INSS Press

Journal of Intelligent and Sustainable Systems

A Deep Learning Algorithm for Travel Time Prediction

Zahra Ejabati Emanab[1], Md Anisul Islam[2]*, Yuvraj Gajpal[3]

[1] Asper School of Business, University of Manitoba, Winnipeg, R3T 5V4, Canada;
zahraejabati@gmail.com
[2] Shannon School of Business, Cape Breton University, Sydney, B1P 6L2, Canada;
mdanisul_islam@cbu.ca
[3] Asper School of Business, University of Manitoba, Winnipeg, R3T 5V4, Canada;
yuvraj.gajpal@umanitoba.ca

*Corresponding author: Md Anisul Islam: mdanisul_islam@cbu.ca

Abstract
Accurate prediction of vehicle travel times is crucial for enhancing intelligent transportation systems, optimizing routing solutions, improving ride-sharing services, and managing traffic effectively. There are various methods available for predicting vehicle travel times between two locations, including both model-based and data-driven approaches. Traditional models often fall short because they assume Euclidean distance when predicting travel times between points. In this study, we focus on predicting vehicle travel times for road segments and entire routes using detailed trajectory data that includes latitude, longitude, time of day, time of week, driver habits, and driver ID. Each trajectory consists of a sequence of GPS points that track a vehicle's movements over time. By defining a road segment as the route between three consecutive GPS points, we can break down the trajectory into smaller segments, enabling more accurate travel time estimates. Given the complexity of travel time prediction, which is influenced by traffic flow conditions at different times and locations, we propose a deep learning algorithm. This algorithm utilizes advanced techniques, including Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and Temporal Convolutional Networks (TCNs). Our approach demonstrates significant improvements over existing methods. Using the Mean Absolute Percent Error (MAPE) metric, we compared our model with established ones, employing large-scale Chengdu taxi datasets. Our results indicate a 2.9% improvement in travel time prediction accuracy, highlighting our model's potential to surpass current solutions and paving the way for future research in travel time estimation.

Keywords: Travel time prediction; Trajectory data; Deep learning; Long short-term memory networks; Convolutional neural networks; Temporal convolutional networks

1. Introduction

Accurate travel times serve as a vital component in enhancing transportation systems, leading to numerous positive outcomes. By optimizing operational costs, sustainability, service quality, and customer satisfaction, accurate travel times also improve emergency response times and inform infrastructure investment decisions to enable the efficient movement of people and goods in logistics. When transportation planners and stakeholders can accurately predict vehicle travel times, they are empowered to make proactive, informed decisions about selecting routes and modes that are flexible, cost-effective, sustainable, and competitive. This adaptability enables a quicker response to delays, changing conditions, traffic congestion, and overall network performance, ultimately creating a more predictable experience for everyone in the system [1, 2]. Such an approach significantly reduces anxiety and frustration by providing drivers with real-time alternative routes, which enhances cooperation and satisfaction throughout the logistics system.

Enhancing travel time predictions offers a powerful opportunity to boost sustainability efforts significantly. By optimizing routing for personal, business, and public transportation, we can effectively reduce congestion and fuel consumption, thereby lowering carbon emissions and increasing efficiency. This improvement makes public transportation a more attractive option, encouraging more individuals to choose it over private vehicles, thereby significantly reducing overall traffic levels. Moreover, accurate travel time predictions pave the way for better urban planning and resource management, contributing to the development of sustainable transportation systems across cities [3, 4]. In business logistics, more accurate travel time forecasts can improve operational efficiency. By enabling thorough planning, reducing costs, and enhancing reliability, businesses can optimize inventory management and allocate resources more effectively at facilities. This not only reduces operational costs but also enhances customer service, creating a win-win situation for both companies and their clients [5, 6].

When considering travel time from a route perspective, it encompasses the total duration needed to complete a specific route, including any waiting times between segments, from the starting point to the final destination. From a network standpoint, travel time impacts overall efficiency and planning. Optimizing these aspects can enhance travel experiences, improve logistics, and other perspectives. The time required to navigate particular road segments (or arcs) varies with departure time and current traffic conditions. This variability in travel time profoundly affects the identification of the shortest path between two locations in a complex, dynamic roadway system. Travel time is systematically calculated by aggregating the duration of each segment along the route, with each segment's duration being influenced by factors such as speed, distance, and time-dependent conditions like traffic patterns and weather. Importantly, while measuring travel time is essential for road networks, it is equally crucial for other transportation modes, including railways, to ensure a comprehensive understanding of logistics efficiency [7, 8].

Navigation applications like Google Maps, Waze, Apple Maps, and HERE WeGo play a vital role in enhancing travel experiences by providing accurate travel time estimates. These estimates are fundamental to Intelligent Transportation Systems (ITS), Advanced Traveler Information Systems (ATIS), Advanced Driver Assistance Systems (ADAS), and Advanced Traffic Management Systems (ATMS). By using advanced route guidance, these systems can predict real-time travel times for various road segments, considering current conditions, congestion, and other influencing factors [7, 9-12]. The value of travel time estimation is evident in its precision and the significant benefits it offers. Accurate estimates help inform users of changing road conditions, optimize their trips, and reduce pollutant emissions. Travel time predictions are categorized into three key horizons: short-term, medium-term, and long-term. Short-term predictions, which focus on a timeframe of minutes to an hour, are beneficial for real-time navigation and for avoiding immediate congestion. Medium-term predictions, covering hours to a day, support daily planning. Long-term predictions, spanning days to weeks or even months, are valuable for strategic initiatives such as infrastructure development and freight scheduling. By leveraging a variety of features and advanced modeling techniques, including deep learning and other data-driven approaches, transportation systems can continuously improve the accuracy of their predictions across all time horizons.

In recent years, trajectory data in transportation, particularly in urban environments, has expanded significantly due to the ongoing development and widespread adoption of location-aware sensing systems such as GPS, Wi-Fi, RFID, and Bluetooth [11, 12]. These technologies provide valuable geographic coordinates or relative positions of moving elements in transportation systems, facilitating a variety of context-specific services. Examples include location-based alerts, business asset tracking, and participatory sensing for environmental monitoring. This wealth of information empowers transportation planners to make informed, data-driven decisions. By predicting travel times, traffic flows, and other critical factors within transportation networks, planners can enhance overall system efficiency. However, despite these advancements, challenges remain, including rising air pollution, increasing transportation costs, greater fuel consumption, higher accident rates, and deteriorating public health. This context highlights the importance of continued investigation into traffic and road data. A key focus of this effort is travel time prediction, which is essential for addressing sustainability issues, detecting accidents, managing traffic and congestion, and improving dynamic navigation. Moreover, travel time estimation serves as a foundational input for various associated analyses.

While predicting travel time is clearly significant, it poses a complex challenge for transportation planners and other stakeholders, given the numerous dynamic factors at play. These factors include both discrete and continuous speed functions, road conditions, traffic congestion (volume and speed), time of day, day of the week, weather conditions, special events, road network characteristics (type and layout), route preferences, and driver behavior. By identifying and addressing these variables, we can enhance travel-time prediction efforts and contribute to more sustainable, efficient transportation systems.

Discrete and continuous speed functions are two distinct approaches used by the transportation systems analyst for time-dependent travel time estimation. Discrete speed functions divide time into predefined intervals and assume a constant speed within each interval. This simplifies the calculations but may overlook the finer variations in speed. On the other hand, continuous speed functions consider speed as a continuous variable, allowing for more precise estimations by capturing the dynamic nature of speed changes over time. Trigonometric functions or other continuous functions are commonly employed to model these variations accurately.

The following example can illustrate the time-dependent travel time calculation. Let's consider a delivery vehicle traveling from Node A to Node B, a distance of 50 kilometers (kms). Using a discrete speed function with two time intervals due to the rush hour, the vehicle maintains a speed of 60 km/h for the first 30 minutes and then reduces to 40 km/h. The estimated travel time would be 60 minutes, which is the first 30 minutes plus the time to travel the remain distance of 20 km.

When considering time-dependent travel time in the transportation problem, using discrete or continuous speed levels to estimate travel times has limitations. Using discrete speed levels, such as predefined speed categories (e.g., low, medium, high), may not capture the nuanced variations in travel times. Traffic conditions can change rapidly, and discrete speed levels may not accurately reflect the speeds experienced at different times of day or on specific road segments.

On the other hand, using continuous speed levels, where speed is treated as a variable, provides greater flexibility than discrete levels. While continuous speed models offer this advantage, they pose unique challenges for accurately estimating travel times. The complexity arises from various factors, including time of day, traffic congestion, and road conditions, which all influence travel time variations. To effectively utilize continuous speed models, it's essential to account for these multiple variables, which adds a layer of complexity to the estimation process. Additionally, it's essential to recognize that these models may face difficulties in generalizing across different scenarios or in capturing non-linear variations in speed accurately. They also necessitate specific assumptions and parameter choices, which can affect their reliability. Aspects such as traffic lights, turns, and speed limits significantly impact travel times and warrant consideration for more precise estimations.

While traditional methods, including statistical approaches and conventional machine learning techniques (such as historical averages, time series models, and regression analysis), provide foundational time estimates, they often lack the detail and accuracy required in complex real-world scenarios with dynamic environments. In this respect, with advancements in artificial intelligence, machine learning, computational techniques, and big data technology, deep neural networks are emerging as powerful tools. They offer researchers and practitioners innovative solutions for accurately predicting travel times, paving the way for more reliable and comprehensive travel time assessments[10].

Given the complexities of predicting travel times under varying spatial and temporal traffic conditions, this paper proposes a deep learning algorithm to enhance travel time prediction. Our objective is to achieve more accurate predictions through the innovative integration of Temporal

Convolutional Networks (TCN) and Long Short-Term Memory (LSTM) architectures, which effectively capture both short- and long-term temporal patterns. This research considers external factors such as time of day, time of week, and driver behavior, which can significantly influence travel time. The deep learning algorithms analyse the dataset to estimate travel time by learning complex patterns from large, diverse datasets, which enables more accurate predictions than traditional methods. To further improve our predictions, we have implemented an attention mechanism and a multitask learning module, allowing us to analyze travel times for individual road segments as well as for entire routes simultaneously. Moreover, we aim to validate our proposed method by comparing its accuracy, as indicated by the mean absolute percent error (MAPE), with existing models that utilize large-scale real-world taxi datasets from Chengdu found in the literature. This approach not only highlights the effectiveness of our algorithm but also contributes valuable insights to the field of travel time prediction. Attention mechanisms in deep learning play a crucial role in enhancing models' ability to capture complex information and focus on the most relevant elements of the input data. This capability significantly improves accuracy, particularly for long sequences [5, 13]. When applied to travel time prediction, these mechanisms allow the model to emphasize certain historical data points (like speed, time of day, or road conditions) that are most pertinent at any given moment. As a result, this focused approach leads to more precise predictions. It is therefore beneficial to strategically assign attention weights to specific data within the model to maximize its effectiveness.

Deep learning models are a type of artificial neural network that aim to learn and extract complex patterns and representations from data. The purpose of deep learning models is to solve tasks such as classification, regression, and prediction by automatically learning and adapting to the underlying patterns in the data. A simplified overview of the steps involved in training a basic neural network, which is the simplest deep learning model, is shown in Algorithm 1.

Algorithm 1: Basic Neural Network Algorithm

Input: data collection and preprocessing (input data, clean and normalize the data, split the dataset into training, validation, and testing sets)

Output: neural network model

1. Initialize parameters: the weights and biases are initialized randomly or using specific initialization techniques, and input data is fed into the input layer of the network.
2. Define the architecture of the neural network, including the number of layers and the number of neurons in each layer. Each neuron in the subsequent layers calculates a weighted sum of its inputs from the previous layer, adds its bias, and the activation functions to be used.
3. Set the learning rate, set the number of iterations or epochs, representing the number of times the entire training dataset will be used for training.
4. For each iteration in the training process:
   a) Perform forward propagation
   b) Calculate the loss or cost function
   c) Perform backward propagation
   d) Update the parameter (the weights and biases) using the gradients and learning rate.
5. Repeat steps 4a-4d until convergence is achieved or the desired accuracy is reached.
6. Evaluate the network’s performance on a separate dataset
7. The trained neural network model is ready for inference or making predictions on new, unseen data.

The provided pseudo-algorithm is for a basic neural network, a fundamental component of deep learning models. Deep learning models encompass a range of architectures, including convolutional neural networks (CNNs) for image processing, recurrent neural networks (RNNs) for sequential data, and transformers for natural language processing. Although the specific implementation details may vary across different deep learning architectures, the general concept of training through forward and backward propagation remains consistent.

While there has been a growing body of research on predicting travel time in transportation systems, significant gaps remain in the literature. For example, few researchers have integrated dynamic driver behaviours into their models. Most studies tend to consider only one or two types of dependencies that influence travel time, and many rely on stationary datasets that do not use GPS [10, 14, 62]. To address these gaps, this study aims to incorporate driver behaviour, along with other dynamic factors, into the travel time prediction model. Additionally, we will examine both short-term and long-term temporal and spatial dependencies, utilizing GPS datasets to provide a more comprehensive analysis. The contributions of this work can be summarized as follows:

1. We present an innovative deep learning algorithm designed to predict travel time by analyzing trajectory data, which encompasses latitude, longitude, driver ID, time of day, time of week, and driver behavior.
2. This travel time prediction model captures two essential types of dependencies: temporal and spatial dependencies, while considering both short-term and long-term effects.
3. To enhance prediction accuracy beyond that of existing state-of-the-art models in the literature, we integrate advanced techniques such as Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and Temporal Convolutional Networks (TCNs) in the proposed model.
4. Furthermore, we conduct comprehensive numerical experiments utilizing real-world ride-hailing GPS trajectory data from floating vehicles, specifically taxis, thereby ensuring the practicality and relevance of our model.

The remainder of this paper is organized as follows: Section 2 presents a literature review on travel time prediction and the various methods involved. Section 3 formally describes our problem, while Section 4 provides a numerical analysis, including an explanation of the datasets used and the results obtained. Finally, Section 5 concludes the paper and suggests directions for future research.

2. Literature review

2.1 *Travel time estimation*

Travel time estimation has been studied extensively and can be divided into two groups based on the input query: route-based and origin-destination based. In other words, the input query can be a trajectory (sequence of locations) or only two points (origin and destination locations). There are two approaches to estimating the travel time of a route: the road segment and the entire path method. The road segment method calculates travel time by estimating the time for each individual segment of a route, while path method considers the total journey as a single unit to estimate overall travel time, often ignoring intermediate stops or turns in favor of travel time between start and end points. The primary difference is the level of detail: segment-based methods offer granular detail on each part of the trip, whereas path-based methods provide a broader estimate for the entire journey [15, 16].

In the road segment-based prediction, every route is divided into several road segments, and the goal is to calculate the travel time of each road segment. Then, the travel time is simply the sum of the estimated travel times for the different segments [17]. As the correlation between road segments affects the travel time of a path, some papers considered the relationships among adjacent road segments using Hidden Markov Model [18] or Predictive Regression Tree (PR-Tree) and Spatial-Temporal Probabilistic Graphical Model (STPGM) [19]. Wang et al. [20] implemented an error-feedback recurrent convolutional neural network (eRCNN) to accurately estimate the traffic speed of each road segment, using spatiotemporal information of neighboring road segments as input. The delay time at intersections is calculated using an interpolation method, a joint probability model, or a dynamic Bayesian network to concatenate road segment travel times more accurately [21, 22]. Jenelius & Koutsopoulos [23] divided the travel time of a route into two parts: the individual travel time of segments and delay time due to intersections, traffic signals, turns, etc. Although these studies considered the time spent on the connection of different road segments, their main focus is the accurate estimation of individual road segment travel time or speed. Furthermore, another issue of road segment-based prediction is that the travel time error of the whole path can acquire a large number after summing up individual errors if the path consists of many road segments [20].

According to the weakness of the individual road segment-based TTE method, some researchers predicted the travel time of a route by mining historical data and calculating the average travel time of extracted frequent patterns [24, 25, 26]. This method also suffers from two issues: first, the historical average-based estimation may not be very accurate. Second, the historical data can not definitely include any or sufficient information for the searched path. Sometimes, there is no trajectory passing the entire given path. This is called a data sparsity problem. To enhance the path-based TTE model and low sampling rate problem, Wang et al. [27] proposed a model called PTTE. They combined frequent sub-path travel times while optimizing the trade-off between the length of a sub-path and the number of historical trajectory data traveling it (i.e., support). Yuan et al. [28] studied data sparseness and coverage using landmark graph. A landmark is defined as the top-k frequently traversed road segment based on historical data. They estimate the travel time between two landmarks whenever the historical data for each road segment is insufficient. However, this method can not be used for solving the data sparsity issue of roads with few traveled data, since the landmarks are chosen from frequently traveled roads [20].

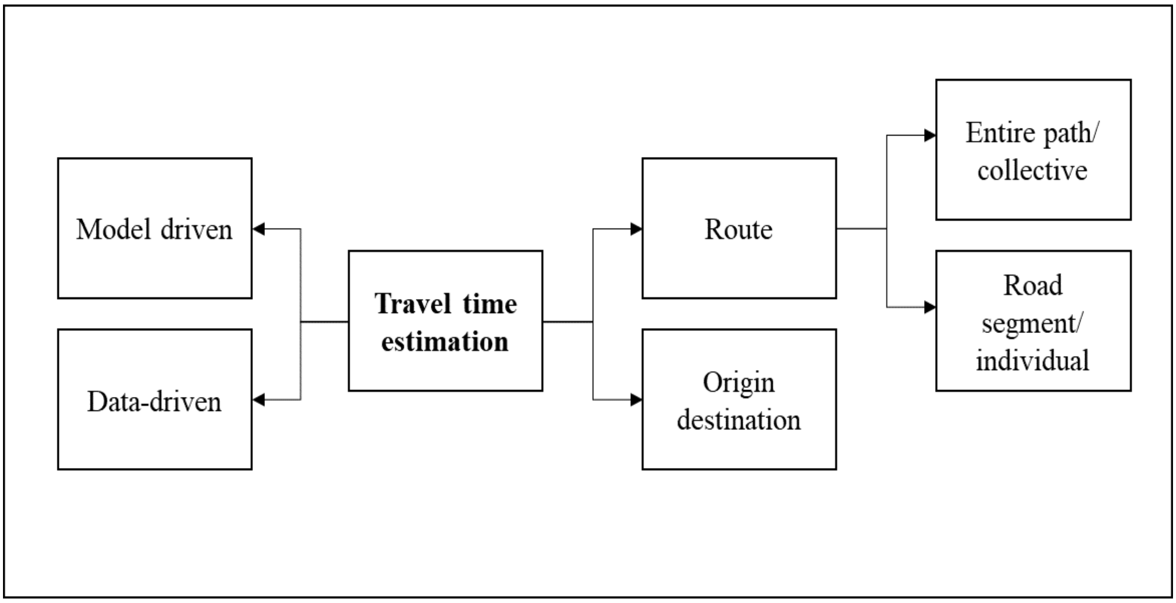

Figure 1: The classification of travel time estimation research

2.2 *Travel time estimation models*

Travel time estimation models can be further classified into two types: model-based and data-driven [29]. The classification of travel time estimation papers is shown in Figure 1. The model-based methods are built on a set of assumptions about the underlying relationship between the input and output variables, like queuing theory [30, 31] and the Cell Transmission Model [32], while data-driven methods rely solely on the input data to make predictions. The data-driven methods are categorized into three groups: statistical methods, basic machine learning methods, and deep learning methods.

Statistical models apply mathematical models and statistical assumptions for the prediction of traffic conditions, such as ARIMA [33, 34], Linear Regression [35, 36], Gaussian process [37, 38], Gaussian mixture regression [39], hidden Markov model [40, 41], Bayesian network [42], Kalman filter [43, 44]. However, the limitation of statistical models including difficulty in handling large complex datasets and the non-linearity of spatial-temporal correlation features in traffic data motivated researchers to explore the machine learning models such as the k-nearest neighbor algorithm (KNN) [45], support vector machine (SVM) [46, 47], and artificial neural network (ANN) to predict travel time

[37]. Lartey et al. [48] developed a comprehensive data-driven mechanism for travel time prediction that synergistically combines support vector machines (SVMs) and autoregressive integrated moving-average (ARIMA) models. This approach capitalizes on the strengths of both techniques to effectively capture the nuances of traffic patterns. The utilization of data from a microsimulation platform further validates the effectiveness of their model. Sheng et al. [14] proposed a forward-looking deep learning spatial-temporal model for predicting travel times, integrating trajectory data and traffic conditions via traffic-feature fusion. This innovative strategy highlights the value of merging diverse data sources for improved travel time forecasting.

Although these models estimate the traffic conditions more accurately and are relatively suitable for more complex data, they cannot deal with the nonlinear correlation problem. The advancement of deep learning models gives researchers the opportunity to apply deep learning-based methodologies for time series data processing. Specifically, deep learning is more frequently being used in traffic prediction tasks due to the impact of feature extraction on prediction accuracy and the power of deep learning models in extracting the spatiotemporal correlation characteristics. Some recent deep learning models are convolutional neural network (CNN) for feature extraction, recurrent neural network (RNN) and its variants, including long short-term memory neural network (LSTM) [49], and gate recurrent unit neural network (GRU) for processing temporal information, graph convolution neural network (GCN) for feature extraction of non-Euclidean structured data, attention mechanism for capturing long time span features, etc. For example, Zhang et al. [50] designed a self-attention mechanism enhanced CNN for the long sequence time-series prediction. Researchers also integrate different deep learning models to achieve higher accuracy in prediction like the DeepTTE model which combines CNN and LSTM [20], the MCT-TTE model, which combines CNN and transformer models [51], and STGNN-TTE, which combines multi-stage spatiotemporal GCN and transformer layers [52]. Chen et al. [9] proposed a deep learning model integrated with bi-directional isometric-gated recurrent unit (BDIGRU) to estimate the travel time for vehicles in the business. Wang et al. [10] predicted the vehicle travel times efficiently by a proposed model including a deep learning framework with graph neural networks (GNN) and RNN. Liu et al. [53] introduced an integrated model to estimate route-specific origin-destination travel times for vehicles. This model uses the advanced machine learning technique of active adversarial inverse reinforcement learning (AA-IRL) alongside a network architecture, the AdaBoost multi-fusion graph convolutional Transformer (AMGC-Transformer). By incorporating dynamic factors such as weather conditions, traffic patterns, time of day, and driver ID, the model enhances the accuracy of travel time predictions. Sun et al. [54] addressed the next challenge in location estimation by developing two deep learning models based on sequential and hybrid long short-term memory (LSTM) networks. Their approach illustrates the potential of leveraging deep learning in traffic forecasting. Similarly, Luo et al. [55] developed a robust deep learning-based framework for predicting vehicle travel times, accounting for various vehicle types and the proximity of locations within the transportation network. By integrating a CNN with spatial-temporal attention, their framework excels at analysing spatial patterns within the datasets, demonstrating its capability in handling complex travel scenarios.

The literature review highlights the existing approaches for travel time prediction and challenges. To tackle the previously identified difficulties, our research proposes an accurate travel time estimation using a combination of TCN and LSTM to learn and extract both short- and long-term temporal information. The impact of external factors like time ID, week ID, and drivers' habits is also considered in this research. A hybrid TCN-LSTM model presents a superior solution by effectively integrating the strengths of both architectures. The TCN layer excels in extracting multi-scale temporal features and identifying local patterns from the raw input data. Subsequently, the LSTM layer builds on these extracted features to capture complex, long-term dependencies [56, 57]. This synergistic approach offers a robust framework that adeptly navigates the intricate, non-linear, and dynamic nature of traffic data, ultimately enhancing prediction accuracy compared to relying on either model independently.

3. Problem definition

In this study, we aim to estimate travel times for road segments and entire routes using trajectory data. We are interested in predicting how long it will take for a vehicle to travel between consecutive GPS points along a given trajectory. This information is valuable for various applications such as traffic management, route planning, and transportation optimization. To accomplish this, we use trajectory data comprising latitude, longitude, time of day, time of week, and driver ID. Each trajectory is composed of a sequence of GPS points that represent the vehicle's movement over time. We define a road segment as the route between three consecutive GPS points. By considering these road segments, we can divide the trajectory into smaller segments and estimate the travel time for each segment. We use different deep learning algorithms to analyze the data and learn from the patterns. This allows us to estimate travel times for individual road segments and the entire route concurrently.
In this section, we define the input and output of the problem.

3.1 *Input:*

The input of our problem is a trajectory T, which consists of several consecutive GPS points. Each GPS point has latitude, longitude, time of day, time of week, and driver ID elements. The link between three consecutive points is defined as a road segment.

3.2 *Output:*

According to the two methods discussed in the literature review, we predict the travel time for each road segment and the entire route simultaneously. So, the outputs of our model for each trajectory are the estimated travel time for each road segment and the whole route. The entire route's travel time is calculated as a trainable weighted sum of the travel times of the road segments.

3.3 *Model architecture*

Our proposed framework includes three sections: attribute embedding, spatiotemporal learning, and multitask learning section. Figure 2 shows the architecture of our model. The problem input is a trajectory with latitude, longitude, time, and driver ID elements. The road segment is defined as the link between $k = 3$ consecutive GPS points. The output of the model is the predicted travel time for each road segment and the entire route.

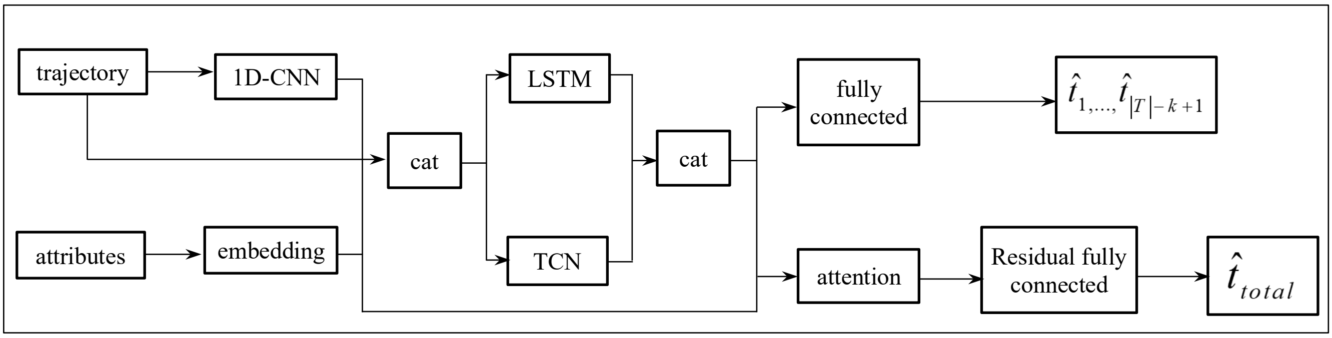

Figure 2: The overview of our proposed model

3.3.1 Attribute embedding

Several factors affect travel time, including time of day, day of the week, driver behavior, and weather. Extracting the pattern of these external factors and considering their impact on travel time using learning algorithms can result in a more accurate estimation. Three external elements are considered in this paper: time ID (time of day: 0 to 1439 min), week ID (time of week: 0 to 6), and driver ID. These factors are categorical attributes that should be converted to numerical vectors using the embedding method.

3.3.2 Spatiotemporal learning

The spatiotemporal learning section aims to learn spatial and temporal correlations of GPS points. In this part, the model learns the special features of GPS data, such as traffic lights, speed limit, turning, changing speed at intersections, etc. Since these conditions are dynamic at different times of a day, we also need to capture temporal dependencies. First, to capture the spatial factors precisely, we apply a non-linear mapping and then a CNN layer [27]. Instead of non-linear Geo-mapping of raw GPS data, some research mapped the GPS coordinates to a two-dimensional grid map and then the features of the grid map are extracted as images [19, 58]. However, the spatial information is not accurately represented by the direct mapping of GPS points to the grid cells. So, we apply a non-linear function to map the GPS point to the *traj* vector based on the following:

$$traj_i = \tanh(W_{traj}.(p_i.lat \circ p_i.lng)) \quad \textbf{(1)}$$

where p_i indicates i^{th} GPS point in the trajectory $T = \{P_1, \ldots, P_{|T|}\}$, the $\circ$ indicates the concatenation operation and W_{traj} is a learnable matrix with a size of $16 * |T|$. $|T|$ is the number of all GPS points in the given trajectory.

A usual approach to capture spatial dependencies is the convolutional neural network (CNN), which is mostly used for image or object processing. Thus, the output of non-linear mapping ($traj_i \in R^{16*|T|}$) is given to 1D-CNN with a kernel size of 3 (k =3) to create the feature map in Equation (2). It is worth mentioning that such input can be considered as a 16-channel input to the Conv layer. The output of the CNN module is a vector with a size of $16 * (|T| - k + 1)$ based on the CNN structure.

$$loc_i = Conv1D(traj_i) \quad \textbf{(2)}$$

As the total distance and other embedded external attributes significantly influence the travel time, they are concatenated to the output of the CNN module, shown in Equation (3). Furthermore, the input of the CNN layer is concatenated with its output through skip connections to keep important information from the input, enabling the network to better learn the relationships between the input and output.

$$loc_f = (loc_i \circ dist \circ attr) \circ traj_i \quad \textbf{(3)}$$

To capture the temporal dependencies of GPS points, two layers of LSTM and a temporal convolutional network are implemented. LSTM is a variant of RNNs, which have three gates, including input, forget, and output gates. These gates can help the model to memorize important information and forget unimportant ones, allowing the network to capture long-term dependencies. On the other hand, a temporal convolutional network (TCN) is a type of CNN for processing sequential data. It is suitable for modeling complex temporal patterns in the data since it can capture local and short-term dependencies in the data and learn hierarchical representations of the data. By applying both TCN and LSTM, the model can effectively benefit from the strengths of both methods, which result in a more complete and precise depiction of the time-related information. Therefore, combining these two methods can enhance the accuracy of travel time estimation. After these temporal learning layers, we have the sequence of hidden states at different time steps ($\{h_1, \ldots, h_{|T|-k+1}\}$), where

$$h_i = LSTM(W_h.h_{i-1} + W_c.loc_f) \circ TCN_i(loc_f) \quad \textbf{(4)}$$

3.3.3 Multitask learning

The multitask learning part uses the spatiotemporal learning section's output $\{h_1, \ldots, h_{T-k+1}\}$ to predict the travel time for each road segment and the entire route. The segment travel time ($\hat{t}_i$) is predicted through fully connected layers, while the attention layer and residual fully connected layers are used to predict the entire path travel time ($\hat{t}_{total}$). The attention layer is a way to sum all segments' travel time with different weights based on their impact on the entire path. To calculate the weights of hidden states, the attention mechanism considers the external attributes and the extracted spatiotemporal features of each segment. This study employs an attention mechanism inspired by the successful work in [20], effectively addressing multi-task learning. By learning from previous layers in the model, it adjusts weights to improve overall task performance.

4. Numerical Analysis for Travel Time Prediction

4.1 Data

To compare the results of our travel time prediction model and several existing methods, one large-scale dataset, the Chengdu dataset (the same as the study [20]), is selected in this research. This dataset is common in many papers based on the literature review, which helps to make a valid comparison. The Chengdu Dataset consists of 9,737,557 trajectories (1.4 billion GPS records) of 14,864 taxis in August 2014 in Chengdu, China. However, limited computation resources make it challenging to use all raw trajectory data. We sampled 1,150,000 trajectories to test the performance of different algorithms and compare them. Before a prediction model, it is essential to preprocess raw data, as improper data records, such as outliers and missing values, can negatively affect the accuracy of the model [59]. The raw dataset contains driver ID, latitude, longitude, states, and timestamp columns, which are illustrated in Table 1. The input to the prediction model is GPS data in the form of trajectories. The processing of the data involves converting GPS points into meaningful trajectories and omitting outliers based on time and distance difference of consecutive points, and traffic-controlling criteria. First, the GPS data was sorted based on time, and the time and distance difference of each consecutive point was calculated. Then, if the time and distance difference of two consecutive points exceeds 1800 seconds or 1 km, or the driver ID is not the same, these two consecutive points are assigned to two different trajectories.
Second, some traffic-controlling criteria were defined and applied to create more meaningful trajectories and remove outliers. These criteria are set based on typical characteristics of an urban trajectory. The following are some of the criteria: the travel distance greater than 100km or less than 0.5km, the average speed greater than 120km/h or less than 5km/h, and the travel time greater than 7200 seconds or less than 60 seconds [51]. In the end, the remaining trajectories are randomly partitioned (according to a Uniform/Normal distribution) to create polylines with lengths of 11 to 128 points.

Table 1: Raw data of the Chengdu dataset [20].

Driver ID	Latitude	Longitude	State	Time
1	*30.624806*	*104.136604*	*1*	*2014-08-03 9:18:46 PM*
1	*30.624809*	*104.136612*	*1*	*2014-08-03 9:18:15 PM*
1	*30.624811*	*104.136587*	*1*	*2014-08-03 9:20:17 PM*
1	*30.624811*	*104.136596*	*1*	*2014-08-03 9:19:16 PM*
1	*30.624811*	*104.136619*	*1*	*2014-08-03 9:17:44 PM*
1	*30.624813*	*104.136589*	*1*	*2014-08-03 9:19:46 PM*
1	*30.624815*	*104.136585*	*1*	*2014-08-03 9:21:18 PM*
1	*30.624815*	*104.136587*	*1*	*2014-08-03 9:20:48 PM*
1	*30.624815*	*104.136639*	*1*	*2014-08-03 9:17:14 PM*
1	*30.624816*	*104.136569*	*1*	*2014-08-03 9:22:50 PM*
1	*30.624816*	*104.136574*	*1*	*2014-08-03 9:22:19 PM*

4.2 Parameter setting

In learning algorithms, data should be shuffled into three sets: train, development, and test. Training data is used for learning the model parameters. The development set is for checking the accuracy of different models and comparing them to find the best one, while the test set is unseen data for the best model to evaluate its performance. Considering the huge amount of data (more than 1 million), the dataset is divided into a 90% training set, 5% dev set, and 5% test set. Other parameters for the experiment are shown in Table 2.

Table 2: Detailed parameter setting for TTE

Section	**Parameter**	**Value**
Embedding	Week ID dimension	R^4
	Time ID dimension	R^9
	Driver ID dimension	R^{17}
Spatial_CNN	Number of filters	48
	Activation function	Elu
Temporal_LSTM	Hidden size	128
	Number of layers	2
Temporal_TCN	Activation function	Tanh and sigmoid
	Number of layers	2
Multitask learning_entire route	Activation function	Leaky relu
	Hidden size	64,64,64 with residual connection
Multitask learning_road segment	Activation function	Leaky relu
	Hidden size	64,32

The optimization algorithm for training parameters is Adam, and the learning rate is scheduled based on an exponential decay formula with an initial learning rate of 0.001 and a decay rate of 0.98. The batch size during training is 100, and we train the model for 60 epochs and the best model is selected based on validation loss. The training and evaluation process is conducted on a workstation with a GPU (NVIDIA TITAN V) with 64 GB of available RAM. Adam, which stands for Adaptive Moment Estimation, is a widely used and efficient optimization algorithm for training deep learning models [60, 61]. It works by adaptively adjusting the learning rate for each parameter of the algorithm based on the first and second moments of the gradients. This approach combines the advantages of momentum with adaptive learning rates, helping models converge faster and more smoothly.

4.3 Results and discussion

The objective function to train different models is set to minimize mean absolute percentage error (MAPE), a widely recognized relative loss metric. The MAPE is suitable for both long and short paths. MAPE is commonly used to assess the accuracy of travel time prediction models [45], and in line with this established practice, we have chosen MAPE as our key evaluation metric. Our model utilizes a loss function that incorporates a weighted sum of losses from individual road segments as well as the entire route, enhancing its accuracy. To evaluate the performance of our proposed model, we conducted a comparison with four other models, which allows us to highlight its strengths and identify areas for improvement. The models included in our comparison are as follows:

- DeepTTE: which uses a Geo-conv layer and LSTM to capture spatial and temporal dependencies [20].

- RNNTTE: a streamlined version of DeepTTE, which uses a basic RNN instead of LSTM [20].
- Transformer: this has the same attribute and multitask learning sections as the proposed model but utilizes a 6-layer encoder transformer with 4 heads to capture temporal information [51].
- GCN-2TCN: which uses a graph convolutional network instead of 1D-CNN to capture spatial correlations and only two layers of TCN for temporal learning [52].

Table 3: Performance comparison of different models

Model	MAPE (%)
DeepTTE	8.714
RNNTTE	8.899
Transformer	11.093
GCN-2TCN	9.859
Our algorithm	***5.787***

Our algorithm demonstrated a higher accuracy, achieving a MAPE of 5.787%, which is better than the other algorithms, as indicated by the MAPE results in Table 3 and illustrated in Figure 3. The key factor for predicting travel time lies in learning spatiotemporal features, which we accomplished using CNN, LSTM, and TCN. The comparison of various algorithms with different structures for extracting spatiotemporal correlations highlights the importance of this module in making accurate predictions. By leveraging LSTM for short-term dependencies and TCN for long-term dependencies, we observed a significant improvement in estimation accuracy.

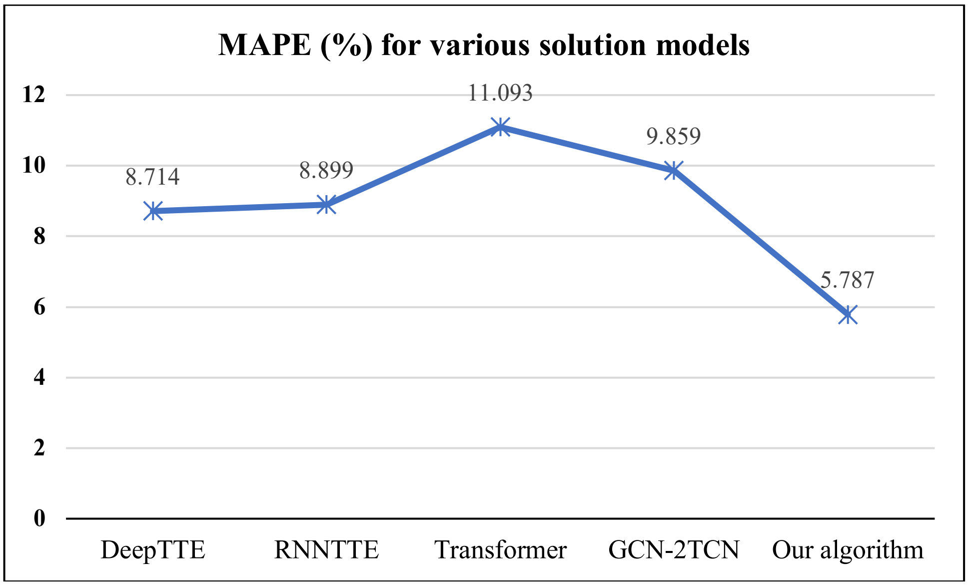

Figure 3: Performance comparison of different solution models

5. Conclusions

This research presents a promising approach to overcoming the challenges of realistic and dynamic transportation systems by enhancing vehicle travel time predictions through the use of an advanced deep learning algorithm. By leveraging the strengths of Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and Temporal Convolutional Networks (TCNs), we have developed an algorithm that shows considerable improvement over existing methods for predicting vehicle travel times. The integration of CNNs allows for effective extraction of spatial features from trajectory data, while LSTMs are particularly effective in capturing long-term dependencies and patterns within sequential data. Additionally, TCNs offer a powerful convolution-based method that combines temporal convolutions and dilation to glean intricate temporal patterns. This unique combination allows our model to thoroughly understand both the spatial structure and the temporal dynamics of complex datasets, significantly enhancing its performance and application potential. In our travel time prediction model, we define a road segment as the route between three consecutive GPS points. To better reflect realistic driving conditions, instead of considering the Euclidean distance, we include comprehensive trajectory data such as latitude, longitude, time of day, time of week, driving habits, and driver ID for each segment in the model. We have validated our proposed model through a comparative analysis against existing models, utilizing large-scale, real-world data from Chengdu taxi datasets and measuring performance using the Mean Absolute Percent Error (MAPE). Our findings indicate a notable improvement of 2.9% in travel time prediction accuracy over other state-of-the-art models highlighted in the literature.

This work presents two key contributions that have the potential to substantially enhance our understanding of transportation systems. Firstly, by integrating deep learning networks for vehicle travel time prediction, this study accounts for real traffic conditions and speed flow. This thoughtful approach results in a more accurate and realistic portrayal of travel times, which can significantly improve transportation problem optimization. Secondly, the development of an innovative deep learning model using a comprehensive real-world GPS dataset has shown remarkable performance, achieving greater accuracy in travel time estimation compared to existing models. This breakthrough opens the door to creating more efficient transportation systems that can adapt to the dynamic nature of travel conditions. In summary, this research offers valuable insights into employing advanced artificial intelligence and data-driven strategies to predict vehicle travel times effectively. These advancements have the potential to substantially improve transportation systems, enabling them to respond adeptly to the constantly changing nature of road transport. The broader implications of this work include enhanced efficiency, reduced operational costs, and a significant improvement in the quality of life within urban areas.

This research can be further extended in several important ways. In our current study, we primarily focused on using a single route generated between each pair of nodes via the Google Direction API to estimate travel times. However, in real-world scenarios, multiple routes exist between an origin and a destination at different departure times, which results in varying travel durations. Therefore, vehicle departure time could be considered as a third dimension in the model, alongside origin and destination, for generating various routes and selecting the best one for minimum travel time in future research. Additionally, future research would greatly benefit from incorporating more factors, including weather data, vehicle counts over specific distances, traffic congestion and its real-time fluctuations, driving behaviour, road construction activities, and large public gatherings into the analysis. In future work, we could also utilize several other datasets to validate the proposed model. By integrating these various elements, we can enhance the accuracy and realism of travel time estimations, leading to more precise system optimizations that respond to changing road and traffic conditions. By continually advancing the integration of artificial intelligence, data-driven methodologies, and real-time data utilization, researchers can unlock greater potential for optimizing transportation systems and developing sustainable and efficient logistics networks.

Acknowledgement

This manuscript is based on Zahra Ejabati Emanab's master's thesis, supervised by Dr. Yuvraj Gajpal. We greatly appreciate the financial support provided by the Asper School of Business in conducting this research.

References

1. Zhang, M., Li, Z., Si, H., Cheng, L., Zhou, X., & Wang, B. (2023). Urban travel time and residential location choice: The impacts of traffic congestion. Sustainable Cities and Society, 99, 104975.
2. Bouzouina, L., Baraklianos, I., Bonnel, P., & Aissaoui, H. (2021). Renters vs owners: The impact of accessibility on residential location choice. Evidence from Lyon urban area, France (1999–2013). Transport Policy, 109, 72-84.
3. Caulfield, B., & Charly, A. (2022). Examining the potential environmental and travel time saved benefits of remote working hubs. Transport Policy, 127, 139-147.
4. Sun, Y. Y., & Lin, Z.W. (2017). Move fast, travel slow: the influence of high-speed rail on tourism in Taiwan. Journal of Sustainable Tourism, 26(3), 433–450.
5. Wang, Y., Tian, Z., & Fu, H. (2024). Multivariate USV motion prediction method based on a temporal attention weighted TCN-Bi-LSTM model. Journal of Marine Science and Engineering, 12(5), 711.
6. Groß, P. O., Ulmer, M. W., Ehmke, J. F., & Mattfeld, D.C. (2015). Exploiting travel time information for reliable routing in city logistics. Transportation Research Procedia, 10, 652-661.
7. Kandiri, A., Ghiasi, R., Nogal, M., & Teixeira, R. (2024). Travel time prediction for an intelligent transportation system based on a data-driven feature selection method considering temporal correlation. Transportation Engineering, 18, 100272.
8. Guo, J., Wang, W., Guo, J., D'Ariano, A., Bosi, T., & Zhang, Y. (2024). An instance-based transfer learning model with attention mechanism for freight train travel time prediction in the China–Europe railway express. Expert Systems with Applications, 251, 123989.
9. Chen, C.Y.T., Sun, E.W., & Lin, Y-B. (2025). Reconciling spatiotemporal conjunction with digital twin for sequential travel time prediction and intelligent routing. Annals of Operations Research, 348, 671–716.
10. Wang, D., Zhu, J., & Yin, Y. (2024). Dynamic travel time prediction with spatiotemporal features: using a GNN-based deep learning method. Annals of Operations Research, 340, 571–591.
11. Kim, Y., Tak, H-Y., Kim, S., & Yeo, H. (2024). A hybrid approach of traffic simulation and machine learning techniques for enhancing real-time traffic prediction. Transportation Research Part C: Emerging Technologies, 160, 104490.
12. Al-Ghobari, M., Muneer, A., & Fati, S. M. (2021). Location-aware personalized traveler recommender system (LAPTA) using collaborative filtering KNN. Computers, Materials & Continua, 69(2).
13. Shih, S.Y.; Sun, F.K., & Lee, H.Y. (2019). Temporal pattern attention for multivariate time series forecasting. Machine Learning, 108, 1421–1441.
14. Sheng, Z., Lv, Z., Li, J., Xu, Z., Sun, H., Liu, X., & Ye, R. (2023). Taxi travel time prediction based on fusion of traffic condition features. Computers and Electrical Engineering, 105, 108530.
15. Mansurova, A., Mussina, A., Aubakirov, S., Nugumanova, A., & Yedilkhan, D. (2025). From raw GPS to GTFS: a real-world open dataset for bus travel time prediction. Data, 10(8), 1-16.
16. Al-Naim, R., & Lytkin, Y. (2021). Review and comparison of prediction algorithms for the estimated time of arrival using geospatial transportation data. Procedia Computer Science, 193, 13-21.
17. Liu, H., Xu, H., Yan, Y., Cai, Z., Sun, T., & Li, W. (2020). Bus arrival time prediction based on LSTM and spatial-temporal feature vector. IEEE Access 2020, 8, 11917–11929.
18. Yang, B., Guo, C., & Jensen, C. S. (2013). Travel cost inference from sparse, spatio temporally correlated time series using Markov models. Proceedings of the VLDB Endowment, 6(9), 769-780.
19. Wang, D., Cao, W., Xu, M., & Li, J. (2016). ETCPS: An effective and scalable traffic condition prediction system. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 9643.

20. Wang, D., Zhang, J., Cao, W., Li, J., & Zheng, Y. (2018). When will you arrive? Estimating travel time based on deep neural networks. 32nd AAAI Conference on Artificial Intelligence, AAAI 2018.
21. Hofleitner, A., Herring, R., Abbeel, P., & Bayen, A. (2012). Learning the dynamics of arterial traffic from probe data using a dynamic Bayesian network. IEEE Transactions on Intelligent Transportation Systems, 13(4), 1679-1693.
22. Hofleitner, A., & Bayen, A. (2011). Optimal decomposition of travel times measured by probe vehicles using a statistical traffic flow model. In 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC) (pp. 815-821), IEEE.
23. Jenelius, E., & Koutsopoulos, H. N. (2013). Travel time estimation for urban road networks using low frequency probe vehicle data. Transportation Research Part B: Methodological, 53, 64-81.
24. Kwesiga, D.K., Guin, A., & Hunter, M. (2025). Analysis of bus dwell times from automated passenger count data and the impact of dwell-time variability on the performance of transit signal priority. Public Transport. 1–23.
25. Yin, Z., & Zhang, B. (2023). Bus travel time prediction based on the similarity in drivers' driving styles. Future Internet, 15, 222.
26. Rahmani, M., Jenelius, E., & Koutsopoulos, H. N. (2013). Route travel time estimation using low-frequency floating car data. In 16th international IEEE conference on intelligent transportation systems, 2292-2297, IEEE.
27. Wang, Z., Fu, K., & 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.
28. Yuan, J., Zheng, Y., Xie, X., & Sun, G. (2011). T-drive: Enhancing driving directions with taxi drivers' intelligence. IEEE Transactions on Knowledge and Data Engineering, 25(1), 220-232.
29. Bai, M., Lin, Y., Ma, M., & Wang, P. (2018). Travel-time prediction methods: a review. In International Conference on Smart Computing and Communication (pp. 67-77). Cham: Springer International Publishing.
30. Shu, W., Cai, K., & Xiong, N.N. (2022). A short-term traffic flow prediction model based on an improved gate recurrent unit neural network. IEEE Transactions on Intelligent Transportation Systems, 23(9), 16654-16665.
31. Ben-Akiva, M., Bierlaire, M., Burton, D., Koutsopoulos, H. N., & Mishalani, R. (2001). Network state estimation and prediction for real-time traffic management. Networks and spatial economics, 1(3), 293-318.
32. Wan, N., Gomes, G., Vahidi, A., & Horowitz, R. (2014). Prediction on travel-time distribution for freeways using online expectation maximization algorithm. In Transportation Research Board 93rd Annual Meeting (No. 14-3221).
33. Zhang, Y., Zhang, Y., & Haghani, A. (2014). A hybrid short-term traffic flow forecasting method based on spectral analysis and statistical volatility model. Transportation Research Part C: Emerging Technologies, 43, 65–78.
34. Xu, F., Lin, Y., Huang, J., Wu, D., Shi, H., Song, J., & Li, Y. (2016). Big data driven mobile traffic understanding and forecasting: a time series approach. IEEE Transactions on Services Computing, 9(5), 796–805.
35. Kwon, J., Coifman, B., & Bickel, P. (2000). Day-to-day travel-time trends and travel-time prediction from loop-detector data. Transportation Research Record, 1717.
36. Sun, H., Liu, H. X., Xiao, H., He, R. R., & Ran, B. (2003). Use of local linear regression model for short-term traffic forecasting. Transportation Research Record, 1836.
37. Xu, Y., Yin, F., Xu, W., Lin, J., & Cui, S. (2019). Wireless traffic prediction with scalable gaussian process: framework, algorithms, and verification. IEEE Journal on Selected Areas in Communications, 37(6).
38. Zhao, J., & Sun, S. (2016). High-order gaussian process dynamical models for traffic flow prediction. IEEE Transactions on Intelligent Transportation Systems, 17(7).

39. Qi, G., Ceder, A., Zhang, Z., Guan, W., & Liu, D. (2021). New method for predicting long-term travel time of commercial vehicles to improve policy-making processes. Transportation Research Part A: Policy and Practice,145, 132-152.
40. Luo, Z., Shi, H., Liu, W., & Jin, Y. (2020). HMM-based traffic situation assessment and prediction method. CICTP 2020: Transportation Evolution Impacting Future Mobility - Selected Papers from the 20th COTA International Conference of Transportation Professionals.
41. Zheng, Y., Li, Y., Own, C. M., Meng, Z., & Gao, M. (2018). Real-time prediction and navigation on traffic congestion model with equilibrium Markov chain. International Journal of Distributed Sensor Networks, 14(4).
42. Xu, Y., Kong, Q. J., Klette, R., & Liu, Y. (2014). Accurate and interpretable Bayesian MARS for traffic flow prediction. IEEE Transactions on Intelligent Transportation Systems, 15(6).
43. Xie, Y., Zhang, Y., & Ye, Z. (2007). Short-term traffic volume forecasting using Kalman filter with discrete wavelet decomposition. Computer-Aided Civil and Infrastructure Engineering, 22(5).
44. Xu, T. Dong, Hao, Y., Peng, Z. ren, & Sun, L-J. (2012). Real-time travel time predictor for route guidance consistent with driver behavior. Canadian Journal of Civil Engineering, 39(10).
45. Murni, Kosasih, R., Fahrurozi, A., Handhika, T., Sari, I., & Lestari, D. P. (2020). Travel time estimation for destination in Bali using kNN-regression method with TensorFlow. IOP Conference Series: Materials Science and Engineering, 854(1).
46. Chen, X. M., Gong, H. B., & Wang, J. N. (2012). BRT vehicle travel time prediction based on SVM and Kalman filter. Journal of Transportation Systems Engineering and Information Technology, 12(4), 29-34.
47. Gao, P., Hu, J., Zhou, H., & Zhang, Y. (2016). Travel time prediction with immune genetic algorithm and support vector regression. Proceedings of the World Congress on Intelligent Control and Automation (WCICA), 2016, September.
48. Lartey, B., Zeleke, L., Yan, X., Gupta, K. D., Homaifar, A., & Karimoddini, A. (2022). A data-driven approach for travel time prediction and analysis. In 2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 834-839.
49. Liu, Y., Wang, Y., Yang, X., & Zhang, L. (2018). Short-term travel time prediction by deep learning: A comparison of different LSTM-DNN models. IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, 2018, March.
50. Zhang, L., Cai, Y., Huang, H., Li, A., Yang, L., & Zhou, C. (2022). A CNN-LSTM model for soil organic carbon content prediction with long time series of MODIS-based phenological variables. Remote sensing, 14(18), 4441.
51. Liu, F., Yang, J., Li, M., & Wang, K. (2022). MCT-TTE: travel time estimation based on transformer and convolution neural networks. Scientific Programming, 2022.
52. Jin, G., Wang, M., Zhang, J., Sha, H., & Huang, J. (2022). STGNN-TTE: travel time estimation via spatial–temporal graph neural network. Future Generation Computer Systems, 126.
53. Liu, S., Zhang, Y., Wang, Z., Liu, X., & Yang, H. (2025). Personalized origin–destination travel time estimation with active adversarial inverse reinforcement learning and Transformer. Transportation Research Part E: Logistics and Transportation Review, 193, 103839.
54. Sun, J., & Kim, J. (2021). Joint prediction of next location and travel time from urban vehicle trajectories using long short-term memory neural networks. Transportation Research Part C: Emerging Technologies, 128,103114.
55. Luo, S., Zou, F., Zhang, C., Tian, J., Guo, F., & Liao, L. (2022). Multi-View Travel Time Prediction Based on Electronic Toll Collection Data. Entropy, 24(8), 1050.
56. Liu, S., Du, L., Cao, T., & Zhang, T. (2024). Research on a passenger flow prediction model based on BWO-TCLS-Self-Attention. Electronics, 13(23), 4849.
57. Tian, J., Liu, H., Gan, W., Zhou, Y., Wang, N., & Ma, S. (2025). Short-term electric vehicle charging load forecasting based on TCN-LSTM network with comprehensive similar day identification. Applied Energy, 381, 125174.

58. Liu, Q., Wu, S., Wang, L., & Tan, T. (2016). Predicting the next location: A recurrent model with spatial and temporal contexts. 30th AAAI Conference on Artificial Intelligence, AAAI 2016.
59. Zheng, Y. (2015). Trajectory data mining: An overview. ACM Transactions on Intelligent Systems and Technology, 6(3).
60. Reyad, M., Sarhan, A. M., & Arafa, M. (2023). A modified Adam algorithm for deep neural network optimization. Neural Computing and Applications, 35(23), 17095-17112.
61. Sun, H., Zhou, W., Yang, J., Shao, Y., Xing, L., Zhao, Q., & Zhang, L. (2024). An improved medical image classification algorithm based on Adam optimizer. Mathematics, 12(16), 2509.
62. Shi, C., Zou, W., Wang, Y., Zhu, Z., Chen, T., Zhang, Y., & Wang, N. (2025). Enhancing travel time prediction for intelligent transportation systems: a high-resolution origin–destination-based approach with multi-dimensional features. Sustainability, 17(5), 2111.