

MODELING SPATIO-TEMPORAL DYNAMICS WITH TRANSFORMER ATTENTION FOR POINT OF INTEREST RECOMMENDATION

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF

**MASTER OF TECHNOLOGY
IN
COMPUTER SCIENCE & ENGINEERING**

Submitted By
SUDEV P S
23/CSE/27

Under the supervision of

Dr. NIPUN BANSAL
(Assistant Professor)



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

May 2025

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College Of engineering)
Bawana Road, Delhi-110042

CANDIDATE’S DECLARATION

I, Sudev PS, Roll Number 23/CSE/27, student of M.Tech (Computer Science & Engineering), hereby declare that the project dissertation titled **“MODELING SPATIO-TEMPORAL DYNAMICS WITH TRANSFORMER ATTENTION FOR POINT OF INTEREST RECOMMENDATION”** which is submitted by me to the Department of Computer Science and Engineering, Delhi Technological University, Delhi, in partial fulfilment for the requirements of the award of degree of Master of Technology in Computer Science and Engineering is original and not copied from any source without proper citation. The material contained in this Report has not been submitted at any other University or Institution for the award of any degree.

Place: Delhi

Sudev PS

Date:

23/CSE/27

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of our knowledge

Signature of Supervisor (s)

Signature of External Examiner

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College Of engineering)
Bawana Road, Delhi-110042

CERTIFICATE

Certified that **Sudev PS (23/CSE/27)** has carried out their search work presented in this project dissertation titled **“MODELING SPATIO-TEMPORAL DYNAMICS WITH TRANSFORMER ATTENTION FOR POINT OF INTEREST RECOMMENDATION”** for the award of **Master of Technology** from Department of Computer Science & Engineering, Delhi Technological University, Delhi, under my supervision. The thesis embodies results of original work and studies are carried out by student himself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution

Place: Delhi

Date:

DR. NIPUN BANSAL
SUPERVISOR

ACKNOWLEDGEMENT

I am extremely grateful to my project guide, **DR. NIPUN BANSAL**, Assistant Professor, Department of Computer Science and Engineering, Delhi Technological University, Delhi for providing invaluable guidance and being a constant source of inspiration throughout my research. I will always be indebted to him for the extensive support and encouragement she provided.

I am highly indebted to the panel faculties during all the progress evaluations for their guidance, constant supervision and for motivating me to complete my work. They helped me throughout by giving new ideas, providing necessary information and pushing me forward to complete the work.

Sudev PS

23/CSE/27

ABSTRACT

Point-of-Interest (POI) recommendation is one of the most important tasks in location-based services to recommend individuals locations based on their past check-ins, spatial interests, and temporal patterns. In this work, we introduce a new method that learns spatio-temporal dynamics via Transformer-based attention to enhance recommendation precision. We represent user and POI IDs via embedding layers and bring geographical context in by normalizing and embedding latitude-longitude points. Temporal relationships between user check-in sequences are modelled with a Transformer Encoder to allow for parallel sequence modelling and learning of distant interactions. Temporal information is combined with user and spatial representations to provide a common latent feature space, which is then passed through a fully connected layer to provide POI probability scores. The model is trained with negative log-likelihood loss and optimizes Adam with gradient clipping for stability. Evaluation by Accuracy@k, Precision@k, Recall@k, F1@k and NDCG@k presents ranking performance of the proposed model on effective POIs. The proposed approach yields an interpretable and scalable solution to next-POI recommendation with deep consideration of spatial, temporal, and behaviour patterns collectively.

CONTENTS

CANDIDATE’S DECLARATION.....	i
CERTIFICATE.....	ii
ACKNOWLEDGEMENT.....	iii
ABSTRACT.....	iv
CONTENTS.....	v
LIST OF TABLES.....	vii
LIST OF FIGURES.....	viii
LIST OF SYMBOLES.....	ix
1. INTRODUCTION.....	1
1.1 Transformer	3
2. BACKGROUND.....	6
2.1 Point Of Interest recommendation.....	6
2.2 Dataset.....	7
2.3 Data Pre-processing.....	8
3. LITERATURE REVIEW	10
4. LITERATURE SURVEY.....	14
4.1 Introduction to Literature Survey.....	14
4.2 History.....	14
4.3 existing methods	15
4.4 Challenges.....	18
4.5 Future Directions.....	20

4.6 Applications.....	21
5. METHADODOLOGY.....	22
5.1 Embedding layers.....	22
5.2 Transformer encoder block.....	23
5.3 Embedding Fusion Module.....	24
5.4 Output Layer.....	24
5.5 Training Loop.....	25
6. RESULT AND DISCUSSION.....	26
6.1 Evaluation Metrics.....	26
6.2 Training and Test Results.....	27
7. CONCLUSION AND FUTURE SCOPE.....	32
REFERENCES.....	33

LIST OF TABLES

Table 1: Statistics about dataset.....	7
Table 2: Sample data from New York dataset.....	7
Table 3: Sample data from Gowalla dataset.....	8
Table 4: Result of STDT model on different dataset.....	30

LIST OF FIGURES

Fig 1: Google map and Yelp.....	1
Fig 2: Foresquare social app.....	2
Fig 3: The encoder-decoder structure of the Transformer architecture Taken from [3].....	5
Fig 4 Example showing the POI Recommendation.....	14
Fig 5: overall architecture of STDT model.....	22
Fig 6: Variation of evaluation metrics on New York Dataset.....	28
Fig 7: Variation of evaluation metrics on Gowalla Dataset.....	29

LIST OF SYMBOLS

Symbol	Description
u	User ID (a unique identifier for each user)
p	POI ID (a unique identifier for each Point of Interest)
g	Geographic coordinates (a vector containing latitude and longitude)
e_u	User embedding vector representing the user's preferences
e_g	Geo embedding vector derived from latitude and longitude
E_p	Sequence of POI embedding's (for T visited POIs)
W_g	Weight matrix for geo embedding projection (size: $d \times 2d$)
b_g	Bias term for geo embedding projection (size: d)
d	Embedding dimension (a fixed number like 64 or 128)
T	Length of the POI sequence for a user
H	Output of Transformer encoder (sequence representations)
e_t	Representation of temporal dynamics (last output in Transformer)
Q, K, V	Query, Key, and Value matrices in self-attention
W_Q, W_K, W_V	Weight matrices for Q, K, and V (each of size $d \times d$)
d_k	Dimensionality of the key vectors used in attention (usually $d_k = d$)
$softmax$	Normalization function used in attention
f	Final fused feature vector combining user, spatial, and temporal info
W_f	Weight matrix for feature fusion projection (size: $d \times 3d$)
b_f	Bias for the feature fusion layer (size: d)
$ReLU$	Rectified Linear Unit activation function
\hat{Y}	Predicted POI distribution (probabilities over all POIs)
W_o	Output projection weights (size: $N \times d$)
b_o	Output projection bias (size: N)
N	Total number of POIs in the dataset
\mathcal{L}	Loss function used for training
\hat{Y}_{p^+}	Predicted probability of the correct POI
p^+	Index of the ground truth POI (the correct next POI)
$-\log(.)$	Negative log-likelihood (used to penalize incorrect predictions)

CHAPTER – 1 INTRODUCTION

With the rapid expansion of mobile apps and growing ubiquity of location-aware technology, Point-of-Interest (POI) recommendation systems have become an inherent part of today's location-based services (LBS), i.e., Foursquare, Google Maps, Yelp. POI recommendation systems are designed to suggest suitable spots — e.g., restaurants, parks, museums, or shopping malls — suitable for a user's profile and situational contexts.

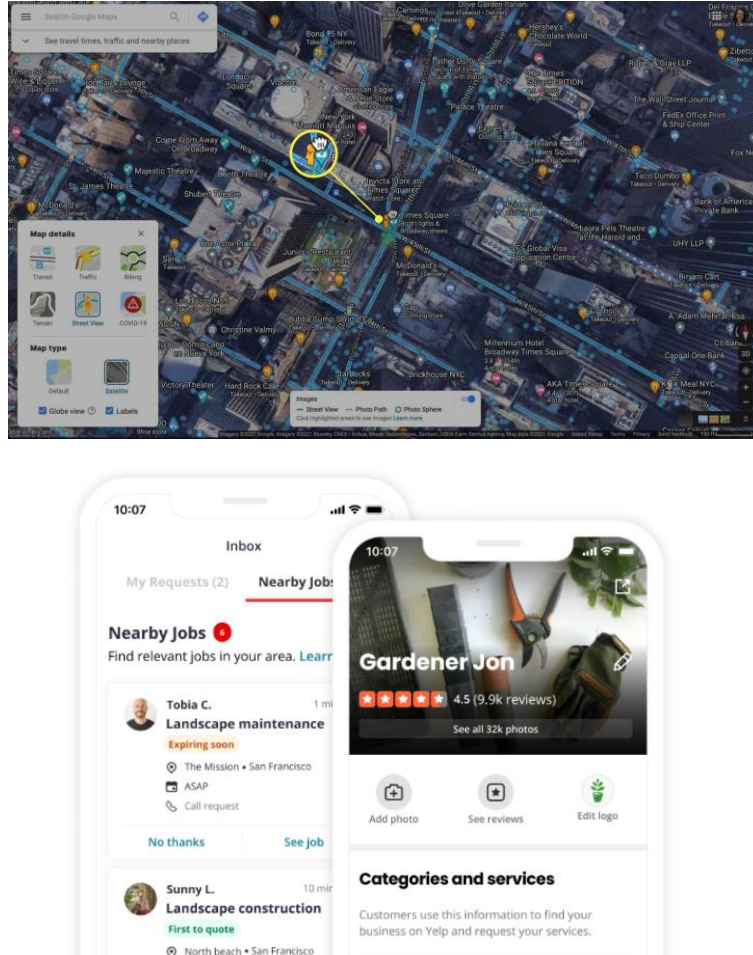


Fig. 1 Google map and Yelp

In contrast to conventional recommendation methods based on user-item interaction information (e.g., movie or item ratings), POI recommendation is characterized by distinctive challenges stemming from the dynamic and high-dimensional nature of user behaviour, which is inherently subject to spatial and temporal influences. For instance, a user can have different favourite locations at different times of a day or day of the week, and geographic proximity is a key factor in the probability to visit a POI. Thus, modelling both spatial relations (geographical locality and proximity) and temporal patterns (periodicity, regency, and time-sensitive preferences) is necessary to improve the accuracy of POI recommendations.

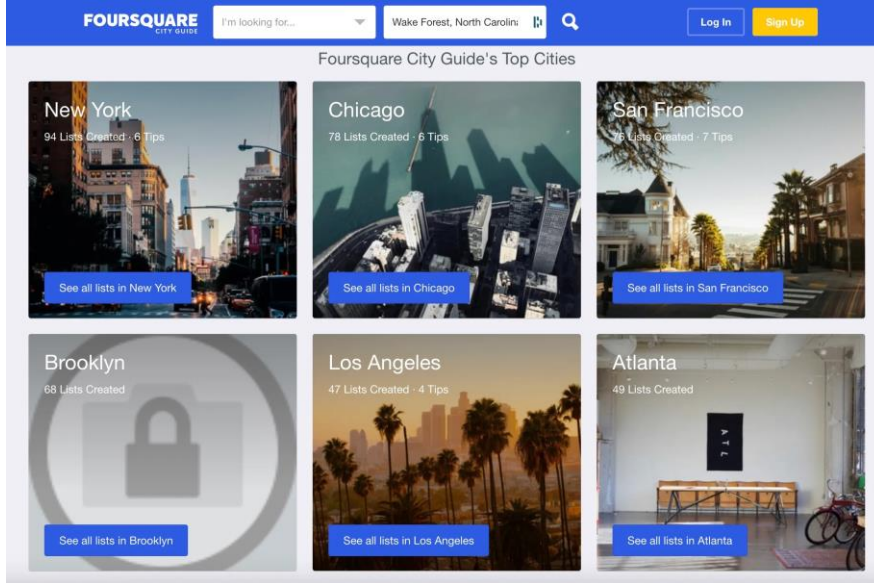


Fig. 2 Foursquare Social app

Most of the available methods in POI recommendation have a bias towards using sequence modelling methods like Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and attention mechanisms in sequential behaviour modelling. Although these models have yielded encouraging outcomes, they suffer from the disadvantage of being unable to capture long-range dependencies owing to their sequential nature and the fact that, in general, they parallelize and scale badly. Moreover, most of the models either apply spatial features inefficiently or independently of temporal sequences and do not capture the subtle interaction between where and when a user is in a location. To overcome these limitations, this research introduces a Transformer-based neural model that merges spatio-temporal modelling into a single framework via self-attention to capture global inter-dependencies across user check-in sequences efficiently.

introduced model, STDT (spatio-temporal dynamics with transformer), provides an end-to-end process for next-POI recommendation. The method starts from pre-processing check-in data, i.e., encoding user IDs and POI IDs into categorical integer indices that can be used as inputs for embedding layers. Geographical coordinates (latitude and longitude) are normalized and embedded by linear transformation to place them in the same latent feature space. The center of the model is a Transformer Encoder Block, replacing LSTM-based temporal modeling with multi-head self-attention for capturing contextual relationships among POIs along a user's past trajectory. This enables more expressive long-term dependency modeling with parallel computation support, making the method highly scalable.

The final representation is obtained by concatenating three groups of features: user embeddings, geo-coordinate spatial embeddings, and the Transformer representation of temporal sequences. The combined feature vector maintains user-specific, spatial, and sequential user behavior information and is sent through a fully connected layer to output POI recommendation scores.

It is tuned with negative log-likelihood loss and Adam optimizer, and training stability is achieved with gradient clipping. It is evaluated with popular ranking metrics like Accuracy@n and NDCG@n that estimate prediction accuracy and ranking quality of the recommended POIs.

1.1 Transformer

The Transformer is a deep learning model presented in the 2017 paper [3]. In contrast to earlier models, which used recurrent (RNNs) or convolutional (CNNs) layers, the Transformer uses self-attention mechanisms to process sequential data in an efficient manner. This architecture is extremely parallelizable and, as such, quicker to train and better suited to deal with long-range dependencies in data. It is the basis for current AI models such as GPT, BERT, and T5.

The Transformer has two key components: the encoder (which processes input data) and the decoder (which produces output). Both of them are composed of several identical layers consisting of self-attention mechanisms, feed-forward networks, and normalization layers. The model also employs positional encoding to preserve sequence order because it does not process data sequentially like RNNs.

Encoder is a stack of N identical layers (typically 6 in base paper). Two main sub-layers per layer are: Multi-Head Self-Attention – Calculates relations between all input words by producing queries (Q), keys (K), and values (V). Feed-Forward Neural Network (FFN) – A simple fully connected network applied to each position independently.

Each sub-layer is then followed by.. layer normalization and residual connections.. for stabilizing training. Self-attention enables the model to put weights on relative word importance across words in a sentence. It does so by:

- Calculating attention scores for all pairs of words.
- Performing a softmax to obtain attention weights.
- Calculating a weighted sum of values (V) based on those weights.

The multi-head version operates several attention mechanisms in parallel to assist the model in attending to various things (i.e., syntax, semantics).

The decoder also consists of N identical layers, but with an additional attention mechanism:

1. Masked Multi-Head Self-Attention – Prevents the decoder from attending to tokens that follow (not precede) when it is producing.
2. Cross-Attention – Along with the decoder from the encoder output, it becomes able to attend to relevant input information.
3. Feed-Forward Network – Similar to the encoder, but position-wise applied.

Because Transformers do not read data sequentially, they require a method to encode word order. Positional encodings are added to the input embeddings, either in the form of fixed sinusoidal functions or learned embeddings. This allows the

model to differentiate between sequences such as “The cat chased the dog” and “The dog chased the cat”.

The decoder output goes through a linear layer and a softmax activation, producing probabilities for the next word in the sequence. This enables the model to produce words one word at a time in an autoregressive fashion (similar to GPT).

Key Strengths of the Transformer Parallel Processing – Unlike RNNs, Transformers process all the tokens at once, which accelerates training. Long-Range Dependencies – Self-attention better captures relationships between far-away words compared to RNNs or CNNs. Scalability– Equally effective on short and long sequences. Flexibility – Applied in encoder-only (BERT), decoder-only (GPT), and encoder-decoder (T5)models.

The Transformer's self-attention-based structure revolutionized NLP by allowing for faster, more efficient, and scalable deep models. Its structure has facilitated advances in machine translation, text generation, and even computer vision (e.g., Vision Transformers). This project contributes to existing research by showing that Transformer-based models, when paired properly with spatial data, have the capacity to surpass standard recurrent approaches in POI recommendation. Attention mechanism not only enhances performance but also adds interpretability by proposing which previous visits made the most contribution to an influence on a recommendation. Through large-scale experiments, we determine that our model has competitive performance on real-world datasets, thereby confirming its efficacy in capturing complex spatio-temporal patterns underlying user mobility. Our approach offers novel methods of embracing attention-based architectures for location-based personalized services and provides a flexible and extensible platform for future enhancements.

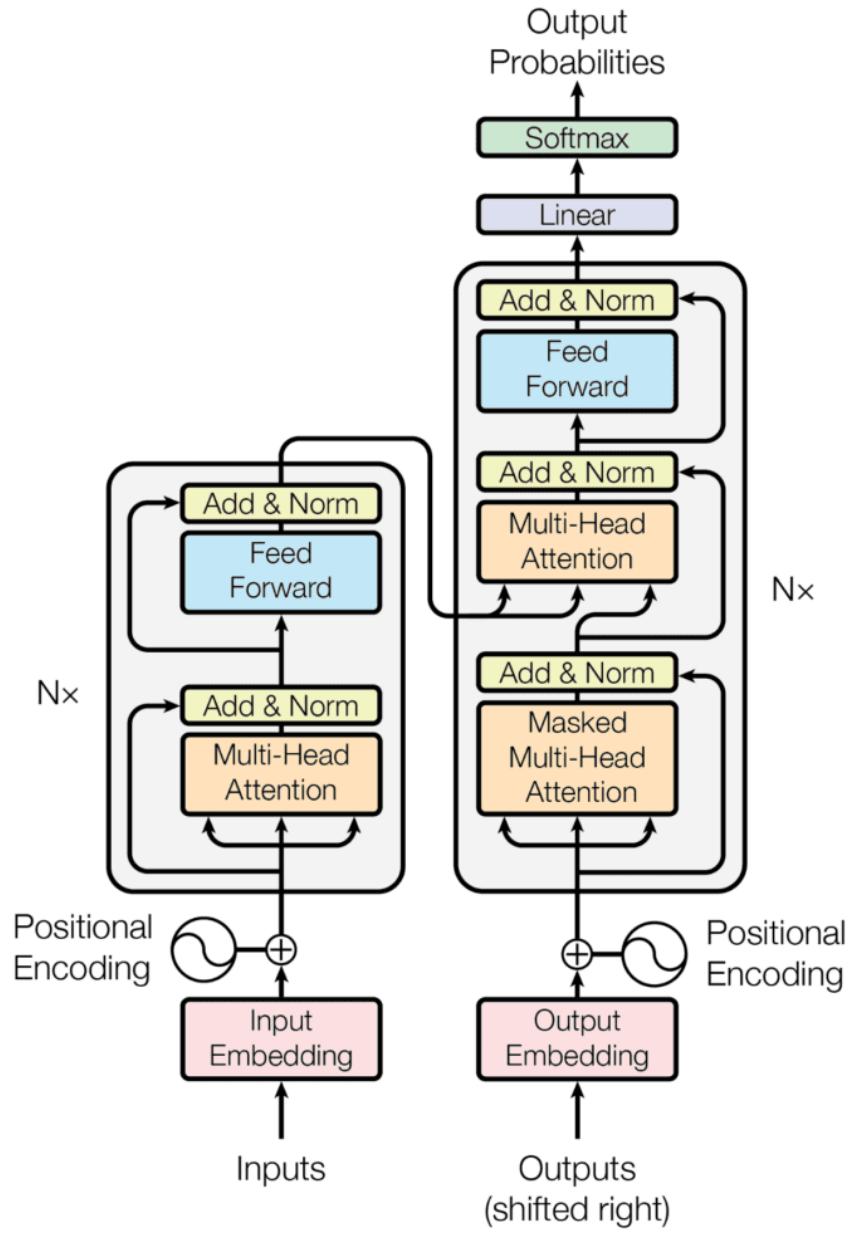


Fig 3: The encoder-decoder structure of the Transformer architecture Taken from [3]

CHAPTER – 2 BACKGROUND

2.1 Point Of Interest Recommendation

Location-Based Social Networks like Foursquare, Yelp, and Facebook Places have experienced tremendous growth, resulting in the ubiquity of location-aware user check-in, review, and geotagged information. The high-quality data stream presents a superior opportunity to capture mobility patterns and user interests, hence personalized Point-of-Interest (POI) recommendation is a highly valued application. Future POI recommendation, specifically, involves predicting a user's probable future location from his or her past movement and context information. The task is extremely useful in that it improves user experience in the form of customized recommendations and, on the other hand, aids businesses with more effective location-based marketing.

Nonetheless, the task is intrinsically difficult because of the sparsity in user check-in information and the intricate spatial-temporal relationship of human mobility. Users' follow-up behaviors are typically regulated by numerous dynamic variables, including time, geography, social setting, and individual taste, which must be accurately modeled to enable accurate predictions. Classical recommendation methods like collaborative filtering and matrix factorization do not account for these intricacies. As a result, more recent studies have moved towards deep learning-based models, i.e., Recurrent Neural Network (RNN) and embedding-based models, to learn sequential and contextual patterns from user behavior data.

Although there has been tremendous progress, there are some limitations. Most of the current models either model context and sequential prediction separately or fail to properly address the significance of heterogeneous contextual information. Subsequent models, like NeuNext and TMCA, have tried to fill these loopholes by incorporating context learning and sequential modeling through sophisticated neural architectures like LSTMs, attention networks, and graph-based embeddings. These models are a tremendous step towards optimal utilization of both labeled and unlabeled data and fusion of multi-level contextual elements in the recommendation pipeline. This research is based on such developments, and it aims to investigate and contrast the performance of cutting-edge next POI recommendation models in discovering rich mobility patterns and context-sensing behavior under real-world situations

2.2 Dataset

This study evaluate model on two widely used real-world datasets: New York (Foursquare) and Gowalla, collected from Liu et al. [1] and Zhao et al.[2], respectively. The statistics of these datasets are Summarized in TABLE I.

Statistics	New York	Gowalla
Check-ins	194108	456988
Users	2321	10162
POIs	5596	24250
Avg no of Check-ins per user	83.6	44.97
Avg no of Check-ins per user	34.7	18.8
Check-in period	04/2012 - 02/2013	03/2009 – 10/2009

Table 1: Statistics about dataset

New York: This dataset contains 194,108 check-ins from 2,321 users and 5,596 POIs in New York, collected between July 2010 and August 2011. The sample of dataset is shown in Table 2. Every entry in this data set is a user ID, venue ID, geographic coordinates (latitude and longitude), timestamp, and engineered features G, T, and C. G is the most appropriate geographical distance metric for user mobility, T denotes temporal periodicity and recency, and C is a context-awareness measure incorporating user and location visit frequency. This information gives us a more geographically representative dataset of check-ins so that user mobility patterns and venue selection may be analyzed with greater detail within the city. It is also a good resource for testing of generalization for POI models to beyond an individual cluster area

user_id	venue_id	latitude	longitude	timestamp	G	T	C
0	536	40.64509	-73.7845	1344616807	1.33046	0	0.622385
0	376	40.77143	-73.9735	1336848038	289.2422	0.00422	0.622385
0	3436	40.71942	-74.0103	1334765223	308.7166	0	0.622385
0	5445	40.72516	-73.9922	1335025691	290.1269	0.002341	0.745133
0	149	40.77436	-73.9817	1337123923	330.7642	0.005018	0.622385
0	1787	40.78864	-73.9742	1338240022	250.0739	0.020462	0.814469
0	6235	40.7202	-74.0054	1339090925	359.6289	0	0.677173
0	6235	40.7202	-74.0054	1337955046	359.6289	0	0.677173
0	6235	40.7202	-74.0054	1343137627	359.6289	0	0.677173

Table 2: Sample data from Newyork dataset

Gowalla: The Gowalla dataset is a real-world check-in dataset gathered from the since-shuttered location-based social networking site, This dataset includes 456,988 check-ins from 10,162 users and 24,250 POIs in California and Nevada, collected between February 2009 and October 2010. Gowalla. The sample of dataset is shown in Table 3 It has user check-in information like the user ID, timestamp, latitude, longitude, and venue ID of every check-in. We also extracted other spatial and temporal features denoted by the columns S, G, and T for this project. In particular, the spatial score (S) captures the user's past preference for a place, the geographical impact (G) captures the spatial proximity of places, and the temporal context (T) captures the periodical pattern or temporal importance of the checking-in time. The spatiotemporal behavior patterns are dense in the dataset and appropriate for modeling individualized next Point-of-Interest (POI) prediction. All of these check-ins in this processed subset share the same user and venue, which provides a controlled setting for time-based user behavior analysis.

user_id	timestamp	latitude	longitude	venue_id	S	G	T
0	1287440263	30.2691	-97.7494	10155	0.998455	0.032331	0.059896
0	1286912680	30.2691	-97.7494	10155	0.998455	0.032331	0.059896
0	1286896743	30.2691	-97.7494	10155	0.998455	0.032331	0.059896
0	1286407534	30.2691	-97.7494	10155	0.998455	0.032331	0.059896
0	1286295035	30.2691	-97.7494	10155	0.998455	0.032331	0.059896
0	1285599121	30.2691	-97.7494	10155	0.998455	0.032331	0.059896
0	1285363933	30.2691	-97.7494	10155	0.998455	0.032331	0.059896
0	1284591853	30.2691	-97.7494	10155	0.998455	0.032331	0.059896
0	1283271541	30.2691	-97.7494	10155	0.998455	0.032331	0.059896

Table 3: Sample data from Gowalla dataset

2.3 Data Pre-processing

Preprocessing of data in this POI recommendation system requires two significant conversions to support the input data preparation for the training of deep learning models. First, categorical codes such as user_id and venue_id are converted to numerical values through Pandas' `astype('category').cat.codes`, where each unique category is associated with an integer value. This conversion is necessary as neural networks, especially embedding layers, need integer indices to map these discrete objects into dense, trainable vector representations that preserve latent similarities and behaviors. Otherwise, the model would be unable to handle categorical string data effectively. Second, the geographical coordinates—latitude and longitude—are normalized to lie in a range [0, 1]. This normalization helps to prevent such continuous spatial properties from overwhelming the learning process as a result of scale differences, particularly since raw lat-long coordinates can be very large. Scaled uniformly, we facilitate more rapid

model convergence and more fair learning across input dimensions so that the geo-embedding layer may learn meaningful spatial relationships without prejudice. These preprocessing operations collectively ensure that continuous and categorical inputs are properly formatted and scaled for subsequent embedding and modeling operations

`CheckInDataset` is a PyTorch special `Dataset` class that steps in to handle and supply preprocessed check-in data to the model for training as well as for testing. It encapsulates raw tabular data (usually a `DataFrame`) into a PyTorch dataset-friendly format so that the data may be readily employed with PyTorch's `DataLoader` for efficient mini-batch training. Inside the `__getitem__` method, each point of data is indexed and transformed into a dictionary of all required elements: the `user_id`, `venue_id`, and `geo_features` (normalized longitude and latitude). These are returned as tensors, the native input format of neural networks in PyTorch. This modularity makes the model efficient at sampling batches, randomizing data during training for the purpose of generalization, and having standard input formats. Thus, the `CheckInDataset` class offers a clean, reusable, and adaptable mechanism for processing input data so that any batches passed to the model include appropriately processed user, POI, and location data to facilitate learning of spatio-temporal patterns.

CHAPTER – 3 LITERATURE REVIEW

In paper [5] explains that In the last several years, Point-of-Interest (POI) recommendation has been one of the major services in Location-Based Social Networks (LBSNs) which can supply users with personalized recommendations based on their location, interests, and behaviors. Many models are put forward to promote the performance and precision of POI recommendations as reflected in literature. The earlier approaches were predominantly matrix factorization and collaborative filtering techniques, while recent approaches have started incorporating social influence, spatial-temporal behavior, and deep learning-based frameworks. Nevertheless, there is no systematic, full-scale comparison of these models under consistent conditions. Recent studies tend to consider specific data or measures for testing, such that it becomes difficult to make general conclusions. This thesis expands upon existing research through the comprehensive review of 11 advanced POI recommendation models, thereby adding to a better knowledge of the model strengths and limitations and their usability in various real-world settings. In [6] Traditional POI suggesting techniques primarily focus on modeling personal preferences, dependent on social influence, and considering spatial proximity to enhance recommendation accuracy. These techniques like collaborative filtering, matrix factorization, and graph-based techniques all focus on capturing different aspects of user activities and spatial aptness. Yet, among the biggest shortcomings of these methods is that they cannot learn how to adapt to temporal contexts, especially when users want recommendations for a particular time period. Recent research tried to incorporate temporal dynamics but tend to use time as a static feature or as a bare timestamp, without considering the high-level temporal patterns in users' activities. This dynamic time-aware recommendation shortfall led to the creation of sophisticated models taking sequential check-in patterns and time periods into consideration.

Following [7] POI recommendation issue has been highly prominent in the mobile social network community because of the potential to improve location-based personalized services. Conventional methods of POI recommendation based on collaborative filtering, matrix factorization, or Markov chain-based models tended to suffer from a lack of ability to capture sophisticated sequential and contextual user behavior patterns. To overcome these constraints, recent research has investigated deep learning methods, i.e., recurrent neural networks (RNNs), that are naturally proficient in learning sequential patterns. A number of research studies have also employed spatial-temporal features and influence of social elements in prediction to enhance the accuracy.

The application of deep learning approaches to recommender systems has registered tremendous progress in recent years. [8] Session-based recommendation, where the system

makes the next item a prediction in an anonymous, brief sequence of user behavior, has been extremely popular. Initial studies were mostly based on collaborative filtering and content-based methods, which failed for short-term and sequential behavior. The introduction of recurrent neural networks (RNNs), particularly GRU4REC by Hidasi et al., was the breakthrough towards utilizing temporal dependencies across sessions using Gated Recurrent Units. Although revolutionary, it is still challenging to evaluate the performance of such deep models because of unstable evaluation protocols and poor baselines in most research. Recent empirical comparisons have confirmed that heuristic methods, including session-based k-nearest neighbors (kNN), outperform GRU4REC in most scenarios, especially when combined with neighborhood sampling and efficient data structures. Of special interest, the integration of RNNs with kNN techniques has been shown to perform even better, demonstrating the complementary strengths of neural and co-occurrence-based methods. This increasing body of literature emphasizes the need for robust benchmarking and hybrid approaches to truly make advances on session-based recommendation. In [9] New methods, such as deep learning models like Recurrent Neural Networks (RNNs) and attention mechanisms, have begun to address these gaps by better capturing user check-in order and context. Challenges remain, though, in determining which of the user's previous check-ins are actually relevant for predicting the correct thing. This thesis addresses these limitations by presenting a Geographically-Temporally aware Hierarchical Attention Network (GT-HAN) that models both spatial-temporal impact and high-fidelity POI-POI relationships to create an enriched portrait of user mobility for more precise POI recommendation. The research work [10] explains classic solutions have privileged Markov models or probabilistic approaches to describe sequential user movement but are inefficient when dealing with intricate temporal relationships and sparse data. With the creation of deep learning in the form of Recurrent Neural Networks (RNNs), hopes were raised with effective abstraction of sequential patterns from user movement traces. Unfortunately, in natural settings with user traces exhibiting sparsity and heterogeneity by their nature, vanilla RNNs are inefficient. Trying to revert the same, the research community enriched RNNs by adding spatiotemporal contexts—utilization of context-aware gates or switch matrices—in more effectively capturing periodicity and location semantics. In spite of such improvements, most models remain short of completely exploiting long-range dependencies and rich contextual information of previous data. Following Point-of-Interest (POI) recommendation has garnered significant interest because it can provide enhanced user experience and increase business activity in Location-Based Social Networks (LBSNs). [11] Most of the existing traditional methods are bound to fail to address the issues of sparsity and complexity of sequential user behaviors in check-in data. Therefore, various methods have been put forward, such as probabilistic models, matrix factorization, and recently, embedding-based

methods that strive to represent contextual information well. Recurrent Neural Networks (RNNs), especially LSTM and GRU variants, have been demonstrated to achieve impressive success in learning user mobility patterns and temporal dependencies. Most current models, however, separate the learning of embeddings and prediction task as two distinct processes, which prevents them from fully exploiting contextual signals. These efforts have sought to leverage auxiliary information like geographical closeness, social bias, and temporal patterns, but in ad-hoc or loosely coupled fashions. This thesis tries to advance on these efforts by investigating a consistent neural structure that combines context modeling and sequential prediction with the goal of leveraging both labeled and unlabeled data more effectively towards better recommendation accuracy. The recent progress of deep learning, i.e., the use of Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) models, has enhanced modeling of sequential spatial-temporal patterns. However, sparsity in the data is still an old problem. In opposition to this, some of the work proposed hybrid models that integrate spatial and temporal dynamics, and some try hierarchical or context-aware approaches. Based on this [12], the ST-LSTM that has been proposed and its extension HST-LSTM are designed to promote prediction performance through successful learning of short-term behavior and long-term contextual dependency in trajectory data. Proper modeling of user mobility behavior is a major issue for future Point-of-Interest (POI) recommendation systems, where spatial and temporal context information plays an important role. Existing methods like Factorizing Personalized Markov Chains (FPMC) introduced personalized transition modeling but made strong independence assumptions among influence factors, thus constraining their performance. [13] Tensor Factorization (TF) approaches tried to capture higher-order interaction but are typically weakened by the cold start issue in their efficacy. Recurrent Neural Networks (RNNs), which have the reputation of being great at sequence modeling, have outperformed FPMC and TF in sequential user check-in pattern modeling. However, conventional RNN-based approaches typically lack the ability to cope with continuous time durations and spatial distances, which form the bases of mobility patterns in the real world. As a response to these limitations, recent studies have attempted to integrate spatial and temporal dynamics into the learning framework directly. This thesis leverages such advancements by exploring and developing spatial-temporal modeling methods, i.e., addressing the shortcomings of traditional RNNs using methods like Spatial Temporal Recurrent Neural Networks (ST-RNN) that utilize time- and distance-based transition matrices to abstract more accurately the intricate patterns in user movement data.

conventional RNN-based approaches necessarily neglect the changing time intervals between actions, which can be important in taking into account user intent as well as interest decay over time. To address this limitation, [14] models like Time-LSTM have been introduced that have

temporal gates to directly use time interval information in the learning process. Such improved architectures have led to better performance through more effective representation of user behavior at different timescales, and are promising directions toward building context-aware, time-sensitive recommendation systems

In [15] the majority of earlier methods either deal with long-term and short-term preferences independently or do not represent the impact of contextual conditions like POI type or check-in time properly. In the pursuit of overcoming these issues, recent research has aimed to incorporate attention mechanisms to learn user preferences on the time horizon. However, difficulty is encountered in learning both types of preferences together effectively. This thesis extends these developments with the introduction of a new Long- and Short-Term Preference Learning (LSPL) model that combines sequential behavior and context information using distinct LSTM models for location and category sequences to finally offer a more precise and personalized next POI recommendation.

The conventional strategies have been to resort to previous user check-ins and proximity for next POI recommendation. [16] explains that while these methods are not capable of dealing with the dynamic nature of users' movement and with multifaceted, heterogeneous contextual factors affecting users' behavior. Latest developments have included increasingly advanced models like Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks to model sequential dependency and make user path prediction. The literature also suggests the need for attention mechanisms in improving model performance through selective attention to useful information. This thesis makes contributions to society by providing the TMCA (Temporal and Multi-level Context Attention) model, which combines temporal patterns and heterogeneous contexts into a single framework, and adaptive attention mechanisms for enhanced POI prediction. The validity of the proposed model in tackling both spatial-temporal dependencies and context heterogeneity is verified by comprehensive experiments with better performance compared to state-of-the-art methods

CHAPTER – 4 LITERATURE SURVEY

1.1 Introduction to Literature Survey

Point-of-Interest (POI) recommendation is a personalization service in Location-Based Social Networks (LBSNs) that recommends points of interest that users are most likely to be interested in, like restaurants, parks, or tourist attractions, based on their interests, activities, and location. Unlike the usual recommendation method, POI recommendation must take into consideration user-item interactions and spatial, temporal, and social factors. Because of these factors, it is especially difficult and data-intensive. The aim is to enable users to discover new locations in a way that improves their overall location-based experience, with commercial value to businesses in targeting the right audience. POI recommendation has been a core issue in mobile applications, city planning, and smart tourism, and continues to evolve with improved machine learning, AI, and location-aware computing.

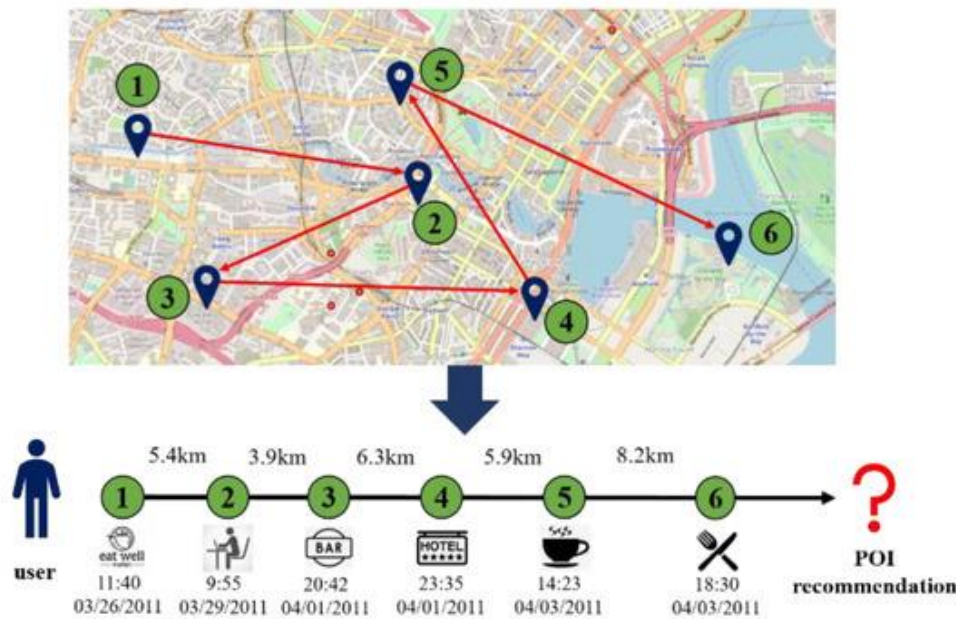


Fig 4 Example showing the POI Recommendation

1.2 History

The construction of POI recommenders started in parallel with the emergence of LBSNs at the tail-end of the 2000s when Foursquare and Yelp, among others, began gathering massive-scale location information and check-ins. Initial POI recommenders utilized collaborative filtering and content-based approaches borrowed from classical e-commerce recommenders. These approaches suffered from data sparsity and the natural spatial-temporal nature of user mobility.

Because of these restrictions, researchers started incorporating geographic information, social influence, and temporal patterns into matrix factorization and probabilistic approaches. The last decade has witnessed a fierce emphasis on deep learning techniques, especially the use of embeddings, Recurrent Neural Networks (RNNs), attention mechanisms, and graph-based techniques. These recent methods focus on more fully capturing the user behavior using sequential patterns and rich context information and therefore lead to more accurate and dynamic POI recommendations.

1.3 Existing methods

- FPMC-LR [17]: It extends an individual Markov chain by putting it in the context of matrix factorization. It represents a user's probability of transition between a POI and another as a first-order Markov transition subject to the fact that transitions will be local when considered geographically. In reality, FPMC-LR factorizes a 3D tensor of (user, last-location, next-location) to get user and POI latent factors and limits the next-POI candidates to be in the neighborhood of the current location. Integrating the sequential transition probability and a spatial "localized region" constraint, FPMC-LR learns both the Markovian dynamics of check-ins as well as the tendency of the user to visit nearby locations. This classical method proved that incorporating geographic locality into a factorized Markov chain significantly enhances next-POI prediction compared to unconstrained approaches.
- PRME [18]: The Personalized Ranking Metric Embedding (PRME) method treats next-POI recommendation as a ranking and metric learning problem. Both users and POIs are represented in two latent spaces: a sequential transition space, and a user-preference space. In the sequential space, each POI has a latent vector and the Euclidean distance between the current location and a candidate location as defined by the transition likelihood. In the preference space, each user and POI have latent vectors whose distance reflects the extent

of the liking of the user for the POI. At prediction time, PRME orders candidate POIs by a weighted sum of these distances such that both the Markovian transition signal and the user's long-term favourite are considered. This approach has a cut-off in time-gap as well: when there are two far-apart in-time check-ins, it simply leverages the user-preference distance perfectly disregarding stale sequential influences. It learns these latent embedding's to minimize pairwise ranking loss, PRME thereby obtains a personalized measure per user accounting for sequential action and user-based preferences. Feng et al. (2015) demonstrated PRME (and geospatial alternative PRME-G) outperforms standard factorization baselines on LBSN check-in data

- ST-RNN [13]: Spatial-Temporal RNN extends a basic recurrent neural network by including continuous time and distance contexts explicitly in recurrence. ST-RNN incorporates time-dependent and distance-dependent transition matrices in each RNN layer, which are gates that depend on the passed time and traversed distance between two consecutive. Intuitively, the effect of a previous visit is weighted by how far back in the past it was and how far the user travelled: shorter time horizons and shorter distances weigh more. By discretizing time and distance into bins and learning different transition matrices for each bin, ST-RNN models short-term and periodic mobility patterns. The model therefore learns dynamic user hidden states that naturally incorporate spatio-temporal missing values automatically, elegantly combining recent (short-term) sequence data with long-term past. Such a design was demonstrated to yield significantly enhanced accuracy over plain RNNs by more accurate modeling of the continuous spatio-temporal properties of check-in sequences cdn.aaai.org.
- DeepMove [20]: DeepMove is an attention-based LSTM model for modeling long, sparse trajectories. It initially embeds check-in, along with its context (user ID, POI ID, time, etc.), into a multi-modal embedding and passes it to an RNN to learn sequential transitions. Most importantly, DeepMove contains a historical attention mechanism: subsequent to processing the sequence, it glances back over the entire history of the user using two levels

of attention to obtain periodic mobility patterns (weekday vs. weekend patterns, for example) and indicate related previous visits. At run-time, the RNN will learn the current path's hidden state, and the attention layers learn to assign different previous hidden states' weights when predicting the next POI. The periodic attention usage at multiple levels allows DeepMove to leverage patterns such as weekly or daily cycles, making the predictions more accurate and interpretable. Experimental results in Feng et al. (2018) demonstrate that this two-stage model (periodic attention + multi-modal RNN) significantly surpasses existing neural methods by a large margin on real LBSN datasets.

- GETNext [21]: GETNext is a novel hybrid model combining graph neural networks and a transformer. It begins by building a global trajectory flow map by compiling all users' check-in sequences into a directed graph of POIs, in which edges represent the frequency of transitions between POIs. This flow map is then processed with a Graph Convolutional Network (GCN) to generate embeddings for each POI that capture frequent, user-agnostic transition patterns.

In parallel, the user's own recent path and situation (short-term sequence, long-term interests, temporal attributes, POI categories, etc.) are represented by a Transformer network.

The fundamental premise is that the GCN-obtained POI embeddings infuse global collaborative information into the Transformer's prediction. During training, the transformer pays attention to both flow-based POI embeddings and the user-specific sequence embeddings, and produces the most likely next POI. By combining a learned global POI-to-POI transition probability map and robust attention-based sequence modeling, GETNext can utilize both collective mobility patterns and individual history. Yang et al. inform us that this graph-enhanced transformer achieves significantly better performance than state-of-the-art baselines by an enormous margin [arxiv.org](https://arxiv.org/abs/2006.08001).

- HKGNN [22]: The Hyper-Relational Knowledge Graph Neural Network (HKGNN) models the data of LBSN as a rich graph and uses hypergraph neural layers. The authors

construct a hyper-relational knowledge graph where nodes contain users, POIs, times, categories, etc., and hyper-edges encode multi-way relations (such as a ternary relation (user, POI, time) for each check-in) This is followed by a specialized hypergraph convolution, which spreads information along these high-order relations, enabling the model to leverage rich structural patterns beyond pair-wise links. For modeling sequential behavior, HKGNN further uses a self-attention module across each user's ordered check-in sequence, enabling the model to concentrate on the most appropriate previous visits for predicting the next location. In practice, HKGNN combines graph-based context encoding (via the hyper-relational graph) with sequence modeling (via attention), enabling it to address data sparsity and utilize side information. Experiments on four real LBSN datasets demonstrate that HKGNN, through the combination of semantic context and sequential attention, outperforms previous deep models in next-POI accuracy

1.4 Challenges

- **Data Sparsity and Cold Start:** Although methods such as embeddings and graph structures improved some of the data sparsity, most users typically have scant check-in records or sparse contextual information, and it is hard to correctly represent their preferences. Cold-start problem of new users and POIs also exists to a large degree, particularly in real-world, dynamic LBSN scenarios.
- **Capturing Long-Term and Short-Term Preferences:** Most models tend to either over-emphasize short-term sequential movements (e.g., Markov models) or attempt to model complete histories with insufficient temporal focus. Keeping the balance between short-term sequential activity and long-term user preference continues to be a challenge, particularly because user behaviors can be non-linear and discontinuous.
- **Scalability and Real-Time Prediction:** Deep learning-based models such as RNNs, Transformers, and GCNs are resource hungry and difficult to deploy for real-time

recommendation tasks. Responsiveness and keeping models lightweight without affecting high performance is a perennial problem.

- **Interpretability:** Most of the deep-learning-based models, although powerful, are black-box models. It is not easy to interpret why a certain POI has been suggested, and that compromises trust and usage in real-world contexts. Comprehensible recommendation mechanisms must be built into models, particularly in sensitive areas like tourism or city planning.
- **Context Integration:** While models such as HKGNN and TMCA are capable of dealing with multiple contexts (e.g., time, category, user profiles), there is no one and adaptive mechanism in most systems to support new and emerging context types (e.g., weather, user sentiment, local events). Dynamic and adaptive context modeling is still an open issue.
- **Evaluation Metrics and Real-World Benchmarks:** Most of the studies work with small datasets and common metrics such as precision@k or MAP. These may be far from the real effectiveness of the recommendation in practical scenarios. Larger benchmarks including user satisfaction, diversity, novelty, and real-time feedback are necessary for comprehensive assessment.

1.5 Future Directions

- **Unified Multi-Objective Models:** Future models will be more integrated in handling objectives such as user preference, POI popularity, geographical closeness, social influence, and even reviews or content of POIs. Training models that optimize together for one or more user-directed objectives (e.g., diversity, novelty, relevance) can improve recommendations.
- **Graph-Enhanced Sequential Models:** By combining graph neural networks (GNNs) and sequential neural networks (RNNs, Transformers), encouraging results have been achieved (e.g., GETNext and HKGNN). Future work can be further investigated using dynamic graphs that evolve with user behavior, or hypergraphs for capturing intricate multi-way relationships.
- **Self-Supervised and Contrastive Learning:** Using self-supervised learning (SSL) to suggest points of interest can minimize reliance on labeled data. Contrastive learning can be used to learn more informative representations through similarities and contrasts of the user's paths despite sparse interactions.
- **Few-Shot and Zero-Shot Learning:** Future systems must deal with data sparsity more aggressively with meta-learning or few-shot learning techniques. Zero-shot recommendation, where the system suggests POIs never encountered during training, would be facilitated through semantic comprehension of POI attributes and user profiles.
- **Privacy-Preserving Recommendation:** Since location data is privacy-sensitive, future models must integrate privacy-conscious mechanisms such as federated learning or differential privacy to offer protection of user data at the cost of performance.
- **Explainable and Interactive POI Systems:** Having explainable AI (explanatory AI) mechanisms at hand for future POI recommendations would raise user trust. Lastly, interactive systems where users can modify or annotate recommendations in real-time would be a satisfaction and personalization booster as well.

1.6 Applications

Then there are POI recommendation systems with numerous applications in real life that facilitate user experiences and assist decision-making in a vast array of fields. In location-based social networks such as Foursquare, Facebook Places, and Yelp, the systems assist users in finding new and relevant places such as restaurants, stores, or attractions based on their current location and past movement patterns. In city planning and intelligent city planning, POI recommendation data can guide traffic control, public transportation optimization, and infrastructure design based on the analysis of aggregate mobility patterns. Travel destinations utilize next POI recommendations to recommend personalized tours and local places of interest to travelers in real-time. In marketing and trade, companies employ POI recommender systems to provide targeted location-based promotions or ads to users when they are about to go to a particular destination. In addition, transportation services such as bike-sharing or ride-hailing apps have the potential to refine routing and resource utilization by predicting where users are most likely to go next. Such apps not only improve user satisfaction and experience but also offer businesses and city planners useful insights regarding consumer behavior and spatial patterns

CHAPTER – 5 METHADODOLOGY

The STDT model is a spatio-temporal dynamics with transformer model is designed particularly for Point-of-Interest (POI) recommendation tasks by seamlessly combining user identity, geographical context, and sequential check-in information. The Fig-5 represent the overall architecture of the proposed spatio-temporal dynamics with transformer model.

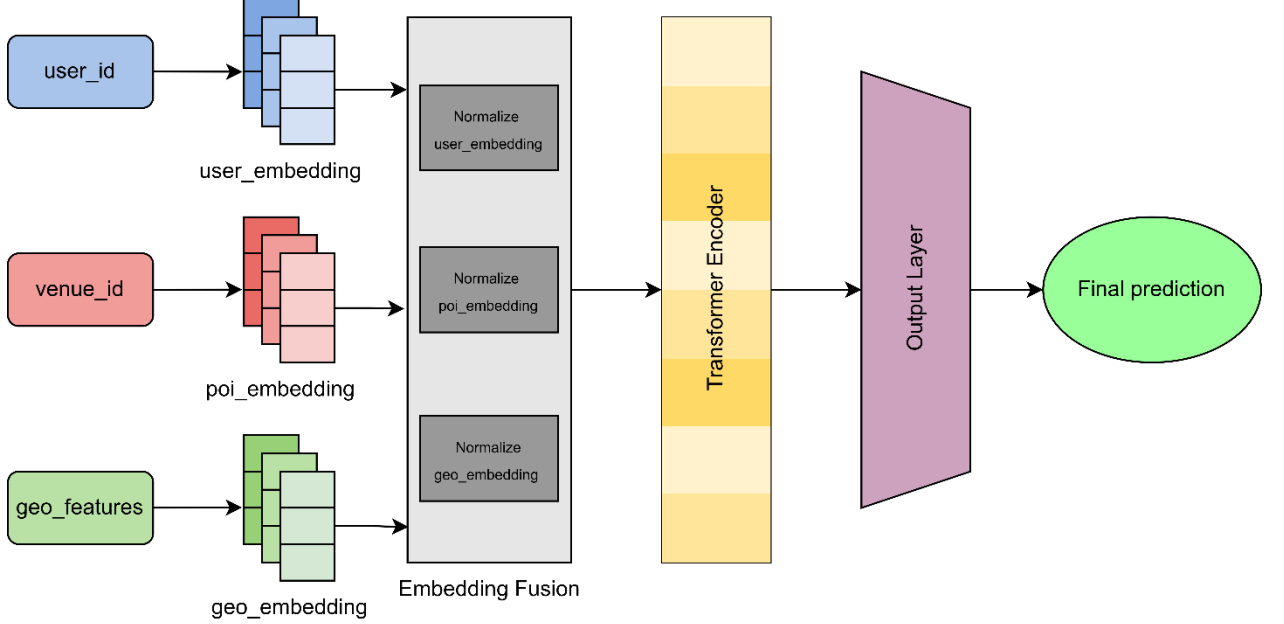


Fig 5: overall architecture of STDT model

5.1 Embedding layers

It starts with embedding layers, i.e., `user_embedding` and `poi_embedding`—both trainable PyTorch `nn.Embedding` layers that transform categorical `user_id` and `venue_id` into dense, fixed-length vector representations (`hidden_dim`). The embeddings enable the model to capture latent behavioral patterns and user and location affinities. Besides, a `geo_embedding` module serves as a linear transformation to project 2D geographical coordinates (longitude and latitude) into the same `hidden_dim` space. Spatial info is embedded together with user and POI embeddings.

Let $u \in Z, p \in Z$ and $g \in \mathbb{R}^2$ represent the user ID, POI ID, and geographic coordinates (latitude, longitude), respectively.

- **User Embedding:**
- **POI Embedding** (for a sequence of POIs P_1, P_2, \dots, P_T):

$$E_p = [Embedding_p(P_1), Embedding_p(P_2), \dots, Embedding_p(P_T)]$$

- **Geo Embedding** (linear projection):

$$e_g = W_g \cdot g + b_g \in \mathbb{R}^2$$

Where $W_g \in \mathbb{R}^{d \times 2}, b_g \in \mathbb{R}^d$

5.2 Transformer encoder block

The temporal modeling module employs a Transformer encoder block, which is sufficiently strong in handling long-range dependencies in sequential data. In this format, the input sequence of POI embeddings is rearranged to the transformer multi-head attention module's required shape. The transformer is applied to the entire sequence in parallel, producing contextualized hidden representations for each timestep. The Transformer Encoder Block is a key building block in the architecture responsible for learning temporal dynamics of sequences of user check-ins. As opposed to normal recurrent models such as LSTMs, the block employs a self-attention mechanism, facilitated through `nn.MultiheadAttention`, to learn context-dependent relations among Points of Interest (POIs) within a sequence of user check-ins. This mechanism enables the model to allocate the relative importance of one POI to others, which allows it to learn short-term and long-range temporal patterns without being bound by sequential processing. For training stability and overfitting avoidance, Layer Normalization (LayerNorm) is used before and after the attention and feedforward layers, and Dropout is used for regularization. The block also contains a Feedforward Neural Network, which is usually composed of two linear layers with a non-linear activation function such as ReLU in between, which provides representational depth and allows the model to learn more intricate patterns in the data. This Transformer-inspired architecture facilitates parallel processing of full sequences and hence is computationally effective and more capable of learning finer user patterns over time, replacing the previously utilized TimeLSTM and simple attention mechanism combination.

The hidden state of the final token (`sequence_representation = transformer_output[-1]`) is taken from this sequence of outputs, a dynamic abstraction of the user's latest activity and temporal behavior patterns.

Let $E_p \in \mathbb{R}^{T \times d}$ be the sequence input to the Transformer

- Multi-Head Self-Attention:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Where $Q = E_p W_q, K = E_p W_k, V = E_p W_v$

- Transformer Layer Output:

$$H = TransformerEncoder(E_p) \in \mathbb{R}^{T \times d}$$

Use the last time step for sequence representation:

$$e_t = H_T$$

5.3 Embedding fusion module

Next, the feature fusion module combines information from different sources by adding up the user embedding, temporal sequence representation, and geographical embedding. This combination aggregation process combines the user's personalized behavior features, the spatial impact of where they are, and the temporal dynamics of their check-in history into one aggregate feature vector. This aggregate feature is the basis to produce the final recommendation.

Combine user, temporal, and spatial features into a unified representation:

$$f = e_u + e_t + e_g \in \mathbb{R}^d$$

Alternatively, use concatenation followed by a linear projection:

$$f = \text{ReLU}(W_f[e_u; e_t; e_g] + b_f)$$

Where $W_f \in \mathbb{R}^{dx3d}$, $b_f \in \mathbb{R}^d$

5.4 Output Layer

Lastly, the output layer includes a fully connected (`nn.Linear`) layer and a `log_softmax` activation, projecting the concatenated feature vector to a log-probability distribution over all possible points of interest (POIs). The architecture facilitates training the model on negative log-likelihood loss (`NLLLoss`), which is most appropriate for multi-class classification problems such as POI recommendation.

Map the fused vector to a POI score distribution over all possible POIs:

$$\hat{Y} = \text{softmax}(W_o f + b_o)$$

Where $W_o \in \mathbb{R}^{N \times d}$, $b_o \in \mathbb{R}^N$, and N is the number of POIs

Loss Function: Use Negative Log Likelihood (NLL) Loss for training,

$$\mathcal{L} = -\log \hat{Y}_{P^+}$$

Where P^+ is the ground truth POI index.

Using `log_softmax` ensures numerical stability and meaningfully interpretable probabilistic outputs, which subsequently are ranked and scored by way of ranking metrics such as Accuracy@k and NDCG@k.

5.5 Training loop

During the course of the study, the training loop is a central component in the model of the POI recommendation that is being optimized by continuously adapting the model parameters to achieve minimization of prediction error as well as enhancement of generalization. Training is concluded to be a supervised learning task, wherein the model is trained to predict the subsequent Point-of-Interest (POI) that a user checks-in based on prior check-in history, geospatial features, and users' preferences. It is optimized with the help of Negative Log Likelihood Loss (NLLLoss), a standard loss function for multi-class classification problems with log-softmax output. The loss function penalizes the model for mistakes by comparing the predicted probability distribution over POIs to the ground truth index of a POI, so that it will learn to give higher probabilities to correct suggestions. To update the model parameters during training, we use the AdamW optimizer, an extension of the Adam optimizer which separates weight decay from the gradient update. The optimizer achieves faster convergence and improved generalization by avoiding over-regularization of parameters, particularly in deep models such as ours comprising embedding layers and Transformer encoders. The training data batch is fed through the model in order to calculate the loss, and gradients are subsequently back-propagated to update model weights. A key element of this cycle is gradient clipping, employed to prevent exploding gradients—of particular interest when training deep neural networks with recurrent or attention architectures on long input sequences. This method preserves numerical stability and convergence smoothness by limiting gradient norms within some pre-specified interval. The training loop also includes a periodic evaluation phase, where the model's performance is tested on a held-out test set using the Accuracy@k and NDCG@k metrics. The model's predictions are compared to the actual POIs in the test set at the end of every epoch to calculate these metrics. This gives instantaneous feedback on model generalization to new data, allowing for early stopping or hyperparameter adjustment based on validation trends. In summary, the training loop combines loss optimization, stability improvement, and online testing, and thus forms the core of constructing a correct and robust POI recommendation system.

CHAPTER – 6 RESULT AND DISCUSSION

6.1 Evaluation metrics

Evaluation metrics are crucial in measuring the performance of recommend models because they give a quantitative measure of how well the model performs to get good items for the users. In this paper, we use a collection of standard metrics—Precision@k, Recall@k, F1@k, Accuracy@k, and NDCG@k at cutoff $k = 1$ and $k = 2$, which measure the quality of top recommendations. These measures enable one not only to estimate the existence of pertinent items in the proposed list but also their rank and density. Examining performance on small values of k corresponds to instances where users might employ only the top handful of the recommendations, for which these measures are especially well-adapted to real-world recommendation tasks.

- Recall@k estimates the ratio of successfully retrieved relevant items to the top-k recommendations. For one user, for a specific k, it is a ratio with the number of relevant items in the top-k results as the numerator and the number of relevant items as the denominator. High recall@k shows that the system is capable of hitting relevant content well. It is calculated as:

$$Recall@k = \frac{|Recommended_k \cap Relevant|}{|Relevant|}, k \in \{1,2\}$$

- Precision@k estimates the top-k recommended items' precision by dividing the number of relevant items in the top-k results by k. It informs us about the number of the recommended items that are relevant. High precision@k indicates that the system returns relevant results with few noises. It is calculated as:

$$Precision@k = \frac{|Recommended_k \cap Relevant|}{k}, k \in \{1,2\}$$

- Accuracy@k checks whether there is at least one relevant item among the top-k recommendations. In binary relevance, it can be interpreted as a hit or success rate. Accuracy@k is especially beneficial in evaluating systems where returning any accurate recommendation is important. It is calculated as:

$$Accuracy@k = \begin{cases} 1, & \text{if } |Recommended_k \cap Relevant| > 0 \\ 0, & \text{otherwise} \end{cases}, k \in \{1,2\}$$

- F1@k is the harmonic average of recall@k and precision@k, combining both into one score by averaging them. It comes in handy when false negatives as well as false positives

are important. A high F1@k signifies that the system not only responds with a large number of relevant items but also with high precision. It is calculated as:

$$F1@k = 2 \times \frac{Precision@k \times Recall@k}{Precision@k + Recall@k}, k \in \{1,2\}$$

- NDCG@k (normalized Discounted Cumulative Gain) estimates ranking quality by giving more weight to relevant items that occur toward the beginning of the recommendation list. Relevance and position are taken into account by nDCG@k. Normalized against the ideal DCG, nDCG@k is between 0 and 1. Evaluates the quality of the ranking list. It is calculated as:

$$NDCG@k = \begin{cases} \frac{1}{\log_2(Rank_i + 1)}, & Rank_i \leq k \\ 0, & Rank_i > k \end{cases}$$

where $Rank_i$ represents the position of the target POI l_i in the ranking list.

6.2 Training and Test results

The below graphs shows the variation of accuracy, precision, recall, f1score and ndcg at different epochs. The graphs are shown separate for New york and Gowalla dataset.shown in Fig 5 and Fig 6 respectively

The results of the evaluation show the performance of the recommendation model on various metrics on the New York and Gowalla datasets at cutoff values $k = 2$ and $k = 5$. On the New York dataset, it does quite well with an Accuracy@2 of 0.5292 and an F1@2 of 0.3528, suggesting that it often has at least one useful item in the top-2 recommendations and has a good balance between precision and recall. With an NDCG@2 of 0.5129, useful items also appear towards the top of the ranked list. With a rise in k to 5, the accuracy and recall improve but at lower precision, showing retrieval of more relevant items with a rise in the number of irrelevant items being retrieved too, rendering the precision worse. On the Gowalla dataset, model performance is significantly lower on all metrics with Accuracy@2 at 0.3818 and F1@2 at 0.2546, which shows the poor capability of retrieving relevant items with effectiveness. The lower NDCG scores consistently also reflect worse ranking quality. The findings imply that the model has better generalizability on the New York dataset compared to Gowalla, maybe because of variation in data sparsity, user behavior, or item diversity among the datasets.

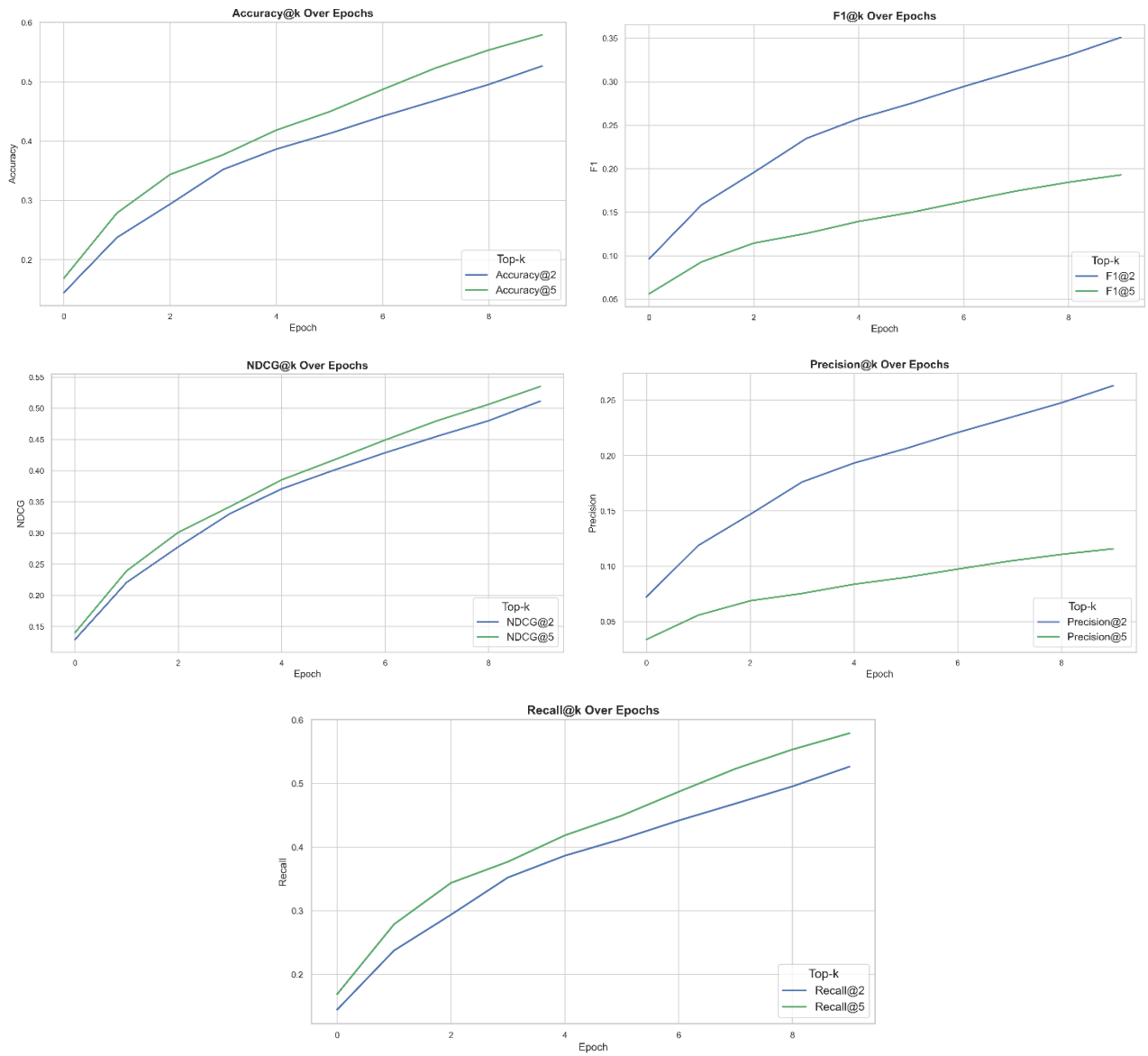


Fig 6: Variation of evaluation metrics on Newyork Dataset

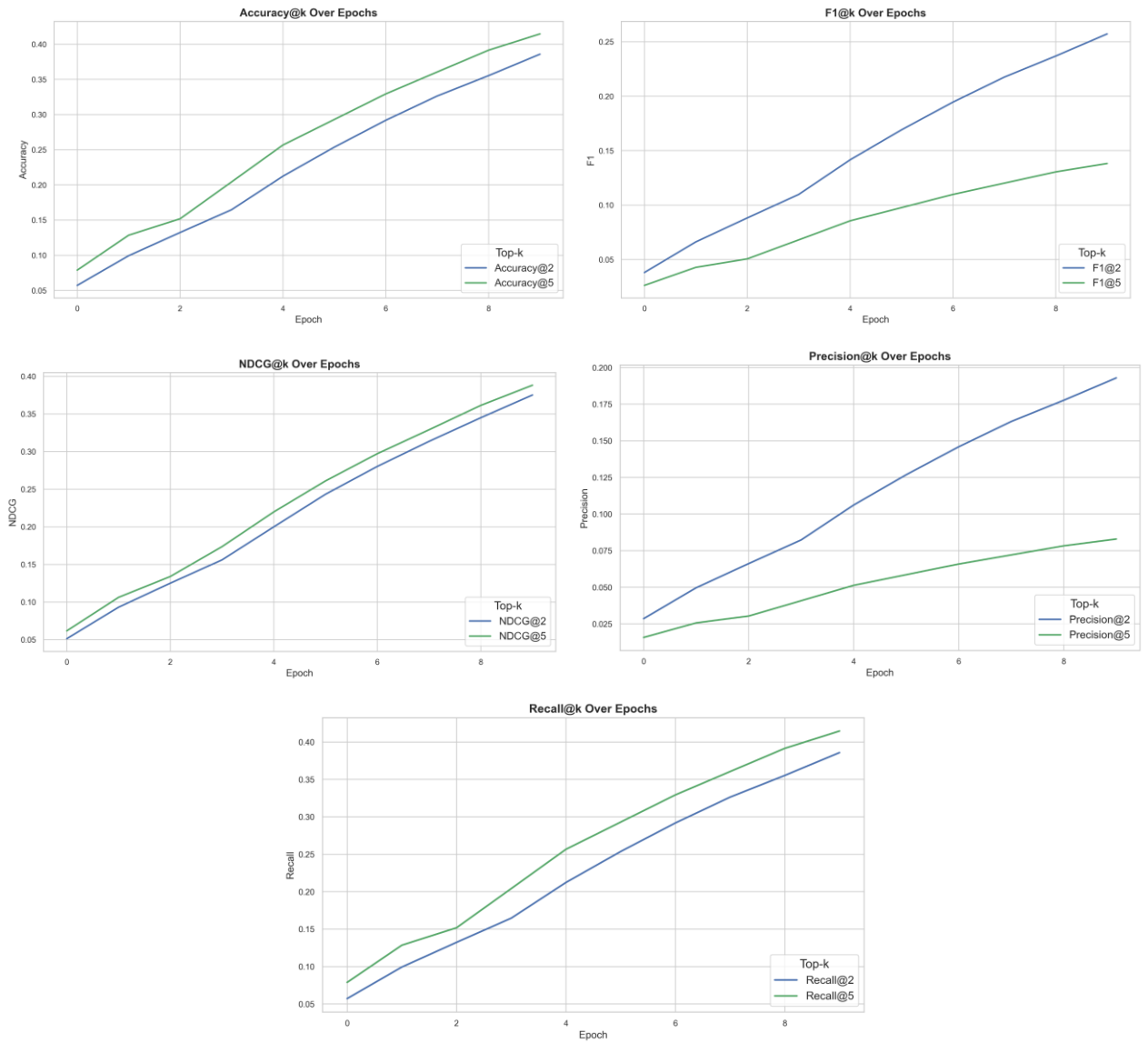


Fig 7: Variation of evaluation metrics on Gowalla Dataset

The test results for the New York dataset show that the recommendation model performs very well in returning relevant items, especially in top-2 and top-5 recommendations. For $k = 2$, the model gets a Recall@2 and Accuracy@2 of 0.5292, which means that in more than 52% of the instances, it can place at least one relevant item in the top two recommendations. The Precision@2 of 0.2646 indicates that, on average, approximately 26% of the top-2 items are relevant, and the F1@2 score of 0.3528 indicates a good balance between recall and precision. The nDCG@2 of 0.5129 indicates that relevant items rank high, leading to an overall good ranking quality. For $k = 5$, accuracy and recall continue to increase to 0.5850, indicating that the model recommends more suitable items more frequently with more recommendations made. Nonetheless, precision significantly decreases to 0.1170, which shows that the proportion of top-5 items that are not relevant increases. The F1@5 score falls to 0.1950, which represents the issue of sacrificing the high relevance density among top recommendations for more relevant item retrieval. Although accuracy has decreased, the nDCG@5 of 0.5379 is still extremely high and indicates that even when looking at an expanded list of recommendations, the model places the correct items in higher positions. Overall, the model is doing very well with this dataset, particularly when in finite recommendation situations.

Evaluation Metrics \ Dataset	New York		Gowalla	
	K=2	K=5	K=2	K=5
Accuracy@k	0.5292	0.585	0.3818	0.4137
NDCG@k	0.5129	0.5379	0.3730	0.3873
Precision@k	0.2646	0.1170	0.1909	0.0827
Recall@k	0.5292	0.5850	0.3818	0.4137
F1@k	0.3528	0.1950	0.2546	0.1379

Table: 4 Result of STDT model on newyork and gowalla dataset on different evaluation metrics

The outcome on the Gowalla dataset reveals that the recommendation model's performance is lower than the New York dataset. With $k = 2$, the model achieves an Accuracy@2 and Recall@2 of 0.3818, which translates to the ability to recall at least one item that is of relevance in about 38% of instances among the top two recommendations. Yet, the Precision@2 is 0.1909 low, with just some 19% recommended items, on average, being relevant, which indicates a very high rate of irrelevant recommendations. The F1@2 of 0.2546 is an indicator of precision-recall imbalance. Furthermore, the nDCG@2 of 0.3730 indicates that the relevant items are recommended at low ranks in the list, which lowers ranking recommendation performance. At $k = 5$, even though both Accuracy and Recall improve to 0.4137, the Precision dips to 0.0827, and dilution of relevance in the recommended list is increasing. The F1@5 score improves to 0.1379, which again indicates that the compromise in achieving recall at the expense of precision deteriorates as k gets bigger. The nDCG@5 of 0.3873 is still low, which means that even when more are recommended, good things don't necessarily come out at the top. In general, these results indicate that the model performs worse on the Gowalla dataset, perhaps because there is more sparsity, more user diversity, or more item diversity, all of which make it more difficult to have a successful recommendation.

CHAPTER – 7 CONCLUTION AND FUTURE SCOPE

Two location-based social network datasets in real-world, New York and Gowalla, were employed to train and evaluate a Transformer-based model. The objective was to utilize the self-attention mechanism of Transformers to learn sequential and context-aware patterns in user-item interactions to provide personalized top-k recommendations. Experiments at a large scale were performed, and the model was tested on common metrics like Precision@k, Recall@k, F1@k, Accuracy@k, and NDCG@k for $k = 2$ and $k = 5$. Performance of the New York dataset was promising with an excellent recall capability of related items along with high precision and ranking quality reflected in better recall, accuracy, and NDCG values. Conversely, performance on Gowalla dataset was relatively poorer, conveying challenges of higher sparsity and heterogeneity in user activity. In spite of the performance variation, the Transformer framework was able to prove its capability for complex modeling of user activity with plenty of data density at hand. The project is optimal in corroborating the possibility of using sequence modeling methods for recommendation application and offers a strong foundation for future optimization. The article also makes reference to precision and recall trade-offs when the number of recommendations is being expanded and the need for ranking quality in user-confronted products.

The work has various promising avenues upon which it may be further improved to counter existing limitations and offer better recommendation results. One promising direction of serious improvement is enhancing the overall generalization capability of the model with respect to sets of datasets involving varying sparsity and user usage patterns. This can be addressed by providing additional context signals like timestamp, geolocation, user type, or even textual content related to items to make the model better aware of user taste. Including cutting-edge methods like contrastive learning to make users more robust, reinforcement learning to represent dynamic recommendation, or graph neural networks to handle higher-order interaction between items and users might even make it even better. In addition, hybrid signal-based collaborative and content-based recommendation systems can alleviate cold-starting and enhancing recommendation diversity. Scalability and efficiency are also key research areas in the future, as these models need to be used in real-world systems with high-speed inference as well as ability to adapt to real-time user behavior. Lastly, evaluation of the model online or through user studies could provide valuable feedback on user satisfaction and recommendation utility to feed back into future research on how to design and integrate more compelling and user-focused recommendation systems.

REFERENCES

- [1] Y. Liu, T.-A. N. Pham, G. Cong, and Q. Yuan, “An experimental evaluation of point-of-interest recommendation in location-based social networks,” *Proc. VLDB Endowment*, vol. 10, no. 10, pp. 1010–1021, 2017.
- [2] R. M. Bell and Y. Koren, “Lessons from the netflix prize challenge,” *ACM SIGKDD Explorations Newslett.*, vol. 9, no. 2, pp. 75–79, 2007
- [3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- [4] Rao, X., Jiang, R., Shang, S., Chen, L., Han, P., Yao, B., & Kalnis, P. (2024). Next Point-of-Interest Recommendation with Adaptive Graph Contrastive Learning. *IEEE Transactions on Knowledge and Data Engineering*.
- [5] Liu, Y., Pham, T.-A. N., Cong, G., & Yuan, Q. (2017). An experimental evaluation of point-of-interest recommendation in location-based social networks. *Proceedings of the VLDB Endowment*, 10(10), 1010-1021
- [6] Y. Liu, C. Liu, B. Liu, M. Qu, and H. Xiong, “Unified point-of-interest recommendation with temporal interval assessment,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1015–1024
- [7] Chen, M., Li, W. Z., Qian, L., Lu, S. L., & Chen, D. X. (2020). Next POI recommendation based on location interest mining with recurrent neural networks. *Journal of Computer Science and Technology*, 35, 603-616.
- [8] Jannach, D., & Ludewig, M. (2017, August). When recurrent neural networks meet the neighborhood for session-based recommendation. In *Proceedings of the eleventh ACM conference on recommender systems* (pp. 306-310).
- [9] Liu, T., Liao, J., Wu, Z., Wang, Y., & Wang, J. (2019, June). A geographical-temporal awareness hierarchical attention network for next point-of-interest recommendation. In *Proceedings of the 2019 on international conference on multimedia retrieval* (pp. 7-15)
- [10] Yang, D., Fankhauser, B., Rosso, P., & Cudre-Mauroux, P. (2020). Location prediction over sparse user mobility traces using rnns. In *Proceedings of the twenty-ninth international joint conference on artificial intelligence* (pp. 2184-2190).
- [11] Sun, K., Qian, T., Chen, T., Liang, Y., Nguyen, Q. V. H., & Yin, H. (2020, April). Where to go next: Modeling long-and short-term user preferences for point-of-interest recommendation. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, No. 01, pp. 214-221)

- [12] Kong, D., & Wu, F. (2018, July). HST-LSTM: A hierarchical spatial-temporal long-short term memory network for location prediction. In *Ijcai* (Vol. 18, No. 7, pp. 2341-2347)
- [13] Liu, Q., Wu, S., Wang, L., & Tan, T. (2016, February). Predicting the next location: A recurrent model with spatial and temporal contexts. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 30, No. 1).
- [14] Zhu, Y., Li, H., Liao, Y., Wang, B., Guan, Z., Liu, H., & Cai, D. (2017, August). What to do next: Modeling user behaviors by time-LSTM. In *IJCAI* (Vol. 17, pp. 3602-3608).
- [15] Wu, Y., Li, K., Zhao, G., & Qian, X. (2019, November). Long-and short-term preference learning for next POI recommendation. In *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 2301-2304).
- [16] Li, R., Shen, Y., & Zhu, Y. (2018, November). Next point-of-interest recommendation with temporal and multi-level context attention. In *2018 IEEE International Conference on Data Mining (ICDM)* (pp. 1110-1115). IEEE.
- [17] Cheng, C., Yang, H., Lyu, M. R., & King, I. (2013, August). Where you like to go next: Successive point-of-interest recommendation. In *IJCAI* (Vol. 13, pp. 2605-2611).
- [18] Zhai, Y., Zhou, Y., Li, X., & Feng, G. (2015). Immune-enhancing effect of nano-DNA vaccine encoding a gene of the prME protein of Japanese encephalitis virus and BALB/c mouse granulocyte-macrophage colony-stimulating factor. *Molecular Medicine Reports*, 12(1), 199-209.
- [19] Feng, J., Li, Y., Zhang, C., Sun, F., Meng, F., Guo, A., & Jin, D. (2018, April). Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 world wide web conference* (pp. 1459-1468).
- [20] Kong, X., Chen, Z., Li, J., Bi, J., & Shen, G. (2024). Kgnext: Knowledge-graph-enhanced transformer for next poi recommendation with uncertain check-ins. *IEEE Transactions on Computational Social Systems*.
- [21] Zhang, J., Li, Y., Zou, R., Zhang, J., Fan, Z., & Song, X. (2023). Hyper-Relational Knowledge Graph Neural Network for Next POI. *arXiv preprint arXiv:2311.16683*.



DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-42

PLAGIARISM VERIFICATION

Title of the Thesis _____

Total Pages _____ Name of the Scholar _____

Supervisor (s)

(1) _____

(2) _____

(3) _____

Department _____

This is to report that the above thesis was scanned for similarity detection. Process and outcome is given below:

Software used: _____ Similarity Index: _____, Total Word Count: _____

Date: _____

Candidate's Signature

Signature of Supervisor(s)

sudev

Final Thesis.pdf



Delhi Technological University

Document Details

Submission ID

trn:oid:::27535:98309197

Submission Date

May 29, 2025, 11:53 AM GMT+5:30

Download Date

May 29, 2025, 11:57 AM GMT+5:30

File Name

Final Thesis.pdf

File Size

1.4 MB

44 Pages

10,692 Words

63,103 Characters





6% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.




Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 10 words)
- Crossref database

Match Groups

-  **28 Not Cited or Quoted 6%**
Matches with neither in-text citation nor quotation marks
-  **0 Missing Quotations 0%**
Matches that are still very similar to source material
-  **0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 5%  Internet sources
- 2%  Publications
- 5%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- 28 Not Cited or Quoted 6%**
Matches with neither in-text citation nor quotation marks
- 0 Missing Quotations 0%**
Matches that are still very similar to source material
- 0 Missing Citation 0%**
Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%**
Matches with in-text citation present, but no quotation marks

Top Sources

- 5% Internet sources
- 2% Publications
- 5% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	dspace.dtu.ac.in:8080	3%
2	Internet	www.dspace.dtu.ac.in:8080	<1%
3	Internet	repositorio.ufsc.br	<1%
4	Submitted works	University of Hertfordshire on 2024-08-18	<1%
5	Internet	discovery.researcher.life	<1%
6	Internet	biologyinsights.com	<1%
7	Internet	medium.com	<1%
8	Internet	cris.brighton.ac.uk	<1%
9	Internet	weaviate.io	<1%
10	Internet	www.isteonline.in	<1%

11	Internet	arxiv.org	<1%
12	Internet	journal.hep.com.cn	<1%
13	Submitted works	University of East London on 2021-09-08	<1%
14	Submitted works	University of East London on 2024-09-07	<1%
15	Internet	www.mdpi.com	<1%
16	Submitted works	CSU, San Jose State University on 2024-04-22	<1%
17	Publication	Dhar, Sudipta. "Bert Based Sequential Mining for Richer Contextual Semantics E-C..."	<1%
18	Submitted works	Queensland University of Technology on 2020-09-03	<1%
19	Internet	www.arxiv-vanity.com	<1%
20	Internet	www2.mdpi.com	<1%

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely AI generated as AI generated and AI paraphrased or likely AI generated and AI paraphrased writing as only AI generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?

The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?

Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

