Improved Deep Learning-based Approach for Spatial-Temporal Trajectory Planning via Predictive Modeling of Future Location

Zain Ul Abideen[1*], Xiaodong Sun[1], Chao Sun[1], and Hafiz Shafiq Ur Rehman Khalil[2]
[1] Automotive Engineering Research Institue, Jiangsu University, Zhenjiang, 212013 China
[e-mail: xdsun@ujs.edu.cn]
[2] School of electronics and information engineering, Xian Jiaotong University, Xian, 710049, PR China
[*]Corresponding author: Xiaodong Sun

Received November 14, 2023; revised March 20, 2024; accepted May 13, 2024; published July 31, 2024

Abstract

Trajectory planning is vital for autonomous systems like robotics and UAVs, as it determines optimal, safe paths considering physical limitations, environmental factors, and agent interactions. Recent advancements in trajectory planning and future location prediction stem from rapid progress in machine learning and optimization algorithms. In this paper, we proposed a novel framework for Spatial-temporal transformer-based feed-forward neural networks (STTFFNs). From the traffic flow local area point of view, skip-gram model is trained on trajectory data to generate embeddings that capture the high-level features of different trajectories. These embeddings can then be used as input to a transformer-based trajectory planning model, which can generate trajectories for new objects based on the embeddings of similar trajectories in the training data. In the next step, distant regions, we embedded feedforward network is responsible for generating the distant trajectories by taking as input a set of features that represent the object's current state and historical data. One advantage of using feedforward networks for distant trajectory planning is their ability to capture long-term dependencies in the data. In the final step of forecasting for future locations, the encoder and decoder are crucial parts of the proposed technique. Spatial destinations are encoded utilizing location-based social networks (LBSN) based on visiting semantic locations. The model has been specially trained to forecast future locations using precise longitude and latitude values. Following rigorous testing on two real-world datasets, Porto and Manhattan, it was discovered that the model outperformed a prediction accuracy of 8.7% previous state-of-the-art methods.

Keywords: Intelligent Transportation System, trajectory planning, future location, Location-Based Social Networks.

http://doi.org/10.3837/tiis.2024.07.002 **ISSN : 1976-7277**

1. Introduction

Citywide trajectory planning with future location is an important problem in the field of transportation and urban planning. The goal of this problem is to generate optimal trajectories for a fleet of vehicles navigating through a city, aiming to reach specific future locations in the most efficient and timely manner possible. This problem is becoming increasingly important with the rise of shared mobility services such as ride-sharing and autonomous vehicles, which require efficient trajectory planning algorithms to operate effectively. Moreover, citywide trajectory planning is critical for reducing traffic congestion and improving the overall efficiency of transportation systems, which has significant economic and environmental implications [1]. Citywide trajectory planning with future locations involves several challenges, including the need to model complex traffic patterns and optimize trajectories in real-time based on changing traffic and weather conditions. Additionally, it requires the use of advanced machine learning and optimization techniques, as well as a deep understanding of urban transportation systems and traffic engineering principles. There are some significant problems with trajectory planning and future location as follows:

- Certain urban regulations, such as Beijing's driving restriction policy [1], impose limitations on traffic flow. Under this policy, certain drivers are authorized to operate within specified areas. For instance, if a region is designated exclusively for taxi drivers and is a popular destination for passengers, these drivers may be restricted from taking orders in that area. Consequently, this restriction can lead to resource inefficiency.
- Few taxi drivers prioritize passengers travelling to familiar areas for their convenience. Conversely, some drivers are disinclined to accept short-distance trips due to their limited profitability. When most passengers' destinations are either close to the pick-up location or fall outside the driver's regular operating regions, they may decline such requests.
- If a driver is directed to an area where most passengers are heading to unfamiliar destinations, they may experience delays even when relying on GPS navigation. These behaviors among taxi drivers can diminish passenger satisfaction levels and reduce the overall operational efficiency of the taxi market.

Trajectory planning and future location prediction are important research areas in transportation and autonomous vehicle systems. Recent studies have explored various approaches to trajectory planning and future location prediction using machine learning and other techniques. In A Survey on Urban Trajectory Prediction [2] provides an overview of research on trajectory prediction in urban environments. This survey discusses various approaches to trajectory prediction, including data-driven methods that use machine learning and deep learning techniques. The authors highlight the importance of future location prediction in trajectory planning and discuss the challenges in accurately predicting destinations. The authors propose a data-driven approach [3] to future location prediction and trajectory planning that uses a combination of historical and real-time traffic data. The approach incorporates a machine learning-based next-destination prediction model and a genetic algorithm-based trajectory planning algorithm. The authors demonstrate the effectiveness of their approach through experiments on real-world urban transportation data.

In [4], the authors propose a trajectory planning framework that leverages real-time traffic data and vehicle-to-vehicle communication to generate optimized trajectories for connected vehicles in a citywide setting. The framework uses a machine learning-based next-destination prediction model to improve accuracy of trajectory planning. The authors evaluate their approach through simulations and demonstrate its effectiveness in reducing travel time and fuel consumption. In [5], we propose a trajectory prediction method that uses multi-source data, including road network data, vehicle sensor data, and real-time traffic data, to predict trajectories for autonomous vehicles in urban environments. The method incorporates a machine learning-based next-destination prediction model to improve trajectory prediction accuracy. The authors demonstrate the effectiveness of their approach through experiments on real-world urban transportation data.

To target the limitations, in this paper, we proposed a novel Spatial-temporal transformer-based feed-forward neural network (STTFFNs); the main objective of this model is to predict the future location and planning through the local and remote dependencies for the first time. In this work, the transformer model learns and is memorized with feed-forward network (FFD) citywide trajectories. In the end, the proposed model is based on the datasets that belong to real taxi trajectories. To conduct an extensive experiment on datasets, the results of the proposed model focus on both regional correlations and achieve a minimum error of 40% km. The results illustrated that this model outperformed the future location compared to existing models.

We have made the following contributions to this paper as follows:

- Our paper presents a novel deep learning model called the Spatial-temporal transformer-based feedforward neural network (STTFFN), which is capable of capturing long-range dependencies in spatial-temporal domains. The STTFFN model consists of units such as encoder and decoder that leverage the transformer architecture. Our approach aims to enhance the accuracy of predicting taxi drop-off locations by considering the driver's historical trajectory. To achieve this, we utilized the location-based social network (LBSN) API and FourSquare data to encode the spatial information of relevant semantic sites. Our proposed model outperforms existing methods in predicting future taxi drop-off locations.
- In this research, we design a spatial-temporal-based transformer that concurrently captures local and distant dependencies in order to forecast future location and trajectory planning. The framework design is built on a transformer-based skip-gram model, which allows us to record local dependencies and learn about spatial coordinate regularities through historical trajectories. The skip-gram model is used in conjunction with a transformer to determine the degree of similarity between the local areas.
- The feed-forward networks (FFNs) encode the trajectories obtained from GPS sequences into a fixed-length vector and then used cosine distance to determine the mean error and forecast the distant decencies.
- The extensive tests were conducted on two publicly available datasets: Manhattan and Porto. The experimental findings demonstrated that the proposed model outperforms other models.

The structure of our work is as follows: In Section 2, we provide a review of related works from both traditional and deep learning perspectives. Section 3 discusses the preliminaries, where we define key terms and present the problem statement. Section 4 describes our methodology and the various parts of our framework, with a focus on the novelty of our approach. In Section 5, we present the empirical study, including a description of the dataset and experimental configuration. We also discuss the implications of our approach from different perspectives. Finally, in Section 7, we provide a conclusion summarizing our work and discussing potential avenues for future research.

2. Related Work

A comprehensive examination of the relevant fields of study will be conducted to gain a thorough understanding of the matter. This will involve a thorough review of diverse literature sources such as books, journals, and conference papers that employ different methodologies. Approaches and strategies relevant to predicting the next destination will be scrutinized.

2.1 Trajectories Mining methods

In today's world, various GPS-enabled vehicles like taxis, buses, ships, and airplanes have become a common part of our daily lives. For instance, a significant number of vehicles in major cities possess GPS sensors that allow them to record time-stamped locations at fixed intervals. This location reports result in a plethora of spatial trajectories that can be leveraged for resource allocation. To enhance the precision of predicting future locations, incorporating semantic details of the places visited by individuals along with their location data was suggested by[6]. The proposed approach revolves around the idea of semantic trajectories, which depict the movement of an individual as a succession of locations annotated with semantic details. To simplify the forecasting of the subsequent location using semantic trajectories, the authors have formulated a two-part system known as Seman-Predict.The online mining module of Seman-Predict extracts semantic trajectories from raw data by initially determining the stopping points of a trajectory [7][8]. These stopping points indicate locations where the user has spent a specific duration of time. The trajectory data presents various possibilities for scrutinizing the movement patterns of mobile entities[9]. To understand human behavior, it is crucial to recognize their movement patterns. Extracting the high-level semantics of these patterns, which allow for inferring the underlying objectives or tasks of moving objects, remains a significant challenge in this domain.

The paper by [10] introduced a method to predict the endpoint of a trajectory based on a partially observed sub-trajectory. They divided the space into cells and modeled the transition probability between adjacent cells using a first-order Markov model. Another study revisited the problem related to RNNs. Two studies[11], proposed a Bayesian model that can predict a vehicle's future movement on a road network. Once again, the spatial transition was modeled using a first-order Markov model. [12] explored the feasibility of modeling trajectories with RNNs and assumed that accurate destination road segments for trips are known. They discovered their representations to assist with route decision-making since the Markov model necessitates explicit dependency assumptions and is not well-suited for accounting for long-term interdependence. When working with low-sampling-rate trajectories, it can be challenging to compute similarity because a fragmented trajectory may correspond to various possible paths. To address this problem, [13] proposed using hidden Markov models to learn the transition patterns among a set of spatial objects based on their historical trajectories. Sparse trajectories are then aligned with these spatial objects to compute similarities. An alternative solution was proposed by [14] using a seq2seq-based model that encodes the most probable route into the trajectory representation, thus resolving the aforementioned issue.

2.2 Future Destination Methods

In this section of the literature review, different approaches for location forecasting have been explored. With the increasing use of mobile devices and wireless networks, location-based services (LBS) [15] have become an area of growing interest. One of the essential tasks in LBS is next location prediction, which anticipates a user's next location based on their previous location history. Predictive models for location-based services (LBS) are essential for providing proactive assistance to users in an ever-changing environment. Individuals tend to exhibit a high degree of regularity in their mobility behaviour, visiting a limited number of locations and travelling between them in a regular pattern[16][17]. These mobility patterns can be characterized as temporal, periodic, or sequential.The increasing availability and popularity of mobile technologies, such as positioning, computing, and communication, has led researchers to realize the potential of utilizing mobility patterns to forecast the movement of objects in the future. This predictive system has broad applications, including transportation research for urban development, optimization of data phone networks, and improvements to location-based services (LBS). The next-generation mobile network operator applications have various fundamental components, including collecting users' current locations and the transition of locations, anticipating their future destinations, providing location-specific information, and managing relevant communication requests[18]. The transition from one location to another is a typical human behavior that can be utilized to generate end-to-end user movement trajectories [19]. Next-place forecasts are considered as the basic unit for generating these trajectories, which can be inferred using previous trace data.

To tackle these issues, we proposed a novel Spatial-temporal Transformer-based Feedforward Neural Network (STTFFN) to tackle these challenges. The primary objective of this model is to forecast future destinations and plan based on both local and distant dependencies.

3. Preliminaries

In this section briefly explains the passenger pickupoff/dropoff prediction definitions and problem statement.

- **Definition 1:** There are city areas with many distinct locations in terms of semantic meanings and varied granularities. In this study, the entire city is split $M \times N$ grid map into longitude and latitude. The regions of a city are designated by the letters R. They are expressed as the $i - \mathrm{th}$ row and $j - \mathrm{th}$ column of the grid map $R(i, j)$.
- **Definition 2:** Each taxi trip must be documented using the following tuples: $(\mathrm{log}_{\mathrm{pickup}}, \mathrm{lat}_{\mathrm{pickup}}, \mathrm{log}_{\mathrm{dropoff}}, \mathrm{lat}_{\mathrm{dropoff}}, \mathrm{t}_{\mathrm{pickup}}, \mathrm{t}_{\mathrm{dropoff}})$, where $\mathrm{log}_{\mathrm{pickup}}, \mathrm{lat}_{\mathrm{pickup}}, \mathrm{t}_{\mathrm{pickup}}$ is the pickup longitude and latitude in time period t. Similarly, $\mathrm{log}_{\mathrm{dropoff}}, \mathrm{lat}_{\mathrm{dropoff}}, \mathrm{t}_{\mathrm{dropoff}}$ is the drop-off longitude, latitude in time interval t.
- **Definition 3:** Suppose a taxi driver $\aleph$, the trajectory of the taxi driver is $T_{\aleph} = P/D_1, P/D_2 \cdots, P/D_k$ is the pickup/dropoff spatial-temporal time order sequence points, which describes the last pickup/dropoff point P/D_k of a taxi driver $\aleph$. In the trajectory $T_{\aleph}$, the $P/D_k \in R^{\mathrm{lat.log}}$ belongs to latitude and longitude and k is the length of the trajectory. The trajectory points P/D_1 and P/D_2 are the movement from one point to other points and so on the geographical coordinates k.

- **Problem Statement:** The trajectory of a taxi driver normalized in such a way $\mathrm{T}'_{\aleph} = \mathrm{R}_{\mathrm{i}}\mathrm{j}$, where ij is the set of regions $\mathrm{ij} \in [\mathrm{i}, 2, \cdots, \mathrm{n}], \mathrm{R}_{\mathrm{ij}} \in \mathrm{D}, \mathrm{R}_{\mathrm{ij}}$ is the set of trajectory clusters with length n of road T'. The trajectory of a taxi driver $\mathrm{T}'_{\aleph} = \mathrm{Pickup}_{\mathrm{k}+1}$, where $\mathrm{Pickup}_{\mathrm{k}+1}$ is the current pickup point of the taxi driver, the task of this research to predict the next actual destination of the $\aleph$ dropoff point $\mathrm{Dropoff}_{\mathrm{k}+2}$.

4. Feature Engineering

This section discusses the feature engineering of a proposed transformer-based model.

4.1 Map Decomposition

In metropolitan areas, it can be found that a sizable portion of data is spatial-temporal in nature, such as traffic data, including taxi trajectory data, metro card swiping, bicycle renting and returning data, etc. These data are constantly changing and include time and geographic locations. To represent and quantify this data at the time and spatial scales. These spatial-temporal data, in order to process in a better way, city map should decompose first. The grid-based decomposition method is used in this work. The grid map is divided into $\mathrm{M} \times \mathrm{N}$ grid cells based on longitude and latitude. For example, the region of a grid map is (i, j), i_{th} is represented row, and j_{th} represents the column of the city grid map.

4.2 Missing Data Cleaning

Spatial temporal data is often missing owing to sensor failure, other human variables, and communication problems. Data analysis subsequently has brought a negative impact due to the data missing problem. At the moment, the primary approach for processing data that is missing is to fill in the missing value. Missing data further has two properties, such as imbalance data and uncertainty of data. Two primary manifestations of spatial-temporal data imbalance are imbalanced data labeling and distribution. With the implementation of machine learning algorithms, researchers advocated using uncertainty quantification to ease the issue of data insecurity. This work addresses missing data problems with spatial-temporal implicit correlation. The K-mean clustering technique is utilized to identify the solution to data imbalances or distribution imbalances and data uncertainty.

4.3 Trajectory Normalization

For the sparsity problem to be overcome, a method is put forward for the normalization of trajectories. To mitigate the effect of sample uncertainty, this technique transforms GPS geographical coordinates into a road junction sequence, complete with the driver's behavior and directions. This procedure has two benefits: first, it maps the trajectory from geographical coordinates to road trajectories, and second, it eliminates the uncertainty associated with location sampling. Three main steps contain the normalizing process: the first one is data cleaning, such as missing data, imbalance, and uncertain data; the second is map decomposition and matching, and the third is the extraction of junction sequence. The data cleaning process removed the data loss, errors in data, drifting sampling, and some atypical sampling. The preprocess data according to driver behavior and driving regulations. The GPS parameters by longitude and latitude can be snapped using the map matching technique by using the skip-gram model and Haversine distance. To extract data from road intersections in order to forecast the driver's next destination-related behavior.

4.4 Extraction of Spatial-Temporal Features

Movement in the ITS is governed by numerous factors, such as the geographical and temporal features of visited locations. Every region has distinct activities that give the region's location a significant meaning in relation to the action. Therefore, the illustration of a specific region is essential for evaluating mobility. This has the advantage of enriching the information associated with each location. The proposed model is an embedding module for multiple modules. The module combines driver features and spatial-temporal information for feed-forward neural networks to represent a vector. Some external features are available to improve the accuracy of predictions; for instance, meteorological conditions characteristics are merged into the weight matrix.

4.5 Behavioral Features of Taxi Driver

As the driving features analysis, the taxi process must be taken starting from two stages, pickup, and dropoff, and then driving have different characteristics in various stages. To encode the behavioural features of a taxi driver, we outline the categorized attributes of the driver. The primary consideration is time, where the time of day is represented in hours. For example, the range is denoted as $h \in [1,24]$, covering the hours from 1 to 24. Another significant aspect is the classification of weekdays, which is represented as weekdays $wkd = [1,7]$, indicating the range from 1 to 7, representing the total number of weekdays. The days of the week are broken down into different groups, including days of work, weekends, vacations, and pre-holiday. These characteristics are modeled using a one-hot approximation. Totally separate spatial and temporal transformers generate the vector's dense encoding with the proposed model network's embedding layer. During the training of the proposed method, weight values are updated.

4.6 Semantic Features

Each cluster in the collection of geographical clusters is endowed with spatial semantic properties. Each POI identifies the closest cluster. The features associated with POI are derived from the mapped route. The location of the POI is hierarchical structures categorized, for example, educational locations and other locations of libraries and institutions. The classifications of POI indicate the area's combined depiction. The spatial semantic features are embedded in the spatial transformer model during its embedding phase. The semantic characteristics are derived from the POI; they are linked with the track and tally of the aggregated macro-categories of the POI. The cluster encoding has the ability to measure every component within the cluster. The distribution of POI is L number of the Chanel matrix, i.e., $\mathrm{Poi} \in \mathrm{R}^{\mathrm{M} \times \mathrm{N} \times \mathrm{L}}$, L is the category of POI, while M and N is the width and height of the city map. In this work, region $\mathrm{R}_{\mathrm{i,j}}$ is the collection of all regional areas, and i and j represent the longitude and latitude of trajectory points. The type of POI in a region mathematically can be expressed as tensor $\mathrm{T}_{\aleph}^{\mathrm{R(i,j)}} = \frac{\mathrm{T}_{\aleph}^{\mathrm{R}}\mathrm{i}}{\Sigma_{\mathrm{i}=1}^{\mathrm{L}} \mathrm{T}_{\aleph}^{\mathrm{R}}\mathrm{i}}$ where represent $\mathrm{T}_{\aleph}^{\mathrm{R}}\mathrm{i}$ kind of the point of interest i_{th} in region R.

4.7 Geographical Zone Embedding

In a semantic way, spatial characteristics represent a region's time-independent properties without capturing the region's zone-specific dynamics. In this scenario, urban transport information is the finest resource for accurately describing human movement in cities. Consequently, the urban region has comparable embedding characteristics and produces identical geometric outcomes. From this vantage point, individual mobility is viewed as a language pattern. In this manner, the location order is the word series. This demonstrates that the stream of text information is represented. Natural language processing is implemented in the sequences model. In this method, the entire sentence is encoded in the hidden context; so every word conveyed to the decoder section has its own hidden layer. The hidden states are responsible for decoding the results from the encoding device at each phase. In this strategy, the transformer is employed to interpret, and a word provided to the contextual embedding is acquired from the adjacent frequency window. Similar terms have the same vector representation from a semantic perspective. Thus, every location aggregation bears a descriptive designation. The word sequences will be used to map the trajectory in this fashion. By utilizing the encoder and decoder units, we can better comprehend the zone embedding illustrated in **Fig. 1**. This embedding is based on various zones with unique mobility patterns and relationships.

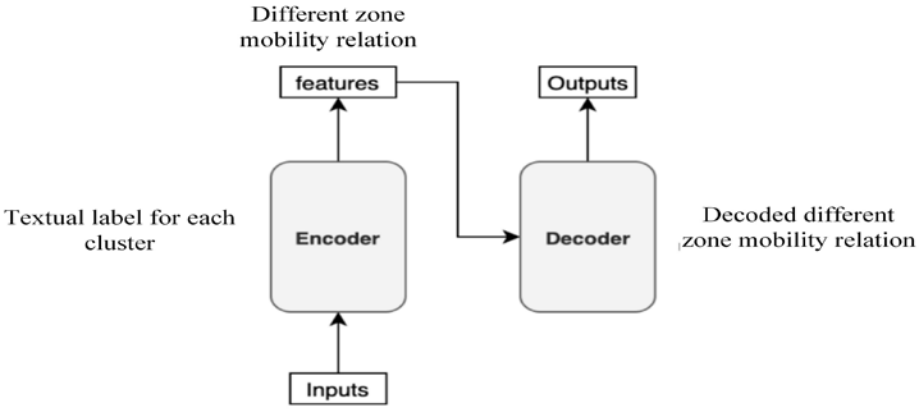

Fig. 1. Operation of the encoder unit as a token for zone embedding

5. Architecture Overview

A transformer is a major constituent that transforms one sequence into another. Encoders and decoders facilitate the completion of all of these operations. In contrast to pre-existing sequence-to-sequence models, the transformer belongs to the non-RNN category and is distinct from RNN. The attention processes are the transformer's primary cornerstone, allowing the subject to acquire any portion of the patterns despite their distance. As shown in **Fig. 2**, the layout of the transformer consists primarily of encoder and decoder modules. Encoder units of the same type are positioned at the base of the layered decoder components. The encoder unit comprises a self-attention layer and a position-dependent feed-forward layer, whereas the decoder module includes an extra layer, which is the encoder-decoder attention layer. An additional layer is introduced between the self-attention and feed-forward layers, serving as a connection between the encoder and decoder segments.

Embedding each driver trajectory sequences to comparable with word/token segments during transformation. The transformer model utilizes multi-head attention layers, which assign varying levels of importance to different words or tokens in a sequence based on different facets or heads. This information is then concatenated and subjected to a linear transformation before being assigned to separate output heads. Both the encoder-decoder attention layers and the self-attention layers employ identical attention mechanisms. For instance, self-attention can serve as a demonstration of this mechanism. Self-attention operates on token sequences, such as trajectories, denoted as $\mathrm{T} = (\mathrm{P}_1, \mathrm{P}_2, \dots, \mathrm{P}_\mathrm{n})$; each token in the sequence is represented by a vector, which undergoes linear transformation and is refreshed through a weighted sum of the other words. The weights are established based on their similarity, referred to as the attention index. Take the update of $T_i$as an example in this context:

$$y_i = \Sigma_{j=1}^{n} a_{ij} \left(W_V T_j \right) \tag{1}$$

In the expression above, y_i represents the updated value of T_i and a_ij represents the attention score. To determine the degree of similarity among both T_i and T_j by:

$$a_{ij} = \frac{exp(C_{ij})}{\Sigma_{K=1}^{n} exp(C_{ik})} \tag{2}$$

where congruence C_ij between two linearly transformed T_i and T_j is measured. To determine the scaled dot product by:

$$(C)_{ij} = \frac{\left(T_{i_{W_Q}}\right)\left(T_{j_{W_K}}\right)^T}{\sqrt{h}} \tag{3}$$

To enhance the adaptability of the transformer, three linear transformation matrices are represented as W_V, W_K, and W_Q are introduced, where the final dimensionality is represented by h. These three matrices perform the same linear transformation as W. Five components make up the proposed deep-wide model STTFFNs, with each element operating as described below.

- The map matching technique is used to divide the whole city into Local areas and global areas, embedding the taxi trajectories in the first step. To compress the trajectory sequences based on their spatial similarity relationship. To use the feed-forward neural network, which helps the learning of the global dependencies. The city-wide average error calculated distance score for local and global regions using Haversine distance.
- In the proposed method for designing novel spatial-temporal transformers, transformers are founded on a component for multi-modal integrating spatial characteristics. The encoder component of the transformer model converts the input tokens into dimensional vectors through the spatial and temporal transformer embedding layers. The input elements are then passed through the positional encoding layer to maintain the sequence order. Mathematically, positional encoding can be defined as follows, with the aim of preserving the sequential ordering of the sequence:

$$Embedding_{(2i)_{pos}} = sin\left(\frac{pos}{10{,}000^{\frac{2i}{h}}}\right) \tag{4}$$

$$Embedding_{(2i+1)_{pos}} = cos\left(\frac{pos}{10{,}000^{\frac{2i}{h}}}\right) \tag{5}$$

In the provided equation, the variable "i" signifies the dimension of the positional embedding, while "h" represents the size of the hidden layers.

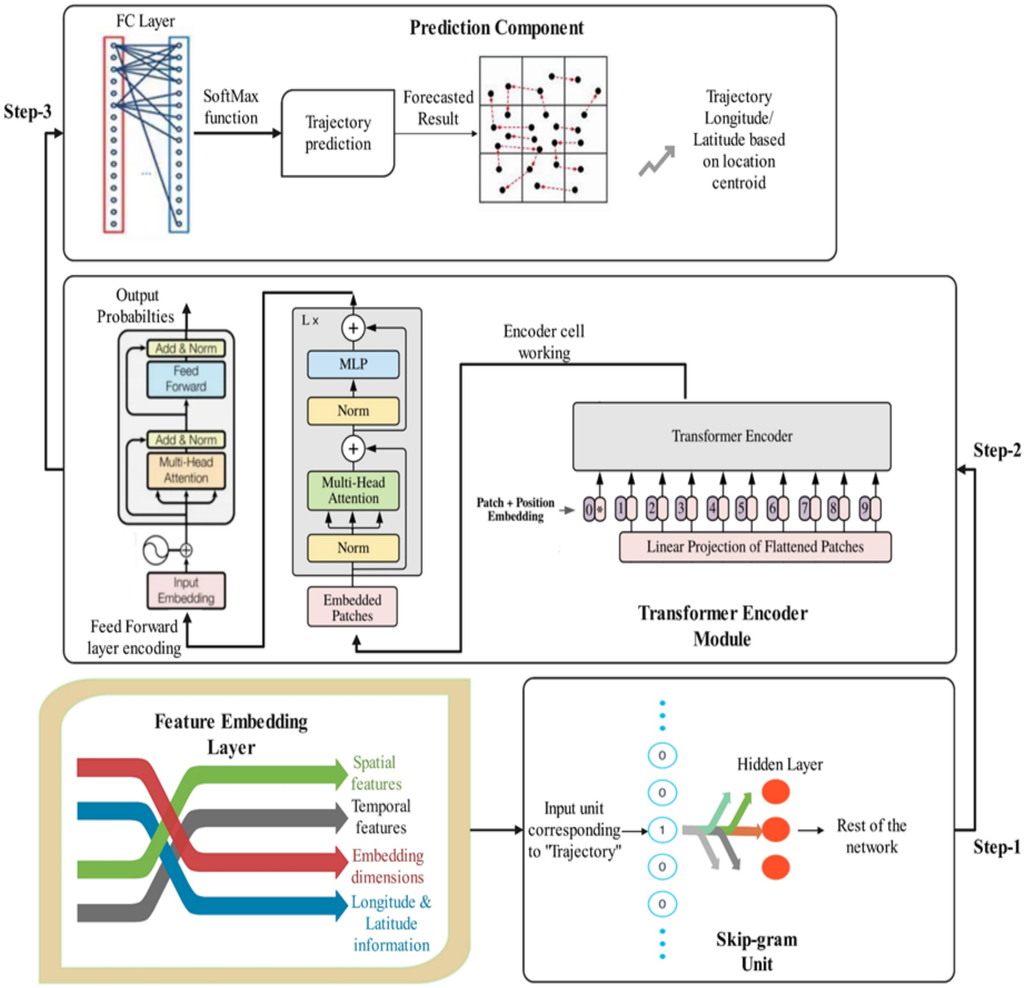

Fig. 2. Mainframe work architecture

$$W \odot (embedding)_{pos} \in R^h \tag{6}$$

In which W is the concatenation operator and the three linear transformation matrices.

- To cope with GPS data, leverage unique information sources to include spatial and temporal aspects. The new sources of data utilized are location-based social networks and mobile phones. These sources can be utilized to gather data on human mobility patterns and to improve the efficiency of municipal traffic made use through Four-Square. This LBSN identifies the quantity and sorts of events happening in the future location, as well as the number of users who can obtain service in that region. All of the information received may subsequently be utilized to deduce the taxi's location.
- The attention mechanism was used in the process of obtaining sequential data from GPS traces and spatial-temporal input characteristics. The transformer network is capable of learning both spatial and temporal features. The learning process of the suggested network entails embedding spatial-temporal features derived from taxi driver behavior alongside semantic features, emphasizing trajectory planning.

- The last component of the proposed network design is prediction. The encoder and decoder stack merge the output of previous units to complete the forecasting operation.The forecasted part is made up of two layers: a linear layer and a Softmax layer. In contrast, the SoftMax layer consists of multiple neurons, where the number of neurons is denoted as $\mathrm{m} = |\mathrm{C}|$. With the help of a clustering algorithm, construct a collection of geographic clusters C. The K-means clustering algorithm is utilized to train all trajectories towards their final destinations. Leveraging latitude and longitude information, each point along the route is allocated to the closest centroid C_i. Incorporating two additional neurons into the output layer enables the representation of cluster center coordinates for longitude and latitude. Notably, the initialization weights of the matrix can be adjusted using the cluster centers operation, akin to the linear output layer.

5.1 Local Trajectory Planning

The local or neighbourhood surrounding locations R(i,j) at each time interval t, treated as $\mathrm{M} \times \mathrm{N}$ image having one channel of the values of local trajectories with R being the image center where spatial granularity controls by the size geographical coordinates k. From a local point of view, used zero padding for the locations R(I,j) at the boundaries of city-wide area. The image of a tensor as a result of $\mathrm{T}_{\aleph,\mathrm{t}}^{\mathrm{R(i,j)}} \in \mathbb{R}^{\mathrm{M}\times\mathrm{N}\times\mathrm{C}}$ in time interval t for each location i, j. To encode the spatial correlation in the layers transformer model $\mathrm{T}_{\aleph,\mathrm{t}}^{\mathrm{R(i,j)}}$ as input to feed with $\mathrm{T}_{\aleph,\mathrm{t}}^{\mathrm{R(i,j),0}}$ with W adjustable weights. For example, the taxi driver pickup and dropoff the passengers in Region R(i,j) during the day $\mathrm{D}_{\mathrm{R(i,j)}} = \{1,2,3,\cdots,\mathrm{n}\}$ with n demand. As denoted $\mathrm{T}_{\aleph.\mathrm{Dt}}^{\mathrm{Pick.R(i,j)}}$ and $\mathrm{T}_{\aleph.\mathrm{Dt}}^{\mathrm{Drop.R(i,j)}}$, mathematically illustrated as:

$$T_{\aleph.Dt}^{Pick.R(i,j)} = \left(T_{t0}^{i,j}, T_{t1}^{i,j}, \cdots, T_{tn}^{i,j}\right)^{T} \tag{7}$$

$$T_{\aleph.Dt}^{Drop.R(i,j)} = \left(T_{t0}^{i,j}, T_{t1}^{i,j}, \cdots, T_{tn}^{i,j}\right)^{T} \tag{8}$$

Where $\mathrm{T}_{\aleph.\mathrm{Dt}}^{\mathrm{Pick.R(i,j)}}$ is the pickup distribution places in region i, j, and $\mathrm{T}_{\aleph.\mathrm{Dt}}^{\mathrm{Drop.R(i,j)}}$ dropoff distribution places in region i, j from the time interval t of Dt.

As mentioned above about W adjustable weights, in this work employed the skip-gram model, which automatically learns the spatial correlations of local areas from the historical trajectories. Bag of words (CBOW) and Skip-gram model is two variants of Word2vec model, which utilizes raw texts to train the word vectors. Both variants are algorithmically identical; the difference is that skip-gram forecasts the middle future location, whereas skip-gram does the opposite and focuses on the surroundings of the middle future location. In the previous research, the author utilized the CBOW version to forecast the future destination; however, the CBOW variant has the restriction of providing efficient performance with limited data; nevertheless, if the data set is large may obtain reliable prediction using the skip-gram model. As definition 2, The sequences of the taxi driver trajectory $\mathrm{T}_{\aleph}$ employed as training, the skip-gram model has the objective to set maximized average log probability can be defined as:

$$\frac{1}{T}\Sigma_{\aleph=1}\{^{T}\Sigma_{\delta=-\chi}^{\chi} lo\,g\,p\,(T_{\aleph}\{'\} + \delta|T_{\aleph}) \tag{9}$$

Where training window size is χ, when computing log probability, the inner summation goes from $-\chi$ to χ and absolutely predicts the trajectory $\mathrm{T}_{\aleph}'$ given the middle of trajectory $\mathrm{T}_{\aleph}$. The outer summation is traversed by all the trajectories in the training corpus, which come from the skip-gram hidden layers.

In the skip-gram model, every trajectory cluster $T_{\aleph}$ with two learnable parameter vectors associated with longitude and latitude point of view, respectively. The set of parameters is $A_{T_{\aleph}}$ and $B_{T_{\aleph'}}$, and they represent the pickup and dropoff trajectory of taxi as a vector $T_{\aleph}$ respectively. To apply the softmax function to correctly calculate the probability of the prediction trajectory $T_{\aleph',i}$ and $T_{\aleph',j}$ with longitude i and latitude j coordinates C is the set of clusters in the trajectories, mathematically as follows:

$$P\left(T_{\aleph'.i} \middle| T_{\aleph',j}\right) = \frac{exp\left(AT_{\aleph'.i}BT_{\aleph'.j}\right)}{\Sigma_{l=1}^{C} exp\left(C_l BT_{\aleph'.j}\right)} \tag{10}$$

The above equation encoded into a tensor as $\mathrm{P}\left(\mathrm{T}_{\aleph',i} \middle| \mathrm{T}_{\aleph',j}\right) \in \mathrm{R}$.

5.2 Global Trajectory Planning

The areas are typically shown spatial co-occurrences among highways, in historical trajectories are comparatively distant from one other, so the trajectories term them remote dependencies. The proposed work is to learn remote dependencies employ feed-forward neural network (FFN); the distant trajectories of the taxi encoded to learn the remote correlation into vector-based illustrations. The basic structure of FFN comprises an input layer, a hidden layer, and an output layer. Neurons comprise the input layer; its role is to accept neurons as input and pass them on to the subsequent layers. The input layer's number of neurons should be sufficient be equal to dataset's total number of attributes. The next layer is discussed as hidden layer, which is sandwiched in both layers of input and output. The function of the hidden layer is to contain a large number of neurons that undergo transformations prior to receiving input. This layer updates and trains the network's weights to make it more predictive. The last layer is the output layer; its primary purpose is to forecast the characteristics of the model that can be created. The operation of FFD is depicted in **Fig. 3**. As stated, before about neuron weights, the main objective is to apply intensity or amplitude between the correlations of two neurons. The weight is distributed randomly, often between 0 and 1. However, neural networks are algorithmically computed data in three simple stages, i.e., the first step is to multiply the inputs and weights, the second step is to add the biases, and the third step is to use the activation function. Finally, the output signal is transformed into a weighted sum using the activation function, which is also known as the transfer function. The operations of FFN mathematically can be expressed as follows:

$$In_t = \sigma\left(W_{H_t} x_t \mu_t + b_{In_t}\right) \tag{11}$$

$$H_t = \sigma\left(W_{In_t} x_t \mu_t + b_{H_t}\right) \tag{12}$$

$$O_t = \sigma\left(W_{H_t} x_t \mu_t + b_{O_t}\right) \tag{13}$$

Where σ is used as an activation function, $\mathrm{W}_{\mathrm{In}_t}, \mathrm{W}_{\mathrm{H}_t}$, and $\mathrm{W}_{\mathrm{O}_t}$ is related input, hidden layer, and output weight matrices, x_t, and μ_t input and output variables, $\mathrm{b}_{\mathrm{In}_t}, \mathrm{b}_{\mathrm{H}_t}$, and $\mathrm{b}_{\mathrm{O}_t}$ are expressed as biases. The finite training set as given $(\mathrm{x}_i, \mu_i) \in \mathbb{R}^{M \times N}$, the input and output variables with η number of neurons is written as general form:

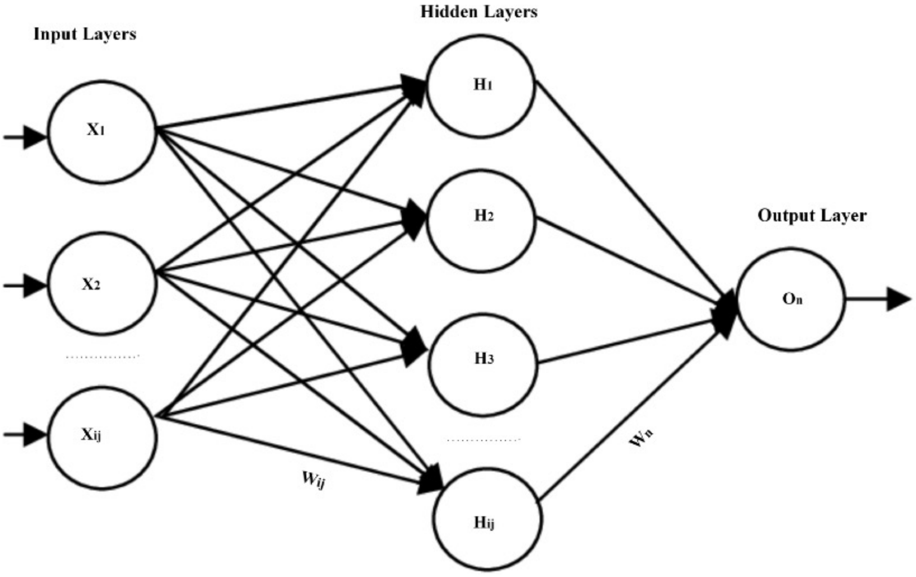

Fig. 3. Working of Feed-forward network

$$\Sigma_{j=1}^{\eta}\gamma_j\sigma_j(x_i) = \Sigma_{j=1}^{\eta}\gamma_j\sigma\left(W_jx_i + b_j\right) = Z_i \tag{14}$$

Where j_{th} hidden layer of neuron has input weight is W_j, and output weight is γ_j. σ is the activation function and b_j is threshold for j_{th} hidden layer of neuron. The training error can be minimized as

$$\Sigma_{i=1}\left||Z_i - \mu_i|\right| = 0 \tag{15}$$

Which is equal to the output matrix and can be expressed as:

$$\Sigma_{j=1}^{\eta}\gamma_j\sigma\left(W_jx_i + b_j\right) = \mu_i \tag{16}$$

5.3 Transformer Networks Learning and Prediction

The GPS location data's temporal order is determined by tracking movement. Similar to previous research, RNN outperforms alternative architectures like MLPs in tracing the mobility of sequential data. These networks are not intended to operate in tandem with spatial-temporal data. Nevertheless, the inability to parallelize training data is a limitation of RNN. Parallelization is utilized by the transformer network to organize sequential and temporal data. In addition, prior studies concentrate primarily on singular, fine-grained trajectories to predict the next vehicle location. In other terms, these approaches are founded on each ride's GPS coordinates. The earlier research has significant drawbacks, such as the need to retain a massive amount of GPS data and almost all trajectory points in order to predict the trip's conclusion. To circumvent the aforementioned issue, the conceptual model trajectory conforms to the definitions provided in the "Preliminaries" section. The sequences of GPS points for both pick-up and drop-off are comprised of pairs from multiple transportation vehicles. This is the most effective method to avoid maintaining pick-up and drop-off locations along the full journey. In this manner, the forecast is instantaneous; only the beginning and conclusion points of the voyage can be determined. Consequently, a large number of transportation strategies are formulated as classification problems to determine the next location — the objective of the mobility model is to classify the predicted location. The primary disadvantage of this strategy is that the majority of locations are unknown during model training. The model never generates multiple locations. To surmount this drawback, the proposed method forecasts the ultimate location using two distinct coordinate functions, longitude and latitude.

The task related to predicting the destination is tricky; in order to decrease the difficulty, use the k-mean clustering technique to create midpoint clusters for sample data and clustered centroid points for the prediction point of view of the next destinations. The main objective of the centroid is to estimate their probabilities with the help of destination prediction. In this work used local and global trajectory optimization to predict the future location geographical coordinates in terms of longitude and latitude, respectively. In the main framework architecture, Figure employed a fully connected layer to map local and remote trajectories with operational points. The probabilities of each operational point can estimate through softmax function. After the training and testing process of this model, the degree of spatial interaction between the roads examined that the value of higher similarity has stronger traffic correlations. The Haversine distance is used as a loss function, and finally, get estimated longitude and latitude coordinates of the nest destination. The two vectors x, μ, and $cos\theta$ are expressed as dot product and magnitude in cosine similarity formula which is mathematically illustrated as:

$$Cos(\theta) = \frac{x.\mu}{||x||\mu|||} = \frac{\Sigma_{i=1}^{t} x_i \mu_i}{\sqrt{\Sigma_{i=1}^{t} x_i^2}\sqrt{\Sigma_{i=1}^{t} \mu_i^2}} \tag{17}$$

In above equation, x_i and μ_i is the component of vectors x and μ. According to the definition, as defined in the problem formulation section, the Haversine distance can be defined on the basis of two points' longitude and latitude, respectively. The trajectory of taxi $T_{\aleph}^{R}(i,j)$ have pickup and dropoff $R(i,j)$ longitude and latitude, mathematically can defined as:

$$T_{\aleph}^{R(i,j)} longitude = \Sigma_i \frac{exp(\tau_i) longitude_i}{\Sigma_j exp(\tau_j)} \tag{18}$$

$$T_{\aleph}^{R(i,j)} latitude = \Sigma_i \frac{exp(\tau_i) latitude_i}{\Sigma_j exp(\tau_j)} \tag{19}$$

The above equation represents the predicted longitude and latitude, and τ_i is the previous layer activation in learning the model. These longitudes and latitudes used two points as pickup and dropoff, employed of loss function as:

$$R(point_1, point_2) = 2.r.\tau tan\left(\sqrt{\frac{\tau}{1-\tau}}\right) \tag{20}$$

$$\tau = sin^2\left(\frac{latitude_2 - latitude_1}{2}\right) + \\ cos(latitude_1)\, cos(latitude_2)\, sin^2\left(\frac{longitude_2 - longitude_1}{2}\right) \tag{21}$$

Where $point_1$ is the forecasting, $point_2$is the actual location, $latitude_1$, and $latitude_2$, is the geographical coordinates of forecasting $point_1$.

6. Experimental Study

The use of taxi trajectory data is essential in developing better transportation infrastructures and policies by observing, evaluating, and optimizing traffic flow. Urban areas face a common issue of traffic congestion, often caused by poor road planning, lack of control, and insufficient maintenance. To evaluate the proposed model, two real-world datasets from Manhattan, New York, and Porto were used.

6.1 Datasets

The Porto dataset comprises 1.7 million cab route information collected from 442 taxis. To obtain 200,000 taxi trajectory trips, 600 drivers were selected from the initial 5,000. The dataset period ranges from 2013-07-01, to 2014-06-30. The GPS points from each ride dictate the pick-up and drop-off locations, with spatial-temporal features sampled every 15 seconds. Conversely, the Manhattan dataset was obtained from 13,426 taxis operated by 35,000 unique drivers, spanning from January 3, 2013, to January 3, 2014. The dataset encompasses 9,100,000 taxi trips' trajectories and records spatial-temporal characteristics every 10 seconds during pick-up and drop-off. The datasets comprising taxi trips include metadata that provides information on the taxi's identification number, the type of origin, the day type, and the starting timestamp. This information was used to gather useful insights into people's mobility patterns, such as their trips to the office or returning home. In this study, the Porto dataset was used, and the cab driver traces were utilized as a series of inputs for the pickup/drop-off destinations. The taxi drivers were grouped by their ID and sorted in ascending order based on different timestamps. The taxi pickup/dropoff points were used to construct a taxi trajectory by considering four past trips and imposing a $T_{\beta} = 8$ .This approach strikes a good balance between maintaining the relevant history and learning the driver behavior model. The development of the model involved the selection of a driver in relation to another driver with the same time-shift and the utilization of trip sequences that have a maximum time gap of three hours between them.

6.2 POI Extraction with Taxi Trajectory

The points of interest (POI) typically include geographical coordinates, such as longitude and latitude, as well as textual information about the activity occurring at that location. POIs can provide various levels of detail for characterizing activities based on hierarchical categorization. Instances of hierarchical classification include categories Examples include medical facilities, food establishments, Chinese restaurants, and Western restaurants. In this work, FourSquare was utilized as an online resource to extract POI points. FourSquare is an online geographical social network platform that offers recommendations for places, along with information on activities. The FourSquare API allows for up to 100,000 requests per day at no cost. Each location-based social network (LBSN) structures its location categories based on the activities taking place at each location, such as shops or restaurants. LBSNs recommend locations and activities semantically based on their semantic characteristics and associated historical paths with POI. For example, to establish the path for activities at Liberty Hospital belonging to the Medical category, it would be structured as Medical---Kingadward Hospital---Youhana Hospital. This type of structure is more informative when future locations are POI names. The structure that has been established is composed of various characteristics, and each node includes augmented proximity data. LBSNs, including FourSquare, categorize POIs into macro-categories such as Proficient and Other Places, Residential areas, colleges and universities, shops, food establishments, services, art and entertainment venues, and travel and transport facilities are included. The dataset extracted from FourSquare yielded 8,500 POIs for Porto and 65,300 POIs for Manhattan. While census data and land usage are interesting factors, the proposed model only considers POI. Although there is potentially more intriguing information about spatial data available, it is not consistently accessible. Additionally, the availability of datasets for other cities may not be uniform in terms of their definition convergence.

6.3 Tools and Technology

Tools and technologies for deep learning experiments include a wide range of software and hardware resources. Popular deep learning frameworks such as TensorFlow, PyTorch, and Keras provide a user-friendly interface to design, train and evaluate neural networks. These frameworks can be run on powerful graphics processing units (GPUs) or tensor processing units (TPUs) to accelerate the training process. Additionally, cloud-based services such as Amazon Web Services (AWS) and Google Cloud Platform (GCP) offer pre-configured deep learning environments, GPU/TPU resources, and storage solutions. Data processing and visualization libraries such as NumPy, Pandas, and Matplotlib are also widely used in deep learning experiments. Furthermore, tools like Docker, Kubernetes, and Jupyter Notebooks enable reproducible experiments and facilitate collaboration among researchers. During the experimentation phase, we utilized both Google Cloud and an NVIDIA 2060 RTX GPU. The model was trained using Adam as the optimizer and Mean Squared Error (MSE) as the loss function. The model was trained using a batch size of 16 and a learning rate of 10^{-3}, with 100 to 300 epochs. To maintain consistency, we conducted 6-7 separate experiments on every dataset.

6.4 Parameter tuning and time Complexity

In the experiments were conducted on two datasets, Porto and Manhattan, which comprised 200,000 and 350,000 taxi rides, respectively. Random partitioning was used to divide the datasets into three sets: training, testing, and validation, with a ratio of 70%, 15%, and 20%, respectively. The datasets consist of full trajectories with pick-up and drop-off points, following trajectory Definitions 1 and 2. The problem's spatial nature for stand classification suggested that the reliability and F1-score were insufficient to quantify the error accurately due to the unbalanced nature of the datasets. Therefore, the Error Distance Score (EDS)" was estimated using the Haversine distance" method, the function estimates the gap between the present and intended endpoints of a taxi journey.

$$ErrorRate.kms = \left(Hav_{Dis}(\hat{y}, y)\right) \tag{22}$$

The given equation represents the relationship between the forecasted location, denoted by ŷ, and the current origin or destination, denoted by y. As an evaluation matrice, RMSE (Root Mean Squared Error) is a measure of the difference between the predicted values and the actual values of a dataset. It is calculated as the square root of the mean of the squared differences between the predicted and actual values. The formula for RMSE is:

$$RMSE = \sqrt{\frac{\Sigma_n^{i=1}(y_i - \widehat{y_\imath})^2}{n}} \tag{23}$$

where n is the number of observations in the datasets, y_i is the actual value of the ith observation $\widehat{y_\imath}$ is the predicted value of the ith observation.

The proposed model was tuned and evaluated on two real-world datasets, with hyperparameters settings and training/testing time summarized in **Table 1**. Throughout the parameter tuning stage, a grid search technique was employed to seek out the best number of neurons, layer depth, and learning rate. The model's performance was evaluated using the validation set, and parameter values were selected accordingly **Table 1**. To ensure fair comparisons with previous studies, the K-means algorithm was employed to train the model with K parameters, with the Porto dataset set at 3392 parameters and the Manhattan dataset set at 2000 parameters. It's worth noting that the model remains unbiased in determining the number of clusters, opting for a minimum threshold based on the proximity of clusters to the centroid. Consequently, the model might yield a greater number of clusters compared to the distance ratio between clusters and centroid.

Table 1. Summary of Hyperparameter Configuration and Model Training

Datasets	Learning Rate	Activation Function	Neurons	Optimizer	Training Time (M)	Testing Time (S)
Porto	10^{-3}	SoftMax	256	Adam	74	0.23
Manhattan	10^{-3}	SoftMax	256	Adam	69	0.17

After a certain point during training, the number of clusters ceased to improve. The feature layer size was designated as 10, and the embedding layer size was set to 20. Furthermore, the transformer encoder and decoder have set the size of FourSquare macro-categories to 10. As per prior research representation, the vector size obtained after concatenating with embedding layers and feed-forward neural networks determined the input dimension of the linear layer and the scaled dot product. Training the model involved using the Adam optimizer with distinct values assigned to longitude and latitude. The model integrated various shared structures, including spatial and temporal transformers, a linear layer, a scaled dot product, weighted sum layers, a prediction layer, a dropout layer, and a feed-forward layer. These settings were carefully selected and tuned for a fair comparison, as outlined in previous research. The MSE was used as the loss function for both training and testing the model. Early stopping was employed as a checkpoint and optimization strategy to ensure that the model converged efficiently. This technique allowed the model training to stop when the MSE score on the validation set did not improve beyond a specified limit of epochs, usually 10 or 20. The MSE score was computed at each epoch during the training phase, and the network hyperparameters were stored if a good MSE future location was achieved on the validation set. During the testing phase, the proposed architecture employed the hyperparameters that produced the most optimal validation MSE score. To prevent overfitting, a dropout rate of 0.5 was implemented, and a window size of 5 was adopted for the model. The word embedding for textual labels was achieved using the encoder and decoder phases with a dimensionality of 20, and Jupiter was utilized as a lab experiment.

6.5 Comparison with Previous Baseline Methods

To demonstrate the performance and effectiveness of our proposed model, we conducted comparisons with seven baseline methods, adjusting parameters accordingly. We benchmarked our proposed model against the following state-of-the-art baselines, as detailed in **Table 2**.

Table 2. Comparison of baseline methodologies

Model	Citation	Description
Autoregressive Integrated Moving Average (ARIMA)	[17]	Highly effective for time series forecasting, predicts future values based on historical data using parameters such as autoregression, differencing, and moving average.
Nearest Neighbors (NN)	[18]	Takes taxi pick-up location as input, the neural network processing outputs the longitude and latitude of the nearest cluster centroid.
Multi-Layer Perception (MLP)		Extracts input features representing taxi trajectory from initial and last five GPS data points combined with

	[19]	metadata. Utilizes MLP with standard hidden layers and ReLu activation functions. Outputs destination cluster's weighted average centroid using SoftMax layer. Trained using Stochastic Gradient Descent with cross-entropy loss function.
Multi-layer Perception (MMLP-SEQ)	[19]	Similar to MLP, this model excludes the initial and last five GPS points from the input.
Fully Connected (FC-LSTM)	[20]	Utilizes LSTM with driver and time expressions as input. Connects longitude and latitude coordinates entirely and feeds them into the embedding layer as location. Weights are randomly adjusted and updated during training.
Long-Short-Term Memory (BOC+W2V)	[20]	Combines Bag of Concept (BOC) and Word2Vec (W2V) with LSTM. W2V provides representation for the RNN, integrated with both BOC and W2V features. Embeds zone coordinates using W2V instead of BOC.
Spatio-Temporal Graph Convolutional Networks (ST-GCN)	[21]	This model solely relies on spatial and temporal components, leveraging graph weights and embedding spatial-temporal features through a transformer architecture. Transportation transformer captures geographical interdependence of time series data to maintain continuity and consistency.

6.6 Performance Evaluation

Table 2 compares the performance results of the proposed model with the previous baseline models on both the Porto and Manhattan datasets. The proposed model demonstrates a significant performance advantage over the state-of-the-art model’s LSTM(BOC+W2V) and ST-GCN by a considerable margin. The anticipated model steadily outperforms the prior models by modeling dynamic spatial and temporal dependencies. The proposed model demonstrates efficient performance and is able to effectively capture long-range temporal dependencies and hidden spatial dependencies. Additionally, it was pragmatic that the presented model outperforms NN and simple MMLP in terms of performance. **Table 2** shows that NN and MMLP perform poorly in manipulative the error distance score in kilometers. On the Porto dataset, the proposed model outperforms NN and MMLP by 46-48% and demonstrates comparable performance to the enhanced variant of MMLP-SEQ. On the Manhattan dataset, the proposed model outperforms NN and MMLP by 35-40% and is on par with MMLP-SEQ. The proposed approach incorporates input features comprising driver and time information, while the spatial zone embedding involves feeding the coordinates of each zone cluster into the embedding layer. Among all the baseline models, the ARIMA model exhibits the poorest performance.

After analyzing the Porto and Manhattan datasets in **Tables 3** and **4**, we observed that ARIMA A had an RMSE of 5.10 and NN had an RMSE of 5.05 on the Porto dataset. Although ARIMA A had a lower RMSE, the difference was slight, so we need to consider other metrics before concluding which model has better performance. On the Manhattan dataset, ARIMA A had an RMSE of 4.65, which was lower than the RMSE of NN, which was 4.69. Similarly, MMLP had an RMSE of 4.47 and MMLP-SEQ had an RMSE of 4.29 on the Manhattan dataset.

When comparing the RMSE values of FC-LSTM and LSTM with BOC+W2V on the Porto dataset, FC-LSTM had an RMSE of 4.06, while LSTM had an RMSE of 3.41, indicating better performance by LSTM. On the Manhattan dataset, the RMSE values were 4.21 for both models.

Finally, when comparing the RMSE of ST-GCN with our proposed method, the RMSE on the Porto dataset was 3.71, and on the Manhattan dataset, it was 3.74. These results suggest that our proposed method performs similarly to ST-GCN in predicting the citywide trajectories. However, we need to consider other factors such as computational complexity, model interpretability, and generalization ability before making any final conclusions.

The FC-LSTM and its enhanced variant incorporating BOC and W2V demonstrate strong performance when integrated with LSTM, particularly LSTM(BOC+W2V), with the improved version exhibiting a lower error rate. In LSTM(BOC+W2V), BOC and W2V are employed to represent the features of the recurrent neural network. ST-GCN is surpassed by LSTM(BOC+W2V), resulting in a further reduction in error rate compared to previous models. The transformer approach, relatively novel, was initially introduced in 2017 for NLP translation. In the Porto dataset, the performance of the proposed approach is contrasted with LSTM(BOC+W2V) and ST-GCN, showcasing a 30-35% improvement over both aforementioned approaches. The results obtained on the Manhattan dataset are 25-32% better than the previous models. **Fig. 4(a)** shows the error distance score and RMSE on both the Manhattan and Porto datasets. In **Fig. 4(b)** and **(c)**, Pickup/Dropoff RMSE Statistics on Porto and Manhattan datasets for the STTFFNs are compared, respectively. To train the classification models, two neurons were removed from the output, and categorical cross-entropy was used as the loss function. This way, the locations' positions were limited to the list of centroids for clusters. In **Tables 3** and **4**, we observed that ARIMA had the worst results on both the Porto and Manhattan datasets, while LSTM and ST-GCN showed good forecasting results compared to our proposed model.

Fig. 4(d) and **(e)** represent the training and testing validation with training Epochs on both Porto and Manhattan datasets. The geo-coordinates were prejudiced by each cluster centroid through the associated probability of the SoftMax layer. This allowed for regression on the longitude and latitude variables to determine the precise position. Empirical findings showed a decrease in the EDS rate on the Manhattan dataset, indicating that the proposed method is capable of performing well in cities with uneven longitudinal and latitudinal stretches.

Table 3. Statistics of Porto and Manhattan datasets

Models	Porto (KM)	Manhattan (KM)	Porto (RMSE)	Manhattan (RMSE)
ARIMA	6.220	5.380	5.10	4.65
NN	3.215	2.375	5.05	4.69
MNLP	3.211	2.543	3.99	4.47
MMLP-SEQ	3.003	2.554	3.91	4.29
FC-LSTM	2.923	2.111	4.06	4.21
LSTM(BOC+W2V)	2.88	2.088	3.41	3.43
ST-GCN	2.67	2.00	3.71	3.74
STFFNs (Ours)	1.90	1.30	3.58	3.63

Table 4. Statistics of Porto

Models	Pickup (RMSE)	Dropoff (RMSE)	Pickup (RMSE POI)	Dropoff (RMSE POI)
ARIMA	4.80	5.50	4.95	5.61
NN	4.10	4.90	4.21	4.98
MNLP	3.65	4.05	3.66	4.11
MMLP-SEQ	3.51	4.02	3.63	4.11
FC-LSTM	3.70	4.10	3.81	4.16
LSTM(BOC+W2V)	3.46	3.98	3.53	4.06
ST-GCN	3.40	3.86	3.51	3.93
STFFNs (Ours)	3.12	3.61	3.26	3.81

Table 5. Statistics of Manhattan

Models	Pickup (RMSE)	Dropoff (RMSE)	Pickup (RMSE POI)	Dropoff (RMSE POI)
ARIMA	4.11	4.80	4.21	4.92
NN	4.05	4.76	4.13	4.83
MNLP	3.91	4.51	4.01	4.62
MMLP-SEQ	3.76	4.31	3.84	4.41
FC-LSTM	3.51	4.27	3.60	4.36
LSTM(BOC+W2V)	3.47	3.99	3.56	4.05
ST-GCN	3.20	3.79	3.29	3.87
STFFNs (Ours)	2.99	3.61	3.09	3.72

7. Conclusion

In conclusion, our research introduces a pioneering deep learning model, the Spatial-temporal transformer-based feedforward neural network (STTFFN), tailored to capture intricate spatial and temporal dependencies. Leveraging encoder and decoder units within the transformer architecture, our model excels in predicting taxi drop-off locations by harnessing the historical trajectory of drivers. We effectively encode spatial information by integrating location-based social network (LBSN) APIs and FourSquare data, enhancing prediction accuracy. Our framework seamlessly integrates spatial-temporal transformers to concurrently capture local and distant dependencies for trajectory planning and future location forecasting. We adeptly discern similarities between local areas by employing a transformer-based skip-gram model, while feedforward networks encode GPS trajectories, enabling precise distant trajectory predictions. Rigorous testing on Manhattan and Porto datasets showcases the superiority of our model over existing approaches, affirming its robustness and applicability in real-world scenarios. Looking ahead, our focus will shift towards integrating diverse data sources, including buses, subways, and other modes of transportation. Moreover, we aim to investigate the impact of COVID-19 on the transportation system, endeavouring to aggregate datasets from various regions to analyze human mobility patterns comprehensively. Additionally, we plan to expand our analysis by incorporating additional points of interest (POI) from platforms like FourSquare or external sources. Furthermore, we intend to explore using POI graphs instead of individual points to enhance the visualization of human mobility and improve the accuracy of predicting future destinations. This study identifies limitations requiring further investigation in future research. We aim to expand evaluation metrics and integrate destination prediction with travel time estimation, taxi dynamics, and MOD systems. Our primary goal is to improve destination prediction accuracy by integrating geographical information, reducing customer and driver waiting times, optimizing taxi services, and aiding urban transport planning. Despite our method's accuracy, challenges like road congestion and air pollution persist. We propose leveraging AI and deep learning to address these issues.

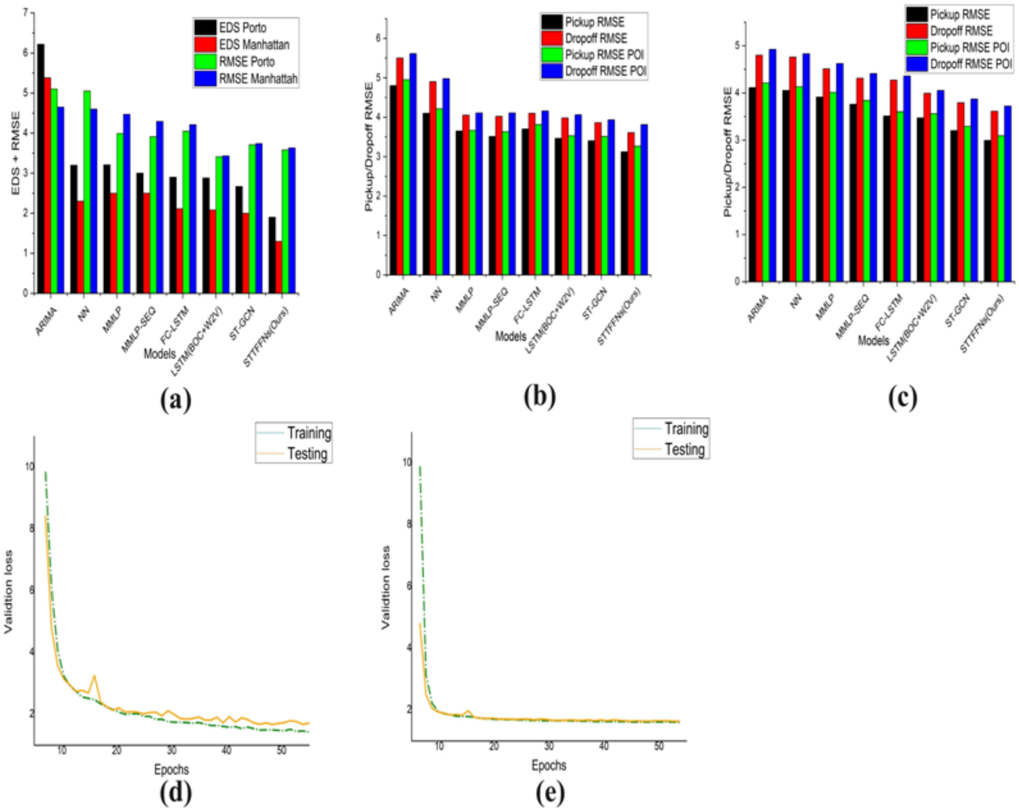

Fig. 4. a,b and c dataset statistics analysis, d and e training and testing analysis.

Acknowledgement

The Jiangsu Provincial Government Excellent Program of China (2023ZB893) partly supported the work. Any opinions, findings, and conclusions expressed here are those of the authors and do not necessarily reflect the funding agencies' views.

References

[1] M. Perić, "Estimating the Perceived Socio-Economic Impacts of Hosting Large-Scale Sport Tourism Events," *Social Sciences*, vol.7, no.10, 2018. Article(CrossRefLink)

[2] M. Geng, Y. Chen, Y. Xia, X. (Michael) Chen, "Dynamic-learning spatial-temporal Transformer network for vehicular trajectory prediction at urban intersections," *Transportation Research Part C: Emerging Technologies*, vol.156, 2023. Article(CrossRefLink)

[3] Y. Wei et al., "A review of data-driven approaches for prediction and classification of building energy consumption," *Renewable and Sustainable Energy Reviews*, vol.82, pp.1027-1047, 2018. Article(CrossRefLink)

[4] Y. Yang, X. Xiong, Y. Yan, "UAV Formation Trajectory Planning Algorithms: A Review," *Drones*, vol.7, no.1, 2023. Article(CrossRefLink)

[5] J. Liu, K. Han, X. (Michael) Chen, G. P. Ong, "Spatial-temporal inference of urban traffic emissions based on taxi trajectories and multi-source urban data," *Transportation Research Part C: Emerging Technologies*, vol.106, pp.145-165, 2019. Article(CrossRefLink)

[6] D. Feng, F. Zhou, Q. Wang, Q. Wu, and B. Li, "Efficient Aggregate Queries on Location Data with Confidentiality," *Sensors*, vol.22, no.13, 2022. Article(CrossRefLink)

[7] X. Jiang, E. Barnett, and C. Gosselin, "Dynamic Point-to-Point Trajectory Planning beyond the Static Workspace for Six-DOF Cable-Suspended Parallel Robots," *IEEE Transactions on Robotics*, vol.34, no.3, pp.781-793, 2018. Article(CrossRefLink)

[8] S. Liu, S. Huang, X. Xu, J. Lloret, K. Muhammad, "Efficient Visual Tracking Based on Fuzzy Inference for Intelligent Transportation Systems," *IEEE Transactions on Intelligent Transportation Systems*, vol.24, no.12, pp.15795-15806, 2023. Article(CrossRefLink)

[9] N. A. N. M. N. Azman, N. H. Abd Rahman, S. S. Md Sawari, S. A. Abas, S. A. A. Latif, "The tourists' spatial behaviour and tourist movement pattern in Muar Johor," *Planning Malaysia*, vol.19, no.2, pp.275-286, 2021. Article(CrossRefLink)

[10] K. Mangalam et al., "It Is Not the Journey But the Destination: Endpoint Conditioned Trajectory Prediction," in *Proc. of Computer Vision – ECCV 2020*, vol.12347, pp.759-776, 2020. Article(CrossRefLink)

[11] D. Sternad and S. Schaal, "Segmentation of endpoint trajectories does not imply segmented control," *Experimental Brain Research*, vol.124, no.1, 1999. Article(CrossRefLink)

[12] X. Qin, Z. Li, K. Zhang, F. Mao, X. Jin, "Vehicle Trajectory Prediction via Urban Network Modeling," *Sensors*, vol.23, no.10, 2023. Article(CrossRefLink)

[13] J. F. W. Zaki, A. M. T. Ali-Eldin, S. E. Hussein, S. F. Saraya, and F. F. Areed, "Time Aware Hybrid Hidden Markov Models for Traffic Congestion Prediction," *International Journal on Electrical Engineering and Informatics*, vol.11, no.1, 2019. Article(CrossRefLink)

[14] S. Y. Han, Q. Zhao, Q. W. Sun, J. Zhou, Y. H. Chen, "EnGS-DGR: Traffic Flow Forecasting with Indefinite Forecasting Interval by Ensemble GCN, Seq2Seq, and Dynamic Graph Reconfiguration," *Applied Sciences*, vol.12, no.6, 2022. Article(CrossRefLink)

[15] L. Wu, X. Wei, L. Meng, S. Zhao, and H. Wang, "Privacy-preserving location-based traffic density monitoring," *Connection Science*, vol.34, no.1, pp.874-894, 2022. Article(CrossRefLink)

[16] P. Pokorny, B. Skender, T. Bjørnskau, and M. P. Hagenzieker, "Video observation of encounters between the automated shuttles and other traffic participants along an approach to right-hand priority T-intersection," *European Transport Research Review*, vol.13, 2021. Article(CrossRefLink)

[17] S. Liu, S. Huang, S. Wang, K. Muhammad, P. Bellavista, J. Del Ser, "Visual tracking in complex scenes: A location fusion mechanism based on the combination of multiple visual cognition flows," *Information Fusion*, vol.96, pp.281-296, 2023. Article(CrossRefLink)

[18] H. Rong, A. P. Teixeira, C. Guedes Soares, "Maritime traffic probabilistic prediction based on ship motion pattern extraction," *Reliability Engineering & System Safety*, vol.217, 2022. Article(CrossRefLink)

[19] S. Liu et al., "Human Inertial Thinking Strategy: A Novel Fuzzy Reasoning Mechanism for IoT-Assisted Visual Monitoring," *IEEE Internet of Things Journal*, vol.10, no.5, pp.3735-3748, 2023. Article(CrossRefLink)

[20] A. Rossi, G. Barlacchi, M. Bianchini, B. Lepri, "Modelling Taxi Drivers' Behaviour for the Next Destination Prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol.21, no.7, pp.2980-2989, 2020. Article(CrossRefLink)

[21] L. Zhao et al., "T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol.21, no.9, pp.3848-3858, 2020. Article(CrossRefLink)

Zain Ul Abideen, currently serving as a Postdoctoral Researcher at the Automotive Engineering Research Institute, Jiangsu University, brings a wealth of exper- tise in the realm of Computer Science. Graduating with a PhD in Computer Science from Xi'an Jiaotong University, China in 2022, his doctoral research focused on deep learning applications within the domains of urban computing and Intelligent transpor- tation systems (ITS) with spatial-temporal features. During his doctoral tenure, Zain conducted extensive research exploring the intricacies of traffic flow pre- diction, the utilization of deep learning methodologies in understanding small city dynamics, and thintegration of spatial-temporal features for enhancing intelligent transportation systems. His work has gar- nered recognition within academic circles for its inno- vative approaches and potential to address pressing urban challenges.

Xiaodong Sun (Senior Member, IEEE) received the BSc degree in Electrical Engineering and the MSc and PhD degrees in Control Engineering from Jiangsu University, Zhenjiang, China, in 2004, 2008, and 2011, respectively. Since 2004, he has been with Jiangsu University, where he is currently a Professor in vehi- cle engineering with the Automotive Engineering Research Institute. From 2014 to 2015, he was a Visit- ing Professor with the School of Electrical, Mechani- cal, and Mechatronic Systems, University of Technology Sydney, Sydney, Australia. His current teaching and research interests include electrified vehicles, electrical machines, electrical drives, and energy management. He is the author or coauthor of more than 100 refereed technical papers and one book, and he is the holder of 42 patents in his areas of interest. Dr. Sun is an Associate Editor of IEEE Trans- actions on Industrial Electronics, an Associate Editor of IEEE Transactions on Transportation Electrifica- tion, and an Editor of IEEE Transactions on Energy Conversion.

Chao Sun was born in Nangtong, Jiangsu, China, in 1995. He received the BSc degree in Electrical Engi- neering from Yangzhou University, Yangzhou, China, in 2018, and the MSc degree in Electrical Engineering in 2022 from Jiangsu University, Zhenjiang, China, where he is currently working toward the PhD degree in Vehicle Engineering. His current research interests are the bearingless induction motor and its intelligent control technology.

Hafiz Shafiq Ur Rehman Khalil is a PhD candidate at Xian Jiaotong University in China. His Areas of research are machine learning, deep learning, and information security.