMSE and MAE, which means the model fitting seems effective. RF has the highest training accuracy as it is an ensemble machine learning method that can fit the training data well with the large number of decision trees. The training accuracy of FCNN and DECN are not as good as RF since the neurons are limited to such a large dataset. As expected, DECN can model the training dataset better than FCNN with the additional cross network for sparse feature learning. In theory, we can train the neural networks to near zero error given a neural network with enough model capacity. However, the training results only show the ability to fit the model with the training dataset. We need to consider the generalization ability for the unseen sample, i.e., the testing set error. Considering the error of validation set during the training, we have set the number of neurons to an appropriate value by experiments to avoid over-fitting. Therefore, the training error of FCNN and DECN are withheld to the levels shown in Fig. 5.

Fig. 6 shows the testing set accuracy of different models. The generalization ability of the models can be seen from the results. Similar to Fig. 5, the accuracy is calculated by one minus the normalized error. HA has a negative impact on the ETA accuracy. The testing accuracy, based both on RMSE and MAE, is similar to the training errors, which suggests the average ETA residual calculated from the training set has a similar effect on the testing set, and thus the sets are independent and identically distributed. LR also has similar testing accuracy compared to the training errors. For k-NN, the change in testing accuracy is below zero, which means that it is poorly generalized on new samples, and it is not a good model for this problem. The improvement of RF indicates a much more serious generalization issue as the testing improvement reduces dramatically compared to the training results. FCNN is slightly better than the mentioned models, and DECN is the best among all the models. Although the testing improvement is slightly higher than the training set, it is a normal consequence of having slightly lower training errors in machine learning models. The results suggest that DECN can improve the ETA accuracy by 11% on average. Fig. 7 shows samples of the target and corresponding prediction values. Considering the two results, it can be seen that predicting the ETA residual from proxy data is not a trivial task. Many well-known machine learning models fail to model or generalize the ETA residual. Nevertheless, by considering the nature of data, we discover an effective deep learning model – DECN for the ETA residual prediction via the experiments. These findings will help us to better understand the model effectiveness and identify the method to improve the ETA accuracy.

2) COMPARISON OF DEEP LEARNING MODELS

We study the training, validation, and testing results of the two neural network models, FCNN and DECN, to understand the model training process. Figs. 8 and 9 present the learning curves of FCNN and DECN. The training and validation RMSE and MAE per epoch of the methods are plotted in the figures. The errors are normalized by the Raw errors. The training errors are similar for both methods and decrease smoothly along with the training epoch. However, the validation errors are quite different. The validation errors of DECN decrease along with the training epoch, while the validation errors of FCNN increase significantly at the beginning. This indicates that the FCNN is seriously overfitted and cannot generalize the prediction from the training data. Note that the testing results are performed by the model instance with the lowest validation error, not the one after the last training epoch. Although FCNN has the same structure as the deep neural network part of DECN, it cannot learn from the data

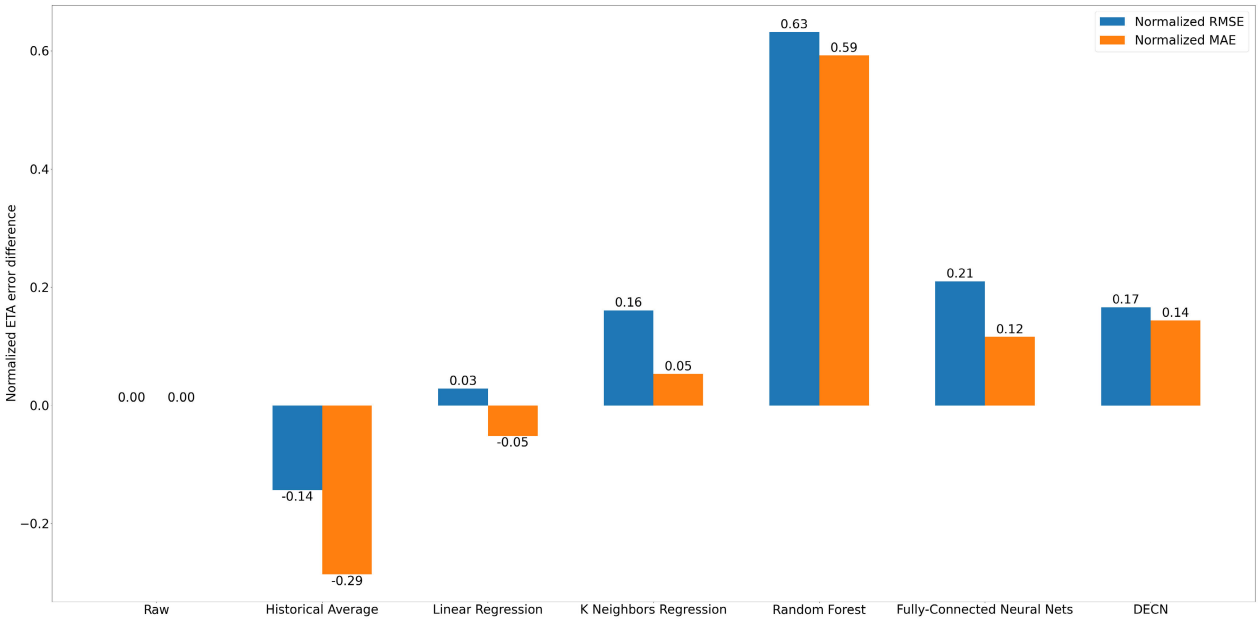


FIGURE 5. Training set normalized ETA error of different models.

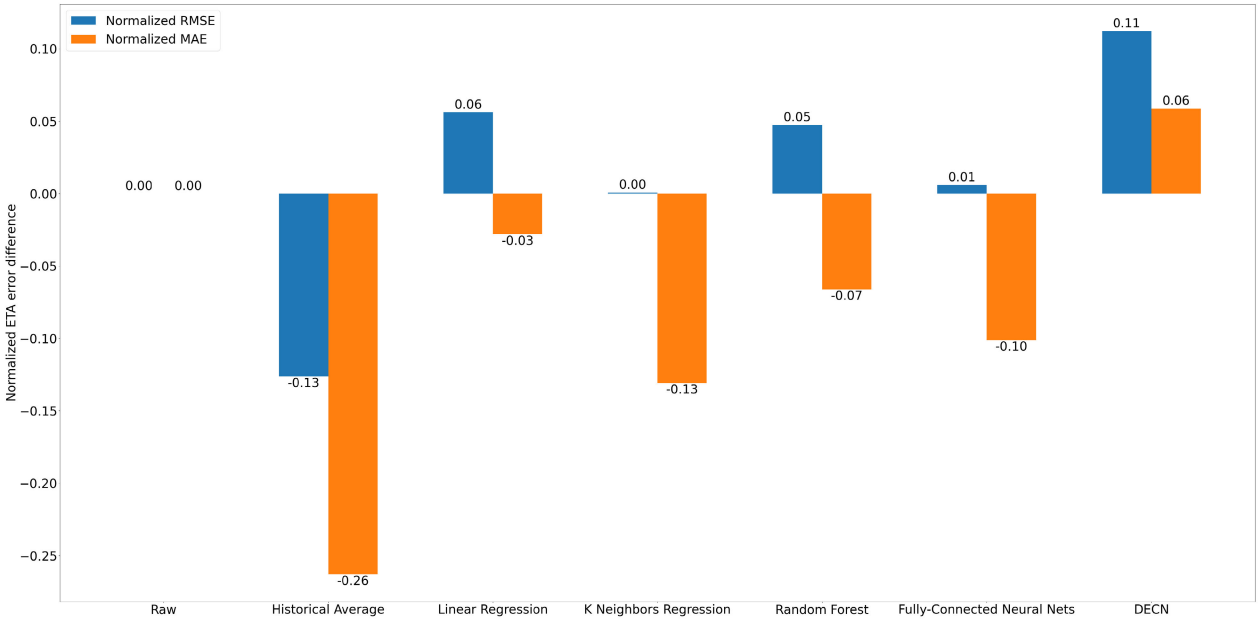


FIGURE 6. Testing set normalized ETA error of different models.

without the help of cross layer. The results suggest that the cross layer facilitates the learning from the sparse feature.

3) COMPARISON OF EARLY AND LATE ARRIVAL

We study the performance of predicting early and late ETA by dividing the dataset based on the ETA error. We use two identical models to predict the early and late ETA separately. Fig. 10 shows the prediction improvement of both early and late prediction. Interestingly, there is a great difference between the prediction improvement of early and late arrival. The prediction of late arrival is much more accurate compared to that of early arrival. This may indicate that the late arrival is much easier to learn compared to early arrival using the proxy data. The reason for this phenomenon could be due to the fact that most of the traffic conditions, such as traffic congestion and bad weather, are delaying vehicles rather than

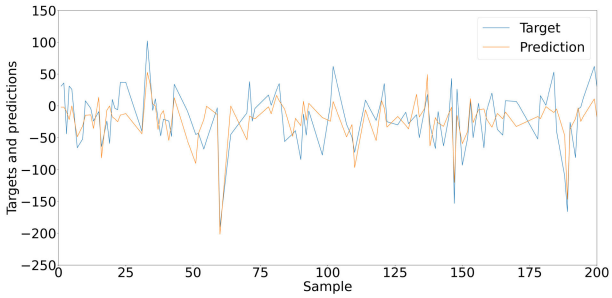


FIGURE 7. 200 samples of target and the corresponding prediction value.

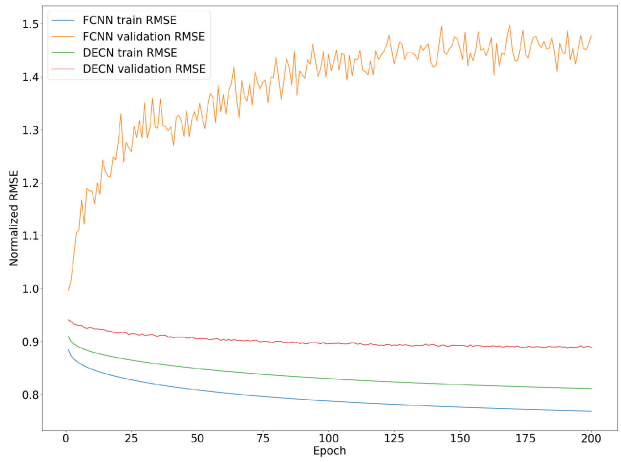


FIGURE 8. Learning curve (RMSE) of the FCNN and DECN.

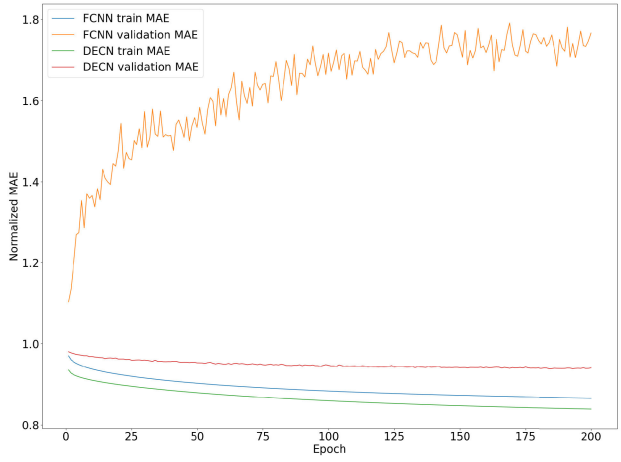


FIGURE 9. Learning curve (MAE) of the FCNN and DECN.

advancing them. In other words, the information extracted from the proxy data mostly signifies the late arrival rather than the early arrival. For example, the vehicle may be late in a bad traffic condition such as in traffic congestion or bad weather even though the location of the vehicle indicates that it is close to the next station. The raw ETA is an estimation of the time required by free-flow speed from point to point

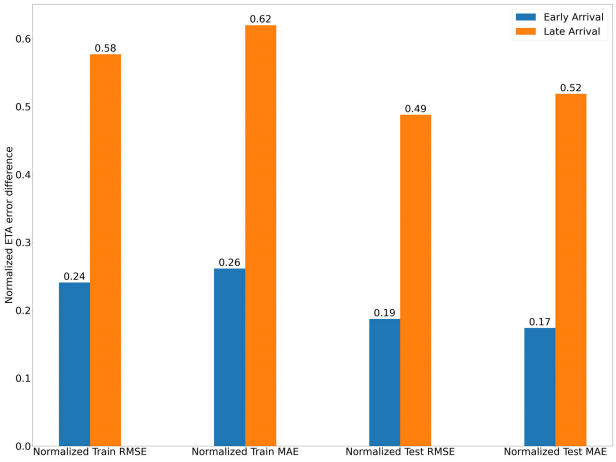


FIGURE 10. Prediction error of early and late arrival.

without the delay caused by bad traffic conditions. Therefore, the early arrival may not be easily predicted from the proxy data. This finding may help us understand more about the cause of ETA errors.

VI. CONCLUSION

A reliable and accurate ETA system is desired by both the transport operators and passengers. Current static distance-based ETA suffers from the deviation by many dynamic traffic factors. To improve the accuracy of the widely used distance-based ETA, we propose a deep learning approach to improve the ETA prediction based on multiple factors that are correlated to the ETA. We use a new neural network structure, DECN, to effectively learn in the large and sparse input feature space. Our experimental results on a large real-world dataset demonstrate the effectiveness of our approach. It can be shown that DECN can learn from the sparse feature and predict the ETA residual that is used to offset the error of distance-based ETA. The resulting average error of ETA is improved by 11% on average, and 49% for late arrival using our approach. In this work, we consider the ETA residual as a deterministic function of the input feature and project the residual using a deterministic neural network. In the future, we may extend this work by using a Bayesian based neural network to capture the stochastic properties of the residual. Another future research direction would be exploring additional input features that are correlated to the ETA for further improving the prediction accuracy. Besides applying the method to public transport, it can be applied to other transport modes such as the ETA of private vehicles and Mobility-as-a-Service [53] and improve the accuracy in a system-wide level. This research work is a significant milestone of improving ETA prediction by using the residual framework, which exceeds the performance of classic methods. Our contribution of introducing DECN to the ETA residual prediction may open the door for many problems in transportation research.

ACKNOWLEDGMENT

The authors would like to thank Dr. Nicky Y. F. Lam for liaising with the Hong Kong Observatory and providing the district-based rainfall data in this article and also would like to thank the support of the RGC Postdoctoral Fellowship Scheme for Dr. Ka Ho Tsoi.

REFERENCES

[1] K.-F. Chu, A. Y. S. Lam, and V. O. K. Li, "Dynamic lane reversal routing and scheduling for connected and autonomous vehicles: Formulation and distributed algorithm," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 6, pp. 2557–2570, Jun. 2020.

[2] S. Kim, M. E. Lewis, and C. C. White, "Optimal vehicle routing with real-time traffic information," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 2, pp. 178–188, Jun. 2005.

[3] K.-F. Chu, A. Y. S. Lam, and V. O. K. Li, "Traffic signal control using end-to-end off-policy deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7184–7195, Jul. 2022.

[4] Y. Liu, K.-W. Chin, C. Yang, and T. He, "Nodes deployment for coverage in rechargeable wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 6064–6073, Jun. 2019.

[5] K. F. Chu, E. R. Magsino, I. W. Ho, and C.-K. Chau, "Index coding of point cloud-based road map data for autonomous driving," in *Proc. IEEE 85th Veh. Technol. Conf. (VTC Spring)*, Jun. 2017, pp. 1–7.

[6] A. Koesdwiady, R. Soua, and F. Karray, "Improving traffic flow prediction with weather information in connected cars: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9508–9517, Dec. 2016.

[7] K. F. Chu, A. Y. S. Lam, and V. O. K. Li, "Travel demand prediction using deep multi-scale convolutional LSTM network," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 1402–1407.

[8] AS. (2021). *Sigspatial Cup*. [Online]. Available: https://sigspatial2021.sigspatial.org/sigspatial-cup/

[9] Z. Huang, J. Xia, F. Li, Z. Li, and Q. Li, "A peak traffic congestion prediction method based on bus driving time," *Entropy*, vol. 21, no. 7, p. 709, Jul. 2019.

[10] L. Meegahapola, N. Athaide, K. Jayarajah, S. Xiang, and A. Misra, "Inferring accurate bus trajectories from noisy estimated arrival time records," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 4517–4524.

[11] K.-F. Chu, A. Y. S. Lam, B. P. Y. Loo, and V. O. K. Li, "Public transport waiting time estimation using semi-supervised graph convolutional networks," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 2259–2264.

[12] K. E. Watkins, B. Ferris, A. Borning, G. S. Rutherford, and D. Layton, "Where is my bus? Impact of mobile real-time information on the perceived and actual wait time of transit riders," *Transp. Res. A, Policy Pract.*, vol. 45, no. 8, pp. 839–848, Oct. 2011.

[13] K. Dziekan and K. Kottenhoff, "Dynamic at-stop real-time information displays for public transport: Effects on customers," *Transp. Res. A, Policy Pract.*, vol. 41, no. 6, pp. 489–501, Jul. 2007.

[14] W.-H. Lin and J. Zeng, "Experimental study of real-time bus arrival time prediction with GPS data," *Transp. Res. Record: J. Transp. Res. Board*, vol. 1666, no. 1, pp. 101–109, Jan. 1999.

[15] D. Sun, H. Luo, L. Fu, W. Liu, X. Liao, and M. Zhao, "Predicting bus arrival time on the basis of global positioning system data," *Transp. Res. Record: J. Transp. Res. Board*, vol. 2034, no. 1, pp. 62–72, Jan. 2007.

[16] M. Chen, X. Liu, J. Xia, and S. I. Chien, "A dynamic bus-arrival time prediction model based on APC data," *Computer-Aided Civil Infrastructure Eng.*, vol. 19, no. 5, pp. 364–376, Sep. 2004.

[17] Y. Lin, X. Yang, N. Zou, and L. Jia, "Real-time bus arrival time prediction: Case study for Jinan, China," *J. Transp. Eng.*, vol. 139, no. 11, pp. 1133–1140, Nov. 2013.

[18] Y. Zhou, L. Yao, Y. Chen, Y. Gong, and J. Lai, "Bus arrival time calculation model based on smart card data," *Transp. Res. C, Emerg. Technol.*, vol. 74, pp. 81–96, Jan. 2017.

[19] P. Wepulanon, A. Sumalee, and W. H. K. Lam, "A real-time bus arrival time information system using crowdsourced smartphone data: A novel framework and simulation experiments," *Transportmetrica B, Transp. Dyn.*, vol. 6, no. 1, pp. 34–53, Jan. 2018.

[20] A. Tirachini, D. A. Hensher, and J. M. Rose, "Crowding in public transport systems: Effects on users, operation and implications for the estimation of demand," *Transp. Res. A, Policy Pract.*, vol. 53, pp. 36–52, Jul. 2013.

[21] J. Krozel, C. Lee, and J. Mitchell, "Estimating time of arrival in heavy weather conditions," in *Proc. Guid., Navigat., Control Conf. Exhib.*, 1999, p. 4232.

[22] Y. Bin, Y. Zhongzhen, and Y. Baozhen, "Bus arrival time prediction using support vector machines," *J. Intell. Transp. Syst.*, vol. 10, no. 4, pp. 151–158, Dec. 2006.

[23] S. I.-J. Chien, Y. Ding, and C. Wei, "Dynamic bus arrival time prediction with artificial neural networks," *J. Transp. Eng.*, vol. 128, no. 5, pp. 429–438, Sep. 2002.

[24] M. Zaki, I. Ashour, M. Zorkany, and B. Hesham, "Online bus arrival time prediction using hybrid neural network and Kalman filter techniques," *Int. J. Mod. Eng. Res.*, vol. 3, no. 4, pp. 2035–2041, 2013.

[25] Z. Wang, M. Liang, and D. Delahaye, "A hybrid machine learning model for short-term estimated time of arrival prediction in terminal manoeuvring area," *Transp. Res. C, Emerg. Technol.*, vol. 95, pp. 280–294, Oct. 2018.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[27] A. Karbassi and M. Barth, "Vehicle route prediction and time of arrival estimation techniques for improved transportation system management," in *Proc. IEEE IV Intell. Vehicles Symp.*, 2003, pp. 511–516.

[28] H. Xu and J. Ying, "Bus arrival time prediction with real-time and historic data," *Cluster Comput.*, vol. 20, no. 4, pp. 3099–3106, Dec. 2017.

[29] B. Y. O. Low, S. H. Dahlan, and M. H. Abd Wahab, "Real-time bus location and arrival information system," in *Proc. IEEE Conf. Wireless Sensors (ICWiSE)*, Oct. 2016, pp. 50–53.

[30] B. Yu, W. H. K. Lam, and M. L. Tam, "Bus arrival time prediction at bus stop with multiple routes," *Transp. Res. C, Emerg. Technol.*, vol. 19, no. 6, pp. 1157–1170, Dec. 2011.

[31] D. Fagan and R. Meier, "Intelligent time of arrival estimation," in *Proc. IEEE Forum Integr. Sustain. Transp. Syst.*, Jun. 2011, pp. 60–66.

[32] J. Ma, J. Chan, G. Ristanoski, S. Rajasegarar, and C. Leckie, "Bus travel time prediction with real-time traffic information," *Transp. Res. C, Emerg. Technol.*, vol. 105, pp. 536–549, Aug. 2019.

[33] Y. Sun, K. Fu, Z. Wang, D. Zhou, K. Wu, J. Ye, and C. Zhang, "CoDriver ETA: Combine driver information in estimated time of arrival by driving style learning auxiliary task," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 5, pp. 4037–4048, May 2022.

[34] X. Mao, T. Cai, W. Peng, and H. Wan, "Estimated time of arrival prediction via modeling the spatial–temporal interactions between links and crosses," in *Proc. 29th Int. Conf. Adv. Geographic Inf. Syst.*, Nov. 2021, pp. 658–661.

[35] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: Deep belief networks with multitask learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2191–2201, Oct. 2014.

[36] K.-F. Chu, A. Y. S. Lam, and V. O. K. Li, "Deep multi-scale convolutional LSTM network for travel demand and origin-destination predictions," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3219–3232, Aug. 2020.

[37] G. Krummenacher, C. S. Ong, S. Koller, S. Kobayashi, and J. M. Buhmann, "Wheel defect detection with machine learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 4, pp. 1176–1187, Apr. 2018.

[38] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. Jennings, and A. Mouzakitis, "Deep learning-based vehicle behavior prediction for autonomous driving applications: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 33–47, Jan. 2022.

[39] J. Li, X. Mei, D. Prokhorov, and D. Tao, "Deep neural network for structural prediction and lane detection in traffic scene," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 690–703, Mar. 2017.

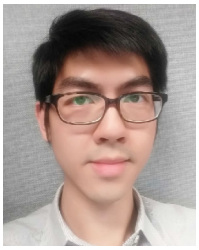
[40] D. Tabernik and D. Skocaj, "Deep learning for large-scale traffic-sign detection and recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1427–1440, Apr. 2020.

[41] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. A. Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4909–4926, Jun. 2022.

[42] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

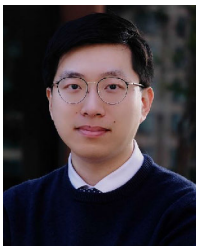
[43] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[44] W. Yang, J. Feng, J. Yang, F. Zhao, J. Liu, Z. Guo, and S. Yan, "Deep edge guided recurrent residual learning for image super-resolution," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5895–5907, Dec. 2017.
[45] D. Lee, J. Yoo, S. Tak, and J. C. Ye, "Deep residual learning for accelerated MRI using magnitude and phase networks," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 9, pp. 1985–1995, Sep. 2018.
[46] A. Abrol, M. Bhattarai, A. Fedorov, Y. Du, S. Plis, V. Calhoun, and A. D. N. Initiative, "Deep residual learning for neuroimaging: An application to predict progression to Alzheimer's disease," *J. Neurosci. methods*, vol. 339, Jun. 2020, Art. no. 108701.
[47] Y. He, L. Dai, and H. Zhang, "Multi-branch deep residual learning for clustering and beamforming in user-centric network," *IEEE Commun. Lett.*, vol. 24, no. 10, pp. 2221–2225, Oct. 2020.
[48] C. Liu, X. Liu, D. W. K. Ng, and J. Yuan, "Deep residual learning for channel estimation in intelligent reflecting surface-assisted multi-user communications," *IEEE Trans. Wireless Commun.*, vol. 21, no. 2, pp. 898–912, Feb. 2022.
[49] M. A. Abdelwahab, M. Abdel-Nasser, and M. Hori, "Reliable and rapid traffic congestion detection approach based on deep residual learning and motion trajectories," *IEEE Access*, vol. 8, pp. 182180–182192, 2020.
[50] *Annual Traffic Census 2020*, Transport Department, Hong Kong, 2021.
[51] TomTom. (2012). *Speed Profiles*. [Online]. Available: https://download.tomtom.com/open/crm/lib/docs/licensing/I.SP.EN.pdf
[52] B. P. Loo and Z. Huang, "Spatio-temporal variations of traffic congestion under work from home (WFH) arrangements: Lessons learned from COVID-19," *Cities*, vol. 124, May 2022, Art. no. 103610.
[53] K.-F. Chu and W. Guo, "Deep reinforcement learning of passenger behavior in multimodal journey planning with proportional fairness," *Neural Comput. Appl.*, to be published, doi: 10.1007/s00521-023-08733-4.



KAI-FUNG CHU (Member, IEEE) received the B.Eng. and M.Sc. degrees (Hons.) in electronic and information engineering from The Hong Kong Polytechnic University, Hong Kong, in 2013 and 2016, respectively, and the Ph.D. degree in electrical and electronic engineering from The University of Hong Kong, Hong Kong, in 2020. He is currently a Research Assistant Professor with the Department of Computing, The Hong Kong Polytechnic University. He was a Research Fellow with the School of Aerospace, Transport and Manufacturing, Cranfield University. He also worked in the industry as an engineer for several years. His research interests include artificial intelligence, optimization, intelligent transportation systems, and autonomous vehicles. He is also an Area Editor of *EAI Transactions on Energy Web*.

ALBERT Y. S. LAM (Senior Member, IEEE) received the B.Eng. degree (Hons.) in information engineering and the Ph.D. degree in electrical and electronic engineering from The University of Hong Kong (HKU), Hong Kong, in 2005 and 2010, respectively. He was a Postdoctoral Scholar with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA, USA, in 2010 and 2012. He is currently the Chief Scientist and the Chief Technology Officer of the Fano Laboratories and an Adjunct Assistant Professor with the Department of Electrical and Electronic Engineering, HKU. His research interests include optimization theory and algorithms, artificial intelligence, smart grids, and smart city. He is also a croucher research fellow. He is also an Associate Editor of IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE, and IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE. He is also the Co-Editor-in-Chief of *EAI Endorsed Transactions on Energy Web*.



KA HO TSOI received the B.Soc.Sc. degree (Hons.) in geography from The University of Hong Kong (HKU), Hong Kong, in 2015, the M.Sc. degree (Hons.) in transport planning from the University of Leeds, U.K., in 2016, and the Ph.D. degree in geography from HKU, in 2021. He is currently a Postdoctoral Fellow with the Department of Geography, HKU. His research interests include sustainable transport, transport decarbonization, road safety, and space-time analysis.

ZHIRAN HUANG received the B.Eng. degree in civil engineering from the Tianjin College, University of Science and Technology Beijing, Tianjin, China, in 2015, the M.Sc. degree in civil engineering for risk mitigation from Politecnico di Milano, Milan, Italy, in 2018, and the Ph.D. degree in geography from The University of Hong Kong, in 2023. He is currently a Postdoctoral Fellow with the Department of Building and Real Estate, The Hong Kong Polytechnic University. His research interests include urban traffic congestion, big data analytics, spatial analysis, and sustainable development.

BECKY P. Y. LOO received the B.A. (Hons.) and Ph.D. degrees in geography from The University of Hong Kong (HKU), Hong Kong, in 1992 and 1998, respectively. She is currently a Professor with the Department of Geography and the Director of the Institute of Transport Studies, HKU. She is also a Chang Jiang Chair Professor with Jiangxi Normal University. Her research interests include smart transport, future cities, and smart technologies. She is also the Co-Editor-in-Chief of *Travel Behaviour and Society* and an Associate Editor of the *Journal of Transport Geography*.