The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)

DeepETA: A Spatial-Temporal Sequential Neural Network Model for Estimating Time of Arrival in Package Delivery System

Fan Wu, Lixia Wu
Artificial Intelligence Department, Zhejiang Cainiao Supply Chain Management Co., Ltd., Hangzhou, China
wf118503@cainiao.com, wallace.wulx@cainiao.com

Abstract

Over 100 million packages are delivered every day in China due to the fast development of e-commerce. Precisely estimating the time of packages' arrival (ETA) is significantly important to improving customers' experience and raising the efficiency of package dispatching. Existing methods mainly focus on predicting the time from an origin to a destination. However, in package delivery problem, one trip contains multiple destinations and the delivery time of all destinations should be predicted at any time. Furthermore, the ETA is affected by many factors especially the sequence of the latest route, the regularity of the delivery pattern and the sequence of packages to be delivered, which are difficult to learn by traditional models. This paper proposed a novel spatial-temporal sequential neural network model (DeepETA) to take fully advantages of the above factors. DeepETA is an end-to-end network that mainly consists of three parts. First, the spatial encoding and the recurrent cells are proposed to capture the spatial-temporal and sequential features of the latest delivery route. Then, two attention-based layers are designed to indicate the most possible ETA from historical frequent and relative delivery routes based on the similarity of the latest route and the future destinations. Finally, a fully connected layer is utilized to jointly learn the delivery time. Experiments on real logistics dataset demonstrate that the proposed approach has outperforming results.

Introduction

Due to the fast development of the e-commerce, over 100 million packages are delivered each day in China. Estimating time of arrival (ETA) of package is important. On one hand, informing the arrival time of packages to customers will help them better arrange when and how to receive their packages, reducing the anxiety of customers and improving the customer experience. More importantly, the ETA will help measuring the service ability and quality of couriers, which are key parameters in the last mile pickup and delivery system. With the efforts made by the logistics companies such as the Cainiao Ltd., the traditional package delivery process is digitized and massive amount of delivery data are collected. The problem of how to precisely predict the ETA of packages has gained increasing attention in logistics research communities.

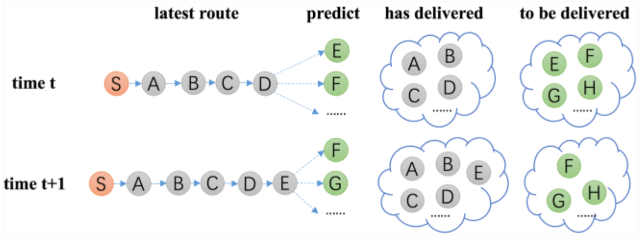

Figure 1: The delivery times of all packages at any time are affected by the latest route, the delivered pattern and the to-be-delivered pattern.

In the real scene, each courier should deliver nearly 100 packages per day. When customers demand, the delivery time of all undelivered packages should be predicted at the same time, which is a multi-destination prediction problem. However, the problem can be referred from similar scenes, such as the car sharing and the free rider problem, where the travel time on the road and the sequence of the route are important. There exists valuable researches that can be referred, such as predicting the next location (Feng et al. 2018), mining delivery pattern (Ying, Lee, and Tseng 2013), and estimating time on the road (Jindal et al. 2018). Furthermore, the ETA of package delivery should consider both spatial and temporal features of the delivery route and researches can be found that propose spatial-temporal models in solving similar problems (Liang et al. 2018).

However, predicting the package delivering time is challenging mainly of the following reasons:

Multiple destinations. Predicting the travel time in transportation problem focuses on the time difference between an origin and a destination. However, in delivery system all undelivered packages should be predicted at any time. The delivery time of different locations may vary due to the delivery sequence and the locations of the packages.

Time-variant delivery status. As shown in Figure 1, the delivery time is affected by many factors especially the sequence of the latest route, the regularity of the delivery pattern and the sequence of packages to be delivered, which are difficult to learn by traditional model. Though recurrent neural networks (RNN) can learn sequential features, it could not handle frequent patterns and regularities of delivery routes.

Time-invariant delivery features. The geographical locations of the packages have huge influence on the delivery sequence and thus determine the delivery time. The representation of the location in the model becomes an important issue. Furthermore, the inherent properties of packages such as the weight or size of the package should be considered.

To overcome the aforementioned difficulties, this paper proposed a novel wide and deep neural network for estimating time of package arrival (DeepETA). DeepETA has specially designed architecture to handle all the relative features of package delivery. The main contributions are as follows:

We develop a spatial-temporal module to capture the sequential features of the latest delivery route. Different from traditional methods that use one-hot or convolution layer to represent geo-locations, we first encode the location according to the geographical proximity and then embed them into short vectors. Combining with delivering status of each node in the delivery route, the long short-term memory cells (LSTM) are used to extract the sequential features of the route.

We design two attention-based modules to learn historical frequent and relative delivery patterns. To tackle the difficulty that RNN cannot learn the correlations between massive historical data, we first extract relative delivery routes and utilize attention mechanism to find the most similar route. Both delivered and undelivered packages are taken into account to enrich the model.

We evaluate the proposed method on a real-world logistics dataset. The results show that our approach outperforms the competing methods.

Related Work

The problem of predicting the delivery time of multiple destinations can be referred from predicting the next location, estimating the travel time of vehicles on the road, and spatial-temporal model used for time series prediction.

Next location prediction: (Ying, Lee, and Tseng 2013) builds the frequent pattern tree and utilizes traditional machine learning methods to predict the next location. (Wu et al. 2017) proposes an LSTM network that can learn the path sequential features. (Zhang et al. 2018) predicts taxi destination by transforming raw trajectories into image and used convolutional neural network (CNN) to extract deep spatial features. RNN is used to model temporal and sequential features. (Feng et al. 2018) develops an attentional recurrent model considering both heterogeneous transition regularity and multi-level periodicity. If the next location is precisely predicted, the travel time can be simply calculated by the distance and velocity.

Estimating travel time: (Wang, Fu and Ye 2018) uses ensemble model that combines linear models, deep neural network (DNN) and RNN. (Jindal et al. 2018) proposes two DNN modules to capture coordinates and time attributes from raw trajectories. (Zhang et al. 2018) develops a bi-directional LSTM layer to capture short-term and long-term traffic features. (Li et al. 2018) utilizes multi-task learning to predict the additional feature of the path and jointly learn the main task. (Wang et al. 2018) designs an end-to-end network that contains a geo-convolution layer to represent raw trajectory and used multi-task to learn both the entire path and each local path.

Spatial-temporal data prediction: Delivery time prediction is a time-series problem and methods in similar fields can be inferred. (Shen et al. 2018) treats the time series data as video and proposed a CNN to simultaneously model all correlated spatial-temporal mobility patterns. (Liang et al. 2018) proposes a multi-level attention networks for geo-sensory time series prediction and utilizes spatial attention to capture the geographical correlations. (Yao et al. 2018) uses geo-convolutional layer to model spatial relations and LSTM to model the time series. (Zhang et al. 2018) extracts distant, near and recent flows manually to model temporal features and uses residual networks to better train the deep networks. (Yao et al. 2018) utilizes the attention mechanism to find the periodicity and temporal shifting.

Preliminaries

In this section, the delivery time prediction problem is defined and related notations are explained.

Locations in delivery route: Packages are clustered into areas like communities or blocks to reduce the scale of the problem and to improve the efficiency of training. We use areas to represent the locations in the delivery route and predicts the delivery time to each location. In some previous literatures, locations are aggregated by rectangle or hexagonal partitions of the city (Wang, Fu and Ye 2018). However, those methods lose geographic and semantic information of the area and suffer from large granularity problem. We utilize an optimized partition method based on road network and areas of interest (AOI). The partition function of mapping package p_i to location a_i is defined as $a_i = O(p_i)$.

Delivery route: After the aggregation step, the task is transformed to predict the deliver time of a set of locations, namely $Set = \{a_i \mid i = 0, \cdots, n\}$. The sequence of locations has great impact on the delivery time. We define the delivery sequence at time t_i in day d as:

$$Route^{d}_{a_c,t_i} = \{node_{a_s,t_0} \rightarrow node_{a_1,t_1} \cdots node_{a_c,t_i}\}, \quad (1)$$

where $node_{a_c,t_i}$ is a spatial-temporal node that means location a_c is visited at time t_i.

Problem definition: The delivery time dt_i of location a_i is the travel time from the current location a_c to destination a_i. The sequence of the current route and the undelivered location set have great influence on the delivery time. We develop a deep learning method to learn the regularity from massive historical data. First, given the latest route $Route^{d}_{a_s,a_c}$ and the predicted location a_i, we find all relative routes from history that are similar with the current route, symbolized as set $\mathcal{H}_{route}$. Then, given the undelivered location set $Set^{d}_{t_i,}$, we find all routes from history that has similar undelivered set, marked as $\mathcal{F}_{set}$. All records have a delivery time of a_i. The objective of our network is to find the most possible delivery time dt_{a_i} from historical relative routes:

$$dt_{a_i} = f(\mathcal{H}_{route}, \mathcal{F}_{set}) \quad (2)$$

Proposed DeepETA Framework

In this section, the proposed spatial-temporal sequential network for estimating time of package arrival (DeepETA) is described in detail. Figure 2 shows the architecture of the proposed method. The DeepETA is an end-to-end network that takes time-variant route feature and time-invariant feature as input, and output the delivery time. DeepETA consists of three modules, namely the latest route encoder, the frequent pattern encoder, and the prediction module.

Latest Route Encoder

This module aims to capture the complicated sequential information that influences the delivery time. The delivery route is consecutive in time and is adjacent in space. Previous location prediction literatures, such as (Monreale et al. 2009; and Ying et al. 2013), claim that human mobility can be regarded as a probability chain. The future locations or behaviors can be predicted through the transition probability matrix given the past behaviors. Hidden Markov Model (HMM) are used to model the process. However, HMM suffers from the deficiency of learning long-term dependencies. Recently, RNN has gained a breakthrough in sequential mining. (Mikolov et al. 2010) develops RNNs in word embedding for sentence modeling. Multiple hidden layers in RNN can adjust dynamically with the input of behavioral history and the transition probabilities can be transmitted through the whole sequence. In this paper, the LSTM cells (Hochreiter and Schmidhuber 1997) are used to overcome the gradient vanishing or exploding problem of RNN. In delivery problem, locations are also important as the geographical distance decides the sequence of delivery. We develop a spatial encoder to vectorize the locations.

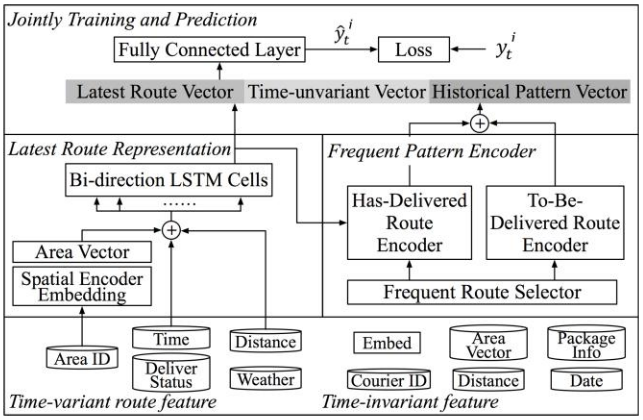

Figure 2: The framework of the proposed DeepETA. The model inputs consist of time-variant route features and time-invariant features and the output y_t^i is the delivery time of package a_i at time t.

Spatial encoder: Locations in the delivery route have spatial correlations. Traditional methods use one-hot encoder to represent locations and manually extract the spatial features, which may lose the information of distant areas. Recently, DNN and CNN are widely used (Zhang et al. 2018; and Shen et al. 2018) because they can automatically extract the spatial relation. However, CNN needs to split the space into grids with the same height and width, which may cause the uncertainty granularity problem. If the size is too big, we cannot distinguish different ETAs in that large area. If it is too small, neighborhood may be divided into several grids and regularity in same area may be lost. In this paper, we proposed a geocoding-based encoder to represent locations by their inherent geographical attributes.

We first utilize a road network based methods to split area into neighborhoods and use Geohash to represent the area by the centroid coordinate. Geohash is a geocoding system which encodes a geographic location into a short string of letters or digits and nearby places will often present similar prefixes. The longer a shared prefix is, the closer the two places are. Furthermore, the digital representation of Geohash requires each bit to be either 0 or 1 that can be utilized by the neural networks. First, location a_i is transformed into Geohash encoding of 40 digits, namely G_{a_i}. Then an embedded layer is added to represent G_{a_i} aiming at reducing computation cost without losing much information, formulated as:

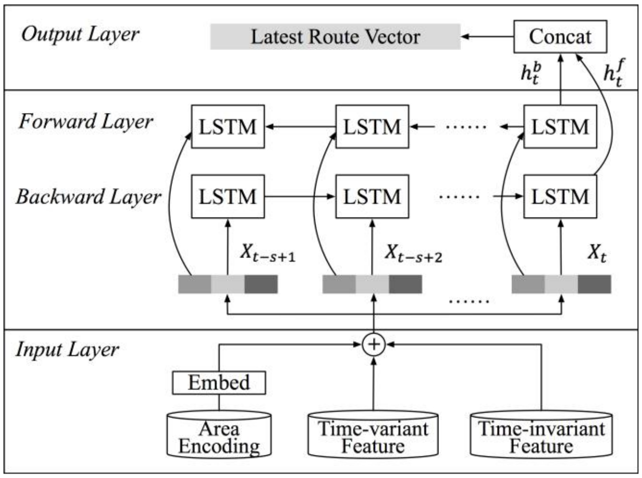

Figure 3: The structure of the Latest Route Representation layer. The locations are encoded by the spatial encoder. Then features of each node in the delivery route are exported to a bidirectional LSTM layer. The outputs of the last cell of the backward layer h_t^b and the forward layer h_t^f are combined to form the latest route vector.

$$V_{a_i} = f\left(W_a G_{a_i} + b_a\right), \tag{3}$$

where W_a and b_a are learnable parameters of the spatial embedding layer and the Relu activation function is utilized to add non-linearity.

BiLSTM: The LSTM cell is able to capture the temporal sequential dependency, which improves the weakness of gradient exploding and vanishing of traditional RNN. As compared with LSTM, bidirectional LSTM (BiLSTM) (Graves and Schmidhuber 2005) utilizes additional backward information and thus enhances the memory capability. In the latest route representation layer, we use BiLSTM to capture the transition probability of each node in the route and to infer delivery time by the hidden state vector of the last cell.

We concatenate the vectors of the spatial encoder V_{a_i}, with time-variant features V_{tv} and time-invariant features V_{ti} to get the global feature vector X of each timestep, i.e., $X_t = [V_{a_i}, V_{tv}, V_{ti}]$. A set of X contained fixed length of time steps is fed into the BiLSTM layer. LSTM utilizes two gates to control the cell state. The forget gate decides how much information of the last cell state c_{t-1} will keep to the current time c_t. The other is the input gate, which decides how much information of the input of the current networks X_t will keep to cell state c_t. LSTM uses the output gate to control how much information of the cell state c_t will output to h_t. Thus the conveyor belt-like structure allows LSTM to remove or add information from the very beginning to the current state. Then we get the latest hidden states h_t^b and h_t^f of the forward and backward layer. Each of them can be calculated by $h_t = LSTM(h_{t-1}, X_t)$. Finally, these two states are concatenated to get the hidden state of the latest route $h_t = [h_t^b, h_t^f]$.

Frequent Pattern Encoder

The frequent pattern encoder is designed to capture the frequent mobility patterns by jointly selecting the most related historical delivery routes under the current delivery status. The module consists of two parts. The route encoder first extracts spatial-temporal features from the historical delivery routes. Then these features are selected by an attention-based layer based on the latest route vector to generate the most related pattern. By combining this vector with the latest route, we could predict the delivery time based on not only the sequential relation but also the frequent pattern of the historical routes.

Route encoder: Although the LSTM cell improves the problem of gradient exploding and vanishing, the performance of LSTM drops significantly when the length of time step is very long (Bengio, Simard, and Frasconi 1994). Simply importing all historical routes into the recurrent layer may reduce the effectiveness and increase training difficulty. In neural machine translation, it suffers from similar problem that RNN cannot memorize long sentences. (Bahdanau, Cho, and Bengio 2014) develops the attention mechanism which can selectively maintain information about the most relative word. Furthermore, attention is widely used in object recognition (Xu et al. 2015) to recognize the most interested area from the whole image.

Different from the attention mechanism, which requires the whole sentences, we design a route selector that extracts only the relative frequent patterns. Assuming that the latest route at time t_i is $Route^d_{a_s,a_c}$ and the undelivered location set is $Set^d_{a_i,t_i}$, we separately select the frequent patterns by the following rules:

Delivered route pattern: Given the current location a_i, the travel time of the latest delivery route t_i and the predicted location a_i, we find historical delivery routes that have the same current location and travel time, which will significantly reduce the scale of the candidates. Then we group these routes by the discrete delivery time bin (30 min) of a_i and find the top 10 most regular routes for each time bin, defined as $\mathcal{H}_{route}$ in Eq. (2). The task is to find which historical route is the most similar to the current route. So we utilized LSTM cells to represent the frequent routes:

$$\widetilde{\mathcal{H}}_{\mathrm{t}}^{\mathrm{k}} = \mathrm{LSTM}\left(\widetilde{\mathcal{H}}_{\mathrm{t-1}}^{\mathrm{k}}, \mathcal{H}_{route}\right), \tag{4}$$

where k is the class of delivery time. $\widetilde{\mathcal{H}}_t^k$ is the context vector of all frequent routes that has k delivery time and we compute the average value of all vectors in each time bin. Furthermore, statistic features such as the frequency of each time bin are concatenated with the context vector.

To-be-delivered pattern: Different from the delivered route, the exact delivery order is unknown. So we use the current location a_c, the number of locations in the delivery set n_i and the predicted location a_i to find historical to-be-delivered sets. Similar with the above step, we group these sets by the same time bin as the delivered pattern module and 10 frequent sets are selected in each time bin, marked as $\mathcal{F}_{\text{set}}$. As the sequence of the current to-be-delivered locations is hard to predict, it is not proper to use sequential layer like LSTM. Instead, a DNN is developed to model the unordered set and reduce the impact from sequences:

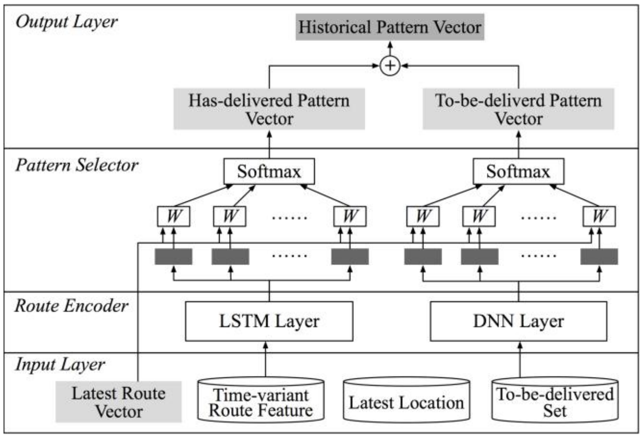

Figure 4: The architecture of the frequent pattern encoder. This module has the same input with the network and is connected to the latest route module. The output is the representation of historical relative pattern.

$$\tilde{\mathcal{F}}_t^k = DNN\left(\tilde{\mathcal{F}}_{t-1}^k, \mathcal{F}_{set}\right), \quad (5)$$

Pattern selector: The goal of the pattern selector is to find which frequent pattern has the biggest impact on the current situation. Traditional pattern mining methods utilize similarity measurements such as the cosine similarity, Levenshtein distance, and the time dynamic wrapping. However, pattern mining methods may suffer from data sparseness problem and complex features cannot be weighted. An attention-based layer is designed to calculate the similarity between the latest route and the frequent routes. First, the frequent pattern vectors are combined with the latest route vector through a score function:

$$f\left(\widetilde{\mathcal{H}}_t^k, h_t\right) = tanh\left(\widetilde{\mathcal{H}}_t^k W_{score} h_t\right), \quad (6)$$

where $\widetilde{\mathcal{H}}_t^k$ or $\tilde{\mathcal{F}}_t^k$ is the frequent pattern vectors, W_{score} is a learnable parameter, and h_t is the latest route vector. Then all scored vectors are exported to the softmax layer to calculate the weight of each vector. The softmax is the extension of the sigmoid function to the multi-class problem, which transforms the K dimension variable into another K dimension variable within $(0,1)$:

$$\sigma(z)_k = \frac{e^{z_k}}{\sum_{i=1}^{K} e^{z_i}}, \quad (7)$$

where $z = f\left(\widetilde{\mathcal{H}}_t^k, h_t\right)$ and K is the total number of the delivery time bins. Finally, the value of the softmax is multiply with the historical pattern vectors that can illustrate the importance of different patterns:

$$V_{fp} = \sum\left(\sigma(z)_j \widetilde{\mathcal{H}}_t^j\right), \quad (8)$$

Jointly Training and Prediction

The predicted delivery time of location a_i at time t is relative with the properties of packages in a_i, the latest route features and the frequent patterns. We concatenate the outputs from the latest route module h_t and the frequent pattern encoder V_{fp}, together with the time-invariant features V_{ti}:

$$\tilde{X}_t = \left[h_t, V_{fp}, V_{ti}\right], \quad (9)$$

Then $\tilde{X}_t$ is fed into a fully connected layer to get the final prediction value $\tilde{y}_t$:

$$\tilde{y}_t = \sigma\left(W_{fc}\tilde{X}_t + b_{fc}\right), \quad (10)$$

where W_{fc} and b_{fc} are learnable parameters and $\sigma(x)$ is the activation function of the last fully connected layer. The Sigmoid function defined as $\sigma(x) = 1/(1 + e^{-x})$ is used to restrict the output in $[0,1]$, as the prediction values are normalized.

The loss function consists of two parts: the mean square error (MSE) and the mean absolute percentage square error (MAPSE). MSE is like a combination measurement of bias and variance of the prediction but is sensitive to large prediction values. However, in delivery task, there might be some cases that the customer is not at home and requires a second delivery which leads to large ETA. MAPSE gives less weight to outliers, which is not sensitive to outliers. So we decide to combine the advantages of MSE and MAPSE to make the prediction mainly focus on normal and small ETA to reducing the influence of outliers. There exists a jointly training trick by adding a hyper parameter to adjust the weight of MSE and MAPSE. The loss function is defined as follows:

$$\mathcal{L}(\theta) = \sum_{i=1}^{N}\left((\tilde{y}_t - y_t)^2 + \lambda\left(\frac{\tilde{y}_t - y_t}{y_t}\right)^2\right), \quad (11)$$

where θ represents all learnable parameters in the whole network and λ is the hyper parameter. All module in DeepETA is parameterized as a feed-forward neural network that can be trained in the whole network. During the training phase, the Adam optimizer (Kingma and Ba 2014) is used to minimize the loss function.

Experiments

Dataset Description

The experiment is conducted on a real-world package delivery dataset collected by Cainiao Ltd., which is one of the largest logistics companies in China, handling over a hundred million packages per day. The dataset contains the delivery routes of 331 couriers from Jun. 1, 2018 to Aug. 1, 2018 (60 days), in Beijing, China. As shown in Table 1, each courier delivers 80 packages in 2 delivery routes every day and the average working time is 8 hours. The average delivery time of all the packages is 3.5 hours. The average number of nodes in each delivery route is 20. Each sample of the dataset contains the static properties of packages, such as the location, weight, date, holiday and courier ID. Also the time-variant features, such as the latest route info. Each node in the route consists of the current location, time, distance to the prediction location and delivery status. In total there are 350 thousand samples to be predicted. The previous 50 days are used as training set and the last 10 days as testing set.

Evaluation Metric

Mean Average Percentage Error (MAPE) and Rooted Mean Square Error (RMSE) are used to evaluate the proposed methods. MAPE can intuitively show the deviation between the prediction and the ground truth, and RMSE is the absolute value to show the performance of the model. Both of them are widely used in time series prediction problem, which are defined as follows:

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|\hat{y}_t^j - y_t^j|}{y_t^j}, \quad (12)$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_t^j - y_t^j)^2}, \quad (13)$$

where $\hat{y}_t^j$ and y_t^j are the prediction and the ground truth of the deliver time of location a_j at time t, and N is the total number of samples.

Methods for Comparison

The proposed methods are compared with the following methods:

Linear regression (LR): We use Lasso (i.e., with ℓ_1-norm regularization) as the linear regression method.

XGBoost (Chen and Guestrin 2016): XGBoost is widely used in many machine learning problems and data science challenges and it always surpasses other traditional models. A common trick on utilizing traditional model in sequential data is to unfold the time series and concatenate each time step together.

Deep neural network (DNN): A neural network of four fully connected layers to extract the high level correlation between the combining vectors. The number of hidden units are 128, 128, 64, and 32 respectively.

Table 1. Statistics of the delivery status of each courier per day.

Pkgs	Route	Nodes	Time	AvgDt
80	2	20	8h	3.5h

LSTM: A stacked LSTM network to model sequential route features. The hidden units are 64 and 64.

DeepTTE (Wang et al. 2018): A state-of-art network in travel time prediction which utilizes geo-convolutional layer to represent raw trajectories and uses a combination of the LSTM cell and attention mechanism to learn long-term dependency of one route. We replace the convolution layer with our location vector and uses the default settings in the open source code.

DeepMove (Feng et al. 2018): A state-of-art method in next destination prediction that has a recurrent layer to model current movement and uses attention mechanism to learn historical patterns. We modify the output softmax layer to a fully connected layer to make regression.

Furthermore, the effects of different modules in DeepETA are evaluated.

Latest route representation (BiLSTM): Only the latest route module is reserved to see the effectiveness of modeling sequential route through the BiLSTM.

Frequent pattern encoder (BiLSTM+DP and BiLSTM +TP): In the frequent pattern encoder, both historical delivered patterns and to-be-delivered patterns that matter. First, only the delivered patterns are remained. Then we adapt the to-be-delivered patterns individually.

Preprocessing and Parameters

In the courier dataset, the average route length is 20 and 90% of the routes have length larger than 50. Long sequence will significantly drop the accuracy and efficiency of the network. We use right padding method to normalize the route of a fixed length 20. When the length is over 20, only the latest 20 nodes are reserved. All missing values are replaced by -1.0 and are normalized to $[0,1]$. The delivery time is normalized by dividing 720 and delivery time over 720 minutes are all considered as 720. When evaluate, we multiply the normalized prediction by 720 to get the real delivery time in minutes. The batch size of each epoch is 1024 and the learning rate is 0.001. The hidden units in the BiLSTM are set to 64 and the LSTM cell in frequent route encoder is also 64. The class number of the delivery time in the route selector is 28 and the final output layer consists of two fully connected layers, which has 64 and 32 units. We further conduct experiment to adjust the hyper-parameter in the loss functions: only RMSE, only MAPE and the combination. The experiment shows that the mix loss function reduces 10.8% error compared to single metrics.

Table 2. The performance of different baselines and DeepETA.

Methods	**RMSE** (min)	**MAPE** (%)
LR	144.18	43.5
DNN	127.58	37.4
XGBoost	123.66	38.2
LSTM	110.37	34.9
DeepTTE	97.85	29.7
DeepMove	72.43	24.3
DeepETA	**63.58**	**20.6**

Model Comparison

Table 2 shows the performance of the proposed method compared to the baseline models. DeepETA achieves the lowest RMSE (63.58 minutes) and the lowest MAPE which improves the best performance of the baseline methods by 13.8% (RMSE) and 16.5% (MAPE). The Lasso based linear regression performs poorly because it only utilizes the properties of time series and could not learn neither short-term or long-term dependencies. Regression methods such as DNN and XGBoost unfold sequential data into unordered vectors and can learn the co-occurrence of each time step, which achieve better performance. Simple LSTM network can extract the correlations from the beginning of the route to the end. However, it suffers the long-term dependency problem when dealing long sequence. DeepTTE gains better performance than LSTM (11.8% improvement of RMSE) because it uses attention mechanism to enhance the ability of learning long sequences. Traditional methods only focus on modeling each route while DeepMove is designed to utilized historical frequent mobility, which lead to a significantly enhance (25.5% improvement of RMSE than DeepTTE). Our method DeepETA gains an improvement of 13.8% in RMSE than DeepMove. The main difference is that in the attention layer, DeepMove uses sampling method to extract high level features to represent historical trajectories and we develop a LSTM-based layer to extract the sequential features from raw routes. Meanwhile, we are more concerned about the sequence of the undelivered set and specially design a DNN layer to focus on these features.

Effectiveness of model components

The latest route representation layer of DeepETA aims to extract sequential relations of the current route by BiLSTM cells. Purely rely on this layer, the performance outperforms the LSTM cell slightly (an improving of 6%). As shown in Figure 5a, when adding the frequent pattern encoder, the performance significantly increases. The delivered pattern encoder improves the RMSE by 36.5% compared to BiLSTM only. The to-be-delivered pattern encoder has an improvement of 29.8% and the combination of all modules improves the overall performance by 3%, which makes the DeepETA to achieve the best score among baseline models.

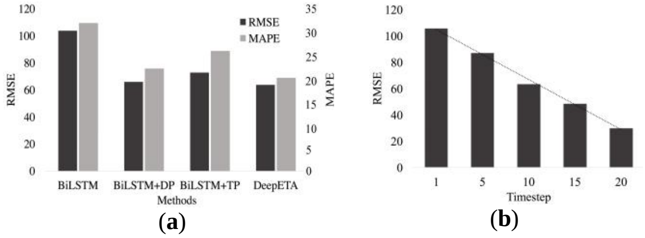

Figure 5: (a) The effectiveness of the components of DeepETA. (b) The performance of predictions at different time.

The differences between the latest route and the frequent pattern encoders is that if we just put all historical routes into the model without any frequent patterns selected ahead, the results are poorly. The reason is that the LSTM-based module may not memory the regularity among all historical routes and similar patterns may be lost. Delivered route and undelivered pattern are treated separately as we do not know the delivery sequence of undelivered packages and sequential features should not be considered in modeling undelivered pattern. As a result, we have two probability distributions drew differently from delivered and undelivered patterns.

Performance of prediction at different time

The goal of the delivery time prediction is to predict all the undelivered packages at any time and there exists a recomputation of estimating time after each package is delivered. With the processing of delivery, more current route information can be inferred and the scope of undelivered sets can be reduced. It can be assumed that more accurate predictions can be made at the end of the delivery task. For example, if only one package is left, the delivery time can be easily inferred by the distance from the current location. Figure 5b illustrates the result of this assumption. The length of the current delivery route can be regarded as time step. During the training step, max length of 20 is used to pad the route. So we manually select route length of 1 (as soon as the package is delivered), 5, 10, 15, 20 (only one package is left to predict) and import them to the model. As shown in Figure 5b, the RMSE decrease significantly as the increasing of time step. At the beginning of the delivery, the prediction deviation is around 106 minutes. The frequent patterns are hard to find as the lack of current route information and the result is similar with using BiLSTM only. Then the prediction error drops harshly and achieves the minimum of 29 minutes at the end of the route. The less the remaining packages are, the more accurate the model predicts because the uncertainty drops when there are less remaining packages.

Conclusion

In this paper, we propose a deep spatial-temporal sequential model for estimating the package delivery time (DeepETA). First, the latest route encoder embeds the location of packages that remain geographical relations and BiLSTM is used to model the sequential features. Then, the frequent pattern encoder selects the frequent routes from historical data and uses LSTM and DNN to represent the routes. An attention-based layer is developed to calculated the most similar patterns with the current route. Finally, by combining all these features, jointly training is utilized to minimize the loss function. Experiments on real logistics dataset show that the proposed method overwhelms the start-of-art methods and the effectiveness of three modules are illustrated.

In the future, we will extend our work in the following aspects. First, we use the unordered undelivered set as the sequence of the route is unknown. If the sequence of the route can be precisely predicted, the delivery time is easy to be inferred. Then, we predict the to-be-delivered packages at time t separately. Inspire by the real-time neural machine translation and the sequence to sequence model, the model can predict the following multiple time steps at once without losing accuracy.

References

Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; ... and Kudlur, M. 2016. Tensorflow: a system for large-scale machine learning. In *Proceedings of the 12th Symposium on Operating Systems Design and Implementation*, 265-283.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bengio, Y.; Simard, P.; and Frasconi, P. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2): 157-166.

Chen, T., and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 785-794. ACM.

Feng, J.; Li, Y.; Zhang, C.; Sun, F.; Meng, F.; Guo, A.; and Jin, D. 2018. DeepMove: Predicting Human Mobility with Attentional Recurrent Networks. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, 1459-1468.

Graves, A., and Schmidhuber, J. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5): 602-610.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735-1780.

Jindal, I.; Chen, X.; Nokleby, M.; and Ye, J. 2017. A Unified Neural Network Approach for Estimating Travel Time and Distance for a Taxi Trip. *arXiv preprint arXiv:1710.04350*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Liang, Y.; Ke, S.; Zhang, J.; Yi, X.; and Zheng, Y. 2018. GeoMAN: Multi-level Attention Networks for Geo-sensory Time Series Prediction. In *Proceedings of the 27th International Joint Conferences on Artificial Intelligence*, 3428-3434.

Li, Y.; Fu, K.; Wang, Z.; Shahabi, C.; Ye, J.; and Liu, Y. 2018. Multi-task representation learning for travel time estimation. In *Proceedings of the 24th International Conference on Knowledge Discovery and Data Mining*, 1695-1704.

Mikolov, T.; Karafiát, M.; Burget, L.; Cernocký, J.; and Khudanpur, S. 2010. Recurrent neural network based language model. In *Proceedings of the Eleventh Conference of the International Speech Communication Association*, 1045–1048.

Monreale, A.; Pinelli, F.; Trasarti, R.; and Giannotti, F. 2009. Wherenext: A location predictor on trajectory pattern mining. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 637–646. ACM.

Shen, B.; Liang, X.; Ouyang, Y.; Liu, M.; Zheng, W.; and Carley, K. 2018. StepDeep: A Novel Spatial-temporal Mobility Event Prediction Framework based on Deep Neural Network. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 724-733. ACM.

Wang, D.; Zhang, J.; Cao, W.; Li, J.; and Zheng, Y. 2018. When Will You Arrive? Estimating Travel Time Based on Deep Neural Networks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*.

Wang, Z.; Fu, K.; and Ye, J. 2018. Learning to Estimate the Travel Time. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 858-866. ACM.

Wu, F.; Fu, K.; Wang, Y.; Xiao, Z.; and Fu, X. 2017. A Spatial-Temporal-Semantic Neural Network Algorithm for Location Prediction on Moving Objects. *Algorithms*, 10(2): 37.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; ... and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, 2048-2057.

Yao, H.; Tang, X.; Wei, H.; Zheng, G.; Yu, Y.; and Li, Z. 2018. Modeling Spatial-Temporal Dynamics for Traffic Prediction. *arXiv preprint arXiv:1803.01254*.

Yao, H.; Wu, F.; Ke, J.; Tang, X.; Jia, Y.; Lu, S.; ... and Ye, J. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*.

Ying, J.; Lee, W.; and Tseng, V. 2013. Mining geographic-temporal-semantic patterns in trajectories for location prediction. *ACM Transactions on Intelligent Systems and Technology*, 5(1): 2. ACM.

Zhang, H.; Wu, H.; Sun, W.; and Zheng, B. 2018. DeepTravel: a Neural Network Based Travel Time Estimation Model with Auxiliary Supervision. *arXiv preprint arXiv:1802.02147*.

Zhang, J.; Zheng, Y.; Qi, D.; Li, R.; Yi, X.; and Li, T. 2018. Predicting citywide crowd flows using deep spatio-temporal residual networks. *Artificial Intelligence*, 259: 147-166.

Zhang, L.; Zhang, G.; Liang, Z.; and Ozioko, E. F. 2018. Multi-features taxi destination prediction with frequency domain processing. *PloS one*, 13(3): e0194629.